



HAL
open science

Optimisation des séquences de pistes et des mouvements au sol sur les grands aéroports

Raphaël Deau

► **To cite this version:**

Raphaël Deau. Optimisation des séquences de pistes et des mouvements au sol sur les grands aéroports. Autre. Institut National Polytechnique de Toulouse - INPT, 2010. Français. NNT : 2010INPT0060 . tel-04274698

HAL Id: tel-04274698

<https://theses.hal.science/tel-04274698>

Submitted on 8 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Université
de Toulouse

THÈSE

En vue de l'obtention du
DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :
SLCHP

Présentée et soutenue par :
Raphaël Deau

le : mardi 2 novembre 2010

Titre :
Optimisation des séquences de pistes et
des mouvements au sol sur les grands aéroports

JURY

Marc Schoenauer (Président du jury)
Eric Féron (Rapporteur)
Hervé Breton (Membre de jury)
Jean-Marc Alliot (Membre du jury)
Dominique Colin de Verdière (Membre de jury)

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
APO (IRIT)

Directeur(s) de Thèse :
Nicolas Durand
Jean-Baptiste Gotteland

Rapporteurs :
Eric Féron
Marc Schoenauer

OPTIMISATION DES SÉQUENCES DE PISTES ET DU TRAFIC AU SOL SUR LES GRANDS AÉROPORTS

Résumé : *Ces dernières années, la phase de roulage au sol des avions a été mise en avant dans l'étude des retards aériens sur les grands aéroports. Cependant, le lien entre cette phase et l'optimisation des séquences d'avions sur les pistes reste encore peu étudié. L'objectif de réaliser des séquences optimales sur les pistes doit pourtant permettre de mieux gérer le trafic au sol, pour respecter les créneaux de décollage imposés tout en réduisant les retards des avions : dans cette thèse, un algorithme de calcul de séquences optimales est mis en place et intégré à la gestion du trafic au sol, modélisée comme un problème de résolution de conflits entre avions. Deux méthodes d'optimisation sont alors comparées : une méthode déterministe (utilisant un algorithme de type branch & bound) et une méthode stochastique (utilisant un algorithme génétique). chacune des deux méthodes pouvant fonctionner avec et sans considération des séquences optimales sur les pistes. Les simulations effectuées montrent qu'une réduction significative des retards peut être espérée lorsque les séquences sont optimisées et anticipées. La méthode stochastique trouve de meilleures solutions, notamment en ce qui concerne la gestion des arrivées, mais la méthode déterministe reste intéressante, grâce à son temps de calcul bien plus rapide.*

Mots clés : aéroport, optimisation, algorithmes génétiques, branch & bound

RUNWAYS SCHEDULING AND GROUND TRAFFIC OPTIMISATION AT BUSY AIRPORTS

Abstract : *In the last few years, many studies concerning air traffic delays have focused on ground traffic management at busy airports. However, the link between the aircraft taxiing stage and runway scheduling optimisation is still rarely considered. Performing optimal aircraft sequences on runways should allow us to enhance the taxiing stage, while applying calculated take-off slots and reducing globally the aircraft mean delay. In this thesis, an algorithm is first defined to compute optimal aircraft schedules on runways. It is then integrated into the ground traffic management process, modeled as a conflict resolution problem between aircraft. A deterministic method (using a branch & bound algorithm) and a stochastic method (using a genetic algorithm) are both used to try and solve this problem. Each of these methods can work with and without the consideration of optimal runway scheduling. The simulations carried out show that the anticipation of the optimal runway schedules can yield a significant delay reduction for airport ground traffic. The stochastic method provides the best solutions, especially for arriving aircraft, while the deterministic method remains a considerable option because of its very fast running time.*

Key words : airport, optimisation, genetic algorithms, branch & bound

PÔLE POM

PLANIFICATION, OPTIMISATION ET MODÉLISATION DU TRAFIC AÉRIEN

DIRECTION DES SERVICES DE LA NAVIGATION AÉRIENNE

DIRECTION DE LA TECHNIQUE ET DE L'INNOVATION

Remerciements

Je tiens tout d'abord à remercier les personnes qui m'ont permis d'effectuer cette thèse, qui fut une expérience extrêmement enrichissante. Je remercie en particulier Jean-Baptiste Gotteland, mon co-directeur de thèse ainsi que Nicolas Durand, mon directeur de thèse qui ont été mes premiers interlocuteurs et qui m'ont suivi tout au long de ces trois années.

Je souhaite adresser ma gratitude envers les membres du jury qui ont accepté de juger mon travail.

Ce long périple m'a permis de découvrir l'ambiance au sein du *Pôle POM* (initialement nommé *LOG*) où règnent la bonne humeur et la convivialité. Un grand merci à David Gianazza pour son soutien et ses conseils avisés. Je n'oublierais pas non plus les longues discussions avec Cyril Allignol à 8h du matin alors que nous n'étions que deux dans les locaux. Merci également à Nicolas Saporito pour sa bonne humeur, son humour incessant et ses citations aussi philosophiques que cocasses. Je pense aussi à François Huchet qui est un compagnon de jeu vidéo exemplaire et qui est capable de parler de stratégies de jeu d'échecs pendant des heures ! Dans le même registre : merci à David Szymanski pour son expérience dans les MMORPG.

Cette aventure m'a également permis de faire la connaissance de quelques personnes de l'ENAC : Estelle Malavolti, toujours à l'écoute et qui fait preuve d'une grande ouverture d'esprit ; Nicolas Barnier, et ses débats sur des sujets variés : algorithme génétique vs programmation par contraintes, ou simplement politiques ; Kévin Guittet, initiateur à Mounty Hall et grand humoriste qui aurait fait un merveilleux duo sur scène avec Nicolas Saporito. Enfin, j'ai une pensée particulière pour Pascal Brisset qui était une personne extraordinaire et dont l'éloge n'est plus à tisser car déjà présente dans les mémoires de chacun.

Le soutien en dehors du cadre du travail fut tout autant appréciable. Je tiens donc à adresser des remerciements à Dédé qui a, entre autres, eu la bonté de faire l'aller-retour Lyon-Toulouse en camionnette pour mon déménagement ; à Alexandre qui me changeait les esprits avec ses aventures amoureuses ; à Jean-Phi & Laura, qui m'ont rendu accro au Rock et au Boogie ; à Kiki, sans qui le groupe de danse n'existerait plus ; et à tous les autres que je ne cite pas mais qui savent qu'ils comptent beaucoup.

Enfin, je ne remerciais jamais assez ma famille : un énorme merci à ma mère pour toutes les raisons du monde, ainsi qu'à mes grands-parents et à ma sœur. J'ai un sentiment particulier envers mon défunt père dont la seule pensée reste un soutien considérable.

Je tiens à adresser un remerciement spécial à ma femme qui a été d'un soutien inimaginable et qui a comblé ma vie.

Je dédie cette thèse à ma fille

Élyssia

dont la naissance, lors de la rédaction de ce rapport, m'a apporté une joie incommensurable.

Table des matières

Introduction	1
I Contexte	3
1 Le trafic aéroportuaire	5
1.1 Introduction	5
1.2 Organisation du trafic aérien en Europe	5
1.2.1 Contrôle du trafic aérien	5
1.2.2 Régulation européenne du trafic aérien	6
1.3 Gestion du trafic au sol	7
1.3.1 Structure d'un aéroport	7
1.3.2 Contrôle d'aérodrome	8
1.3.3 Domaines d'étude pour le contrôle d'aérodrome	8
1.4 Difficultés identifiées	9
2 Analyse préliminaire	11
2.1 Introduction	11
2.2 Projets pour le contrôle d'aérodrome	11
2.2.1 Optimisation des temps de roulage	11
2.2.2 Optimisation des séquences d'avions sur les pistes	13
2.2.3 Gestion du cheminement des avions	13
2.2.4 Problèmes posés	14
2.3 Simulateurs de trafic aéroportuaire	14
2.3.1 Objectifs	14
2.3.2 Exemples de simulateurs	14
2.4 Cadre de travail de la thèse	15
2.4.1 Questions posées	15
2.4.2 Ordonnancement des avions sur les pistes	15
2.4.3 Lien entre gestion des pistes et gestion du roulage	15
II Séquencement des avions sur les pistes	17
3 Définition du problème	19
3.1 Introduction	19
3.2 Analyse préliminaire	19

3.2.1	Gestion opérationnelle actuelle	19
3.2.2	Heures minimum d'accès à la piste	20
3.2.3	Règles de séparation	20
3.2.4	Gestion des créneaux imposés par le CFMU	21
3.3	Modélisation du problème	22
3.3.1	Présentation générale	22
3.3.2	Variables	22
3.3.3	Contraintes	23
3.3.4	Critère d'optimisation	24
3.4	Analyse du problème	25
3.4.1	Paramètres	25
3.4.2	Espace de recherche	26
3.4.3	Résolution par fenêtre glissante	27
3.4.4	Mesures sur le trafic réel	27
3.5	Conclusion	29
4	Calcul d'une séquence d'avions sur une piste	31
4.1	Introduction	31
4.2	Méthodes de résolution envisageables	31
4.2.1	Algorithmes gloutons	31
4.2.2	Programmation dynamique	32
4.2.3	Programmation linéaire	33
4.2.4	Programmation par contraintes	33
4.3	Résolution à l'aide d'une librairie de programmation par contraintes	35
4.3.1	Principes des bibliothèques de programmation par contraintes (PPC)	35
4.3.2	Description générale de la librairie Facile	35
4.3.3	Implémentation du problème des séquences d'avions	36
4.4	Élaboration d'un algorithme spécifique	37
4.4.1	Principes	37
4.4.2	Première mise en œuvre de l'algorithme	37
4.4.3	Amélioration de l'algorithme	39
4.4.4	Version finale de l'algorithme	40
4.5	Conclusion	42
5	Applications et résultats	43
5.1	Introduction	43
5.2	Exemples de calculs de séquences	43
5.2.1	Présentation	43
5.2.2	Comparaison des deux méthodes	44
5.2.3	Influence du critère	45
5.2.4	Influence de la flexibilité et de la marge	46
5.3	Calcul d'une séquence sur une journée complète	47
5.3.1	Principes	47
5.3.2	Influence de l'horizon	48
5.3.3	Influence des différents critères	50
5.3.4	Influence de la flexibilité et de la marge	51
5.4	Conclusion	51

III	Résolution du trafic au sol	55
6	Formulation du problème	57
6.1	Introduction	57
6.2	Prévision de trajectoires au roulage	58
6.2.1	Affectation de chemins	58
6.2.2	Modélisation des incertitudes	59
6.3	Détection de conflits	61
6.3.1	Définition d'un conflit	61
6.3.2	Conflits liés à l'horizon de prédiction limité	62
6.3.3	Conflits avec incertitude sur les vitesses	62
6.3.4	Groupes d'avions en conflit	63
6.4	Simulation du trafic au sol	64
6.4.1	Principes	64
6.4.2	Heures minimales de décollage	64
6.4.3	Notions de détour et de délai	64
6.4.4	Attente stratégique	65
6.4.5	Boucle de simulation	67
6.5	Formulation en problème d'optimisation	68
6.5.1	Présentation	68
6.5.2	Variables	68
6.5.3	Contraintes	69
6.5.4	Critère sans séquence d'avions à réaliser sur les pistes	69
6.5.5	Critère visant à réaliser des séquences d'avions sur les pistes	70
6.5.6	Taille de l'espace de recherche	72
6.6	Conclusion	73
7	Résolution des conflits au roulage	75
7.1	Introduction	75
7.2	Résolution par une méthode déterministe	75
7.2.1	Classement des avions	75
7.2.2	Calcul de la trajectoire d'un avion sur l'un de ses chemins	77
7.2.3	Résolution des conflits d'un groupe d'avions	78
7.3	Algorithmes évolutionnaires	79
7.3.1	Principes	79
7.3.2	Cycle d'évolution	80
7.3.3	Opérateurs de variation	81
7.3.4	Fonction d'évaluation	82
7.3.5	Améliorations classiques	82
7.3.6	Problèmes partiellement séparables	85
7.4	Algorithme génétique pour la résolution des conflits au roulage	85
7.4.1	Codage des données	85
7.4.2	Population et génération initiale	86
7.4.3	Fonction d'évaluation	86
7.4.4	Reproduction des individus	87
7.4.5	Améliorations classiques	88
7.5	Conclusion	89

8	Résultats	91
8.1	Introduction	91
8.2	Étalonnage de l'algorithme génétique	91
8.2.1	Présentation	91
8.2.2	Taux de croisement et de mutation	92
8.2.3	Taille de population et nombre de générations	93
8.2.4	Sharing	93
8.3	Comparaison des deux méthodes	95
8.3.1	Temps de calcul	95
8.3.2	Comparaison des délais et du respect des créneaux	96
8.3.3	Pertinence des séquences d'avions prévues sur les pistes	98
8.4	Conclusion	99
IV	Applications	101
9	Influence des principaux paramètres	103
9.1	Introduction	103
9.2	Horizon de prédiction des séquences d'avions sur les pistes	104
9.2.1	Présentation	104
9.2.2	Résultats	104
9.3	Taux d'incertitude et pas de rafraîchissement	106
9.3.1	Présentation	106
9.3.2	Résultats	106
9.4	Niveau de priorité entre départs et arrivées	108
9.4.1	Présentation	108
9.4.2	Résultats	108
9.5	Conclusion	109
	Conclusions et perspectives	111
	Glossaire	113
	Bibliographie	115

Table des figures

1.1	Contrôle de la circulation aérienne	6
3.1	Séparations entre avions sur la piste	21
3.2	Fenêtre de tolérance d'un créneau imposé par le CFMU	22
3.3	Insertion d'une marge dans la séquence	24
3.4	Critère pour les vols régulés	25
3.5	Modification de l'espace de recherche en fonction de l'horizon de prédiction	25
3.6	Fenêtre de temps glissante	27
3.7	Nombre maximal d'avions en fonction de la taille de l'horizon, sur la piste 26R	28
3.8	Nombre maximal d'avions en fonction de la taille de l'horizon, sur la piste 27	28
4.1	Exemple d'arbre de solutions en programmation par contraintes	34
4.2	Arbre de permutations	37
4.3	Branches sous-optimales pouvant être supprimées	40
5.1	Temps de calcul des différentes méthodes	44
5.2	Valeur du critère pour chaque méthode	45
5.3	Délai total des départs en fonction du critère	46
5.4	Programmation des créneaux en fonction du critère	46
5.5	Délai des départs en fonction de la marge et de la flexibilité (piste 27 à 12h30)	47
5.6	Programmation des créneaux en fonction de la marge et de la flexibilité (piste 27 à 12h30)	48
5.7	Délai moyen et maximal des départs (toutes pistes) sur la journée, en fonction de l'horizon de prédiction	49
5.8	Programmation des créneaux (toutes pistes), en fonction de l'horizon de prédiction	49
5.9	Délai moyen et maximal des départs (toutes pistes) sur la journée, en fonction du critère	50
5.10	Programmation des créneaux (toutes pistes), en fonction du critère	50
5.11	Délai des départs (sur la piste 27) sur la journée, en fonction de la marge et de la flexibilité	51
5.12	Mesures pour les départs sur la piste 27, en fonction de la marge et de la flexibilité	52
5.13	Programmation des créneaux sur la piste 27, en fonction de la marge et de la flexibilité	52
6.1	Calcul des chemins possibles depuis un parking vers une piste dans ATOS	58
6.2	Représentation des avions avec incertitudes sur la vitesse dans ATOS	61
6.3	Détection des conflits avec incertitude	62
6.4	Chevauchements possibles de zones d'incertitude	62

6.5	Groupement des avions en <i>clusters</i>	63
6.6	Heure stratégique de départ du parking	66
6.7	Simulation par fenêtres glissantes	68
6.8	Fonction de calcul de la pénalité sans prise en compte des séquences de pistes	70
6.9	Fonction de calcul de la pénalité avec prise en compte des séquences de pistes	72
7.1	Processus d'itération d'un algorithme génétique	80
7.2	Exemple de mutation	81
7.3	Exemple de croisement	82
8.1	Étalonnage des taux de croisement et de mutation	93
8.2	Étalonnage de la taille de la population et du nombre maximal de générations	94
8.3	Étalonnage du taux de sharing (%)	94
8.4	Comparaison des temps de calcul des deux méthodes	96
8.5	Temps de calcul et pénalité sur la journée complète pour chaque méthode	97
8.6	Comparaison des délais moyens des deux méthodes	97
8.7	Délais en fonction du nombre d'avions considérés	98
8.8	Répartition des décollages soumis à un créneau imposé par le CFMU	99
8.9	Mesure du respect des séquences d'avions sur les pistes	100
9.1	Influence de l'horizon de prédiction sur les délais	104
9.2	Influence de l'horizon de prédiction sur les créneaux	105
9.3	Influence du taux d'incertitude et du pas de rafraîchissement	107
9.4	Cas d'échecs de la résolution de conflits avec la méthode déterministe	107
9.5	Délai moyen par avion en fonction de la réduction du niveau de priorité des arrivées	109
9.6	Nombre de créneaux ratés en fonction de la réduction du niveau de priorité des arrivées	110

Liste des algorithmes

II.1	Fibonacci : calcul de la suite de Fibonacci par programmation dynamique.	33
II.2	Sequence : calcul de la meilleure séquence d'avions sur une piste	38
II.3	Sequence_Opt : calcul optimisé de la meilleure séquence d'avions	41
III.1	Choix_Attentes : meilleure trajectoire pour un avion, sur l'un de ses chemins	78
III.2	Resoudre : calcule les trajectoires sans conflit des avions	79

Introduction

À l'heure actuelle et malgré la crise économique, le trafic aérien a repris sa croissance : l'OACI (Organisation de l'Aviation Civile Internationale) prévoit un taux moyen de croissance annuel de 4,6% d'ici 2025 (en passagers/kilomètres). Le nombre de vols en constante augmentation est régulé par le CFMU (Central Flow Management Unit) au niveau européen mais expose le système à un risque de saturation dans les dix prochaines années. La crise économique vient par ailleurs ajouter de nouveaux facteurs à prendre en compte : la réduction des coûts pour les compagnies aériennes aussi bien que pour le contrôle aérien devient une priorité. De plus, la prise de conscience des enjeux écologiques est également de mise et conduit à entreprendre de nombreuses actions pour réduire la consommation de carburant ou toute autre forme de pollution, ce qui transforme en profondeur les objectifs des plus grands aéroports.

Une fois de plus, l'aéroport apparaît ainsi au centre de toutes ces considérations. Il est le point de rencontre d'un nombre de plus en plus grand d'intervenants, dont la coordination reste complexe et parfois perfectible. La diversité des procédures d'exploitation au sein d'un aéroport, mais également entre différents aéroports, rendent malheureusement leurs modes de fonctionnement très peu accessibles et difficiles à modéliser dans leur globalité. Le trafic aéroportuaire est en outre caractérisé par des incertitudes élevées (bien plus élevées que celles du trafic en vol), notamment en ce qui concerne les heures de départs effectives (pouvant s'avérer très différentes des heures initialement prévues), les vitesses de roulage (pouvant dépendre des habitudes des équipages) et finalement les temps de roulage (très sensibles aux zones de congestion sur l'aéroport). Encore aujourd'hui, toutes ces incertitudes rendent les heures de décollage effectives des avions très peu prévisibles.

L'aéroport est de plus très sensible à de nombreux facteurs extérieurs comme les conditions météorologiques (un simple orage ou la neige peut paralyser l'ensemble du trafic pour une durée indéterminée) ou les menaces terroristes. Dans des cas moins extrêmes, une mauvaise visibilité peut s'avérer particulièrement problématique pour les avions (contrairement à ce qui se passe en vol). Différentes normes de séparation sont utilisées selon les conditions d'exploitation : par exemple, une largeur des aires protégées de 90 mètres autour des pistes est suffisante par temps clair, mais est élargie à 150 mètres par mauvaise visibilité, pour éviter toute perturbation des aides radio-électriques utilisées pour les atterrissages aux instruments. La sécurité est naturellement un point essentiel pour les aéroports, aussi bien autour des pistes que près des installations, d'autant qu'en cas d'incident ou de collision, la responsabilité entre le contrôle et les pilotes n'est pas toujours évidente à démontrer.

Dans le contexte écologique, l'aérodrome doit également faire face à de nombreux problèmes environnementaux. Les deux points centraux concernent naturellement la limitation des émissions chimiques et la réduction du bruit, y compris lors du roulage. Ainsi, il devient de plus en plus souhaitable de faire attendre les avions au parking, moteurs éteints, plutôt que devant la piste, prêts à décoller.

Afin de s'adapter à tous ces éléments, les aéroports mettent en place de nombreuses innovations technologiques, qui doivent permettre d'aider tous les intervenants dans leurs différentes tâches. Ainsi, des outils tels que les radars de surface précis et sûrs donnent lieu à l'A-SMGCS (Advanced - Sur-

face Movement Guidance & Control System), permettant de localiser précisément tout mobile sur l'aéroport et de diffuser ces positions par l'ADS-B (Automatic dependent Surveillance - Broadcast). D'autres outils intègrent une cartographie détaillée des voies de circulation de l'aéroport et permettent de représenter la position de l'avion et le chemin qu'il doit suivre dans le cockpit (D-GPS : GPS différentiel). La transmission automatique du chemin à suivre aux avions peut alors être envisagée par Data-link (Transmission automatique de données entre le sol et le bord). Dans un contexte plus global, le CDM (Collaborative Decision Making) permet la diffusion de nombreuses informations (retards prévus, changements de vols, etc.) entre les compagnies et le contrôle aérien, pour optimiser l'exploitation des ressources en fonction des imprévus.

Toutes ces innovations donnent accès à de nombreuses données et ouvrent de nouvelles possibilités pour la gestion du trafic au sol. Les concepts associés sont classés en trois catégories : la notion de AMAN (Arrival MANagement) se focalise sur la gestion des arrivées, le concept de DMAN (Departure MANagement) concerne l'organisation des départs et la notion de SMAN (Surface MANagement) se rapporte à la gestion du trafic au roulage. Encore aujourd'hui, ces trois concepts ont tendance à être étudiés séparément, alors que la nécessité de les relier devient de plus en plus évidente, pour intégrer les arrivées bien avant qu'elles n'atterrissent et prévoir les départs suffisamment tôt, de façon réaliste.

Ce dernier point correspond au sujet de cette thèse : après un état de l'art sur les évolutions en cours présentant les différents projets, outils d'aide à la décision ou simulateurs existants (première partie), une étude des possibilités d'optimisation des séquences de décollages et d'atterrissages sur les pistes est menée (seconde partie). Cette étude présente les difficultés liées à la gestion des pistes : les règles de séparation entre mouvements impliquent que l'ordre des avions influence fortement les délais engendrés et le nombre (toujours croissant) de créneaux imposés par le CFMU (en vue de réguler le trafic) restreint les possibilités d'optimisation pour les vols sans créneaux. L'objectif est alors de réduire les délais dus aux pistes tout en gérant de façon pertinente les créneaux de décollages imposés, en optimisant les séquences d'avions programmées sur les pistes.

Par la suite, la réalisation de ces séquences lors de la gestion de la phase de roulage est étudiée (troisième partie) : un modèle de simulation des avions au roulage, incluant la prévision de trajectoires et la détection des conflits entre avions est présenté. Ce modèle permet une résolution optimisée des conflits, en essayant d'absorber les délais dus au roulage dans ceux inhérents aux pistes : deux méthodes d'optimisation sont définies pour résoudre ce problème : l'une est déterministe et s'applique à une simplification du problème global, l'autre est stochastique. Ces deux méthodes sont comparées en terme de réduction des délais, respect des créneaux de décollage et pertinence des séquences d'avions calculées en avance. Enfin, les deux méthodes d'optimisation sont utilisées pour mesurer les effets de différents facteurs caractérisant la gestion de la phase de roulage des avions (dernière partie).

Cette thèse a donné lieu à trois publications : les deux premières aux conférences *International Conference on Research in Air Transportation* [Deau 08] et *Air Traffic Management R&D Seminar* [Deau 09] organisées par EUROCONTROL ; la troisième à la conférence *Recherche Opérationnelle et Aide à la Décision* [Deau 10] organisée par la Société Française de Recherche Opérationnelle et d'Aide à la Décision.

Première partie

Contexte

Chapitre 1

Le trafic aéroportuaire

1.1 Introduction

Le trafic aéroportuaire est constitué des différents appareils utilisant les voies de circulation et les pistes de l'aéroport. L'importance du trafic aérien engendre une grande complexité du trafic sur l'aéroport.

Ce chapitre présente brièvement le rôle des contrôleurs ainsi que le fonctionnement des aéroports et la gestion de l'espace aérien en Europe. Il n'a pas pour objectif d'expliquer en détail le fonctionnement aéroportuaire mais il en présente une vue globale.

Une présentation du trafic aérien permet de comprendre comment ce dernier est géré par les contrôleurs mais également par l'organisme de régulation du trafic européen. Une étude du trafic au sol présente la structure d'un aéroport, son fonctionnement et les outils mis à disposition des contrôleurs pour les aider. Enfin, la dernière partie de ce chapitre présente les points qui semblent être un problème dans la gestion du trafic et qui seront donc à l'étude.

1.2 Organisation du trafic aérien en Europe

1.2.1 Contrôle du trafic aérien

La gestion du trafic aérien représente une lourde responsabilité étant donné que la sécurité doit être garantie tout en maximisant le débit du trafic. Cette sécurité se définit par des règles de séparation entre les avions qui doivent être respectées pour s'assurer qu'aucun problème ne survienne. Afin d'en simplifier la gestion et de répartir les responsabilités, l'espace aérien est découpé en plusieurs centres de contrôle délimités géographiquement. Ces centres sont divisés en secteurs qui seront alors chacun sous la charge de deux contrôleurs : un contrôleur radar en relation directe avec les pilotes et un contrôleur organique en relation avec les autres contrôleurs.

Selon les phases de vol, on distingue différents types de services de contrôle rendus à l'usager de l'espace aérien (figure 1.1 extraite de [Gotteland 04]) :

- **Le contrôle en route** se concentre sur les vols à haute altitude (*niveau de croisière*). A ce niveau, la surveillance du trafic s'effectue par l'intermédiaire d'un radar et en contact radio avec les pilotes. Le contrôleur a la charge de donner des ordres aux avions (changement d'altitude, de cap ou de vitesse) en vue de respecter les séparations réglementaires entre appareils.
- **Le contrôle d'approche** s'occupe des phases de montée et d'approche initiale des avions (la phase de descente se décompose en deux étapes d'approche : initiale et finale). La surveil-

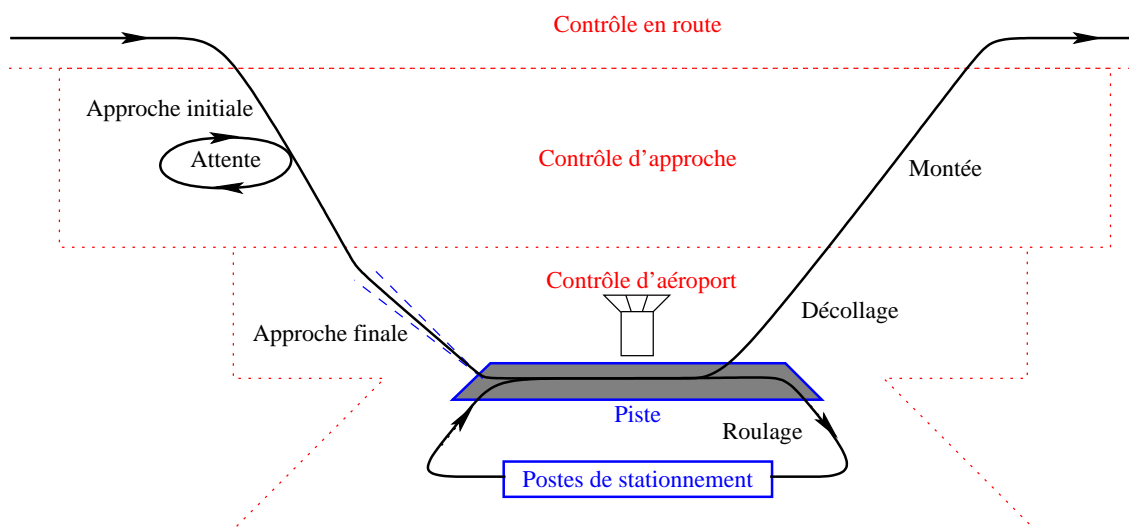


FIG. 1.1 – Contrôle de la circulation aérienne

lance s'effectue également par contrôle radar et contact radio. Lors de l'approche initiale, le contrôleur a la possibilité de retarder l'approche finale d'un avion en rallongeant sa trajectoire ou en utilisant une procédure d'attente (*stack*) afin de réguler le flux d'arrivées.

- **Le contrôle d'aérodrome** se charge des appareils en approche finale ou au décollage, et de la phase de roulage sur l'aéroport qui consiste à acheminer les avions d'un point à un autre en passant par la gestion des zones de parking. Cette phase inclut également la gestion du séquençement des départs et des arrivées au niveau des pistes. Une partie de la surveillance est visuelle (depuis la tour de contrôle).

1.2.2 Régulation européenne du trafic aérien

L'organisme européen CFMU (*Central Flow Management Unit*) a pour objectif d'optimiser l'occupation de l'espace aérien en imposant des contraintes à certains vols.

La régulation du trafic aérien s'effectue en fonction des plans de vol déposés disponibles pour le jour étudié. Elle peut également être influencée par les données des années ou des semaines précédentes ou par certains phénomènes répétitifs (tels qu'une pointe de trafic autour d'un jour férié par exemple).

La gestion des flux de trafic aérien (*ATFM : Air Traffic Flow Management*) permet d'adapter la quantité d'avions qui occupent l'espace aérien afin de ne pas le saturer. La régulation du trafic est réalisée en trois étapes distinctes qui représentent différentes phases de cette gestion des flux :

- **ATFM Stratégique** : cette phase prend place dans les mois qui précèdent le jour étudié. Elle a pour objectif de s'assurer que la quantité de trafic prévu ne dépassera pas la quantité théorique possible. Les vols réguliers se voient affecter certains créneaux de départ et d'arrivée qui sont négociés entre les compagnies lors de cette étape.
- **ATFM Pré-tactique** : quelques jours avant le vol, les prévisions de trafic plus précises sont étudiées et permettent de calculer les créneaux de décollage à imposer en vue de minimiser les retards et de maximiser l'utilisation de l'espace aérien. Les schémas d'ouverture des centres de contrôle sont également définis. Ils déterminent le découpage des centres en secteurs pour chaque période de la journée.
- **ATFM Tactique** : la gestion de la ré-allocation de créneaux en temps réel s'effectue dans cette

phase le jour même du vol. Certains créneaux de décollage sont donc affectés lors de cette étape. Certains vols peuvent également se voir proposer un changement de route ou même être annulés dans certains cas très particuliers.

La régulation est mise en place par l'affectation de créneaux de décollage à certains vols lors des phases pré-tactique et tactique. Ces créneaux correspondent à une heure précise de décollage qui doit impérativement être respectée dans une fenêtre de temps de 15 minutes autour de cet horaire (de 5 minutes avant l'heure fixée à 10 minutes après). Lorsqu'un créneau est manqué, une demande pour un autre créneau doit être effectuée par le pilote.

1.3 Gestion du trafic au sol

1.3.1 Structure d'un aéroport

L'aéroport est l'unique lieu de roulage des avions. Il est donc structuré en fonction de ces derniers. On peut y trouver différentes zones qui leurs permettent d'effectuer diverses tâches :

- **Les zones de parking** : c'est une zone de manœuvres diverses telles que le push-back (sortie du parking en marche arrière) et la vitesse des avions y est donc réduite. Chaque parking est autorisé à certains types d'avions, il y a donc des règles d'accès. Il est également nécessaire de séquencer les entrées-sorties de ces zones.
- **Les pistes** : les vitesses de roulage dans ces zones sont élevées. Il y a donc également nécessité de séquencer les avions en tenant compte de ceux qui doivent traverser la piste. Certaines règles de séparation sont alors mises en place pour assurer la sécurité. L'*aire de piste* détermine une zone de sécurité dans laquelle aucun mobile ne doit être devant un décollage ou un atterrissage. Les avions accèdent à la piste via une bretelle d'accès. Sur les grands aéroports, il existe plusieurs bretelles d'accès et de libération de piste ainsi que des bretelles de libération rapide qui permettent aux avions de terminer leur freinage en ayant déjà libéré l'aire de piste.
- **Les taxiways** : ce sont les routes qui permettent d'acheminer un avion d'un point à un autre de l'aéroport. Des règles de séparation sont également de rigueur dans ces zones et les avions sont guidés par les contrôleurs. Les taxiways peuvent être à sens unique et parfois réservés à un certain type d'avions.

Dans le cas des aéroports comprenant plusieurs pistes, chaque piste peut se voir attribuer un unique type de mouvements : soit des décollages, soit des atterrissages. Cela permet de simplifier la gestion des taxiways et des pistes. Il peut arriver également qu'une piste accepte les deux types de mouvements. Les pistes réservées à un seul type sont dites "spécialisées" et celles qui acceptent les deux types en même temps sont dites "banalisées".

En fonction de plusieurs facteurs tels que la météo ou le trafic prévu, chaque aéroport possède une configuration d'exploitation différente (par exemple : face à l'est ou face à l'ouest). Chaque configuration est définie par des règles de circulation sur l'aéroport telles que les sens uniques pour des taxiways ou les voies d'accès aux parkings et les sens d'utilisation des pistes.

Lorsqu'un avion décolle ou atterrit, il crée une turbulence de sillage qui empêche un second appareil d'utiliser immédiatement la piste. Il existe trois principales catégories d'avions : Léger, Moyen et Lourd. Les règles de séparation des avions sur les pistes dépendent avant tout de la catégorie des deux avions concernés. Par exemple, un avion de catégorie légère devra attendre plus longtemps qu'un avion lourd après le décollage d'un avion moyen.

1.3.2 Contrôle d'aérodrome

Les contrôleurs ont pour rôle de gérer le trafic au sol sur les différentes zones de l'aéroport. On distingue trois zones pouvant être gérées par différents contrôleurs au sein de l'aéroport :

- **Les pistes** : les contrôleurs ont la charge du séquençage des mouvements sur la piste en respectant les règles de séparation. Ils ont la possibilité d'effectuer des actions telles qu'autoriser le décollage ou l'atterrissage d'un avion, modifier la vitesse d'un avion lors de sa descente finale afin qu'il arrive plus rapidement ou plus lentement selon la disponibilité de la piste, affecter une bretelle d'entrée ou de sortie de piste ou encore imposer une remise de gaz lorsque la piste n'est pas disponible pour une arrivée. On appelle *incursion piste* le fait qu'un appareil entre sur la piste alors qu'il n'a pas obtenu l'autorisation du contrôleur.
- **Les parkings** : les contrôleurs ont la responsabilité du séquençage des avions dans les zones de stationnement. Certaines zones de parkings pouvant être dédiées à une compagnie aérienne particulière.
- **Les taxiways** : les contrôleurs sont chargés de la surveillance du trafic au roulage et peuvent imposer un chemin à chaque avion. Ils peuvent donc donner la priorité à un avion sur une intersection ou encore ordonner l'arrêt d'un appareil. Chaque avion est pris en charge par ces contrôleurs durant toute sa phase de roulage.

Parallèlement à ces tâches, les contrôleurs ont également la charge de s'assurer du respect des créneaux de décollage imposés par le CFMU. (cf. 1.2.2) Cette gestion s'effectue à tous les niveaux de l'aéroport puisqu'il faut coordonner les trois zones : ne pas sortir trop tard ou trop tôt du parking, ne pas perdre trop de temps lors du roulage et être face à la piste à l'heure imposée.

1.3.3 Domaines d'étude pour le contrôle d'aérodrome

L'aide au contrôle consiste à mettre en œuvre des outils informatiques qui permettent de simplifier la tâche des contrôleurs. Ce type d'outils peut, à titre d'exemple, fournir la liste des avions sur les taxiways et les chemins qui leurs sont affectés, ou encore mettre à disposition la liste des avions en approche.

Les trois principaux domaines d'étude pour le contrôle d'aérodrome concernent la gestion au sol des taxiways et des parkings (Surface MANagement ou SMAN) et la gestion des pistes qui se compose de la gestion des décollages (Departure MANagement ou DMAN) et des atterrissages (Arrival MANagement ou AMAN).

La notion de **Surface MANagement (SMAN)** concerne la gestion de la phase de roulage des avions, depuis leur parking jusqu'à la piste de décollage (ou respectivement pour un avion qui atterrit depuis la sortie de la piste jusqu'à son parking). Ce domaine d'étude traite donc de l'affectation d'un chemin à chaque avion, de l'heure de départ du parking des avions et de la résolution des conflits au roulage. Les outils actuellement développés dans le domaine du SMAN permettent généralement aux contrôleurs d'avoir une vision d'ensemble des mouvements actifs sur l'aéroport à chaque instant. Ces contrôleurs peuvent alors anticiper les actions à venir et prendre des décisions en conséquence.

Les notions de **Arrival MANagement (AMAN)** et **Departure MANagement (DMAN)** sont relatives à la gestion des flux d'arrivées et de départs. Leur objectif principal est de fournir aux contrôleurs une séquence prévisionnelle d'atterrissages afin d'optimiser la gestion des arrivées. Tout comme on distingue les pistes qui n'acceptent que des décollages ou des atterrissages, on distingue également le AMAN et le DMAN (les deux domaines d'étude peuvent être traités séparément). Les outils développés dans le domaine des AMAN ont pour objectifs l'affectation des heures d'atterrissage

et le séquençage des arrivées, alors que ceux de la catégorie DMAN sont principalement élaborés pour affecter des heures de départ du parking et prévoir les heures de décollage.

À l'heure actuelle, une multitude d'outils d'aide à la décision sont déployés sur les aéroports pour faciliter le travail des contrôleurs. Ces outils sont exploités dans le cadre de l'aide à la décision et ne font en aucun cas des choix à la place du contrôleur. Ils ont donc un rôle informatif avant tout. A titre d'exemple, voici quelques outils utilisés sur l'aéroport de Roissy Charles de Gaulle :

- **AVISO** (Aide à la Visualisation au Sol) [Vallée 01] traite les données radar sur Roissy et permet la visualisation de tout appareil en mouvement aussi bien sur les taxiways que sur les pistes. Il est considéré dans la catégorie SMAN.
- **SALADIN** (Système d'Aide à LA Détection d'INcursions) [Vallée 01] est une extension d'AVISO qui permet de détecter les incursions piste et déclenche une alerte le cas échéant.
- **MAESTRO** (Moyen d'Aide à l'Écoulement Séquencé du Trafic avec Recherche d'Optimisation) [Bourdon 98] est un outil d'aide au contrôle qui gère principalement les arrivées. Il appartient donc à la catégorie AMAN. Il permet de visualiser l'écoulement du trafic à l'arrivée en temps réel. Parmi ses principaux objectifs figurent :
 - l'amélioration de la coordination entre le contrôle en route, l'approche et l'aéroport ;
 - la réduction des attentes en l'air ;
 - la gestion globale des délais.

1.4 Difficultés identifiées

La gestion des différentes phases du trafic aéroportuaire implique de pouvoir maîtriser le lien entre chacune des phases. La gestion du trafic au sol peut donc avoir une lourde influence sur le trafic en vol et occasionner des retards. Les opérations au sol sont très variées et sont sujettes à certaines incertitudes, par exemple : la vitesse de roulage des avions sur les taxiways ou encore les heures de départ du parking sont inconnues. Ces incertitudes impliquent une certaine difficulté pour anticiper le trafic au roulage et donc pour définir un DMAN pertinent.

Le contrôleur doit donc être capable de gérer tous ces imprévus tout en essayant d'optimiser le trafic au sol :

- minimiser les temps de roulage,
- fluidifier le trafic (éviter les points de congestion),
- respecter les créneaux imposés par le CFMU.

Un objectif s'impose alors pour les contrôleurs qui doivent gérer à la fois les arrivées, les départs et le cheminement des avions jusqu'à leur parking (ou jusqu'à la piste) en respectant les créneaux imposés par le CFMU : l'anticipation. En effet, les contrôleurs doivent gérer le trafic en temps réel et prendre en compte les avions qui doivent respecter un créneau dans un futur plus ou moins proche afin de s'assurer que les avions concernés pourront bien s'insérer dans le trafic et décoller dans le temps imparti. Il est donc indispensable de pouvoir anticiper le trafic sur deux horizons différents : la résolution des conflits sur les quelques minutes à venir, en considérant les avions actifs (au roulage) ; et la prise en compte des séquences de pistes sur la demi-heure ou l'heure qui suit afin de prévoir à l'avance l'ordonnancement des avions sur la piste.

Chapitre 2

Analyse préliminaire

2.1 Introduction

Le chapitre précédent décrit le cadre général dans lequel s'insère l'étude du trafic au sol. Il en ressort que certaines améliorations peuvent prétendre à la fois réduire les retards, limiter la congestion au sol et renforcer la sécurité.

Ce chapitre présente un état de l'art de différents travaux concernant l'optimisation du roulage au sol sur les grands aéroports, incluant la gestion des pistes. La seconde partie présente quelques simulateurs d'aéroports existants, dont celui qui est utilisé lors de cette thèse. Enfin, la dernière partie a pour but de souligner les points qu'il semble possible d'améliorer et qui constituent ainsi l'objectif de cette thèse.

2.2 Projets pour le contrôle d'aérodrome

De nombreux travaux traitent de la gestion du trafic aéroportuaire. Cette section en présente la partie en lien avec l'objet de cette thèse. Ces travaux peuvent se classer en trois catégories :

- **Prédiction et optimisation des temps de roulage** : cette partie concerne quelques travaux qui présentent des méthodes pour calculer, diminuer ou visualiser les temps de roulage des avions.
- **Optimisation des séquences d'avions sur les pistes** : cette section présente certaines études portant sur la gestion exclusive des pistes. Des algorithmes de calcul de séquences d'avions sur les pistes ou d'optimisation de l'attente devant la piste y sont présentés.
- **Gestion du cheminement des avions** : cette dernière partie présente un ensemble de travaux qui traitent de la résolution des conflits lors du déplacement au sol des avions. Cela inclut donc la prédiction d'un chemin pour chaque avion sur l'aéroport et la gestion du temps de roulage.

Enfin, une dernière partie présente les différents problèmes soulevés par les travaux exposés.

2.2.1 Optimisation des temps de roulage

La complexité de la phase de roulage et toutes les incertitudes qui lui sont propres font que l'étude de l'amélioration de la prédiction du trafic est récurrente dans la littérature. La diversité des aéroports et de ses acteurs (pilotes, contrôleurs, techniciens, ...) joue également un rôle dans l'impossibilité d'appliquer tel quel un algorithme étalonné sur un aéroport particulier, aussi générale que puisse-t-êre la méthode de calcul. La prédiction du trafic est donc un point clé pour pouvoir ensuite optimiser les temps de roulage.

Ces temps peuvent être obtenus par une multitude de méthodes de par la diversité des informations disponibles. Ainsi, on peut simplement comparer et lier les données fournies par différentes sources (telles que les compagnies aériennes, les traces radars ou tout autre organisme pouvant fournir des données pertinentes) [Legge 08]. Ces informations peuvent être mises à disposition de certains organismes dans le cadre du *Collaborative Decision Making (CDM)*, qui consiste à faire travailler ensemble des personnels de la navigation aérienne et des compagnies aériennes [FAA 10] : cette mise en commun des informations en temps réel améliore nettement leur pertinence et permet de prendre des décisions plus adaptées aux situations de trafic à venir.

Il est également possible de calculer une estimation des temps de roulage par des algorithmes plus complexes tels qu'un processus de décision de Markov utilisant une méthode d'apprentissage. Cet apprentissage se base sur l'état courant du trafic et fournit une note pour chaque temps de roulage prédit en le comparant avec la réalité. Une fois étalonné, l'algorithme est capable de prédire un temps de roulage en fonction d'une situation donnée avec une faible marge d'erreur [Balakrishna 09]. Ce type d'algorithme ne peut en revanche calculer correctement des temps de roulage dans les situations non encore expérimentées (nouvelles configurations, modification de voies de circulation, nouvelles procédures d'exploitation, etc.)

Pour être plus évolutive, le modèle de prédiction peut également utiliser une méthode de type *Fix & Relax* [De Pablo 05] qui explore l'arbre des prévisions de trafic, contenant les heures de passage ou d'attente de chaque avion en un point de l'aéroport pour calculer les meilleurs temps de roulage possibles. L'arbre des prévisions est dynamique et s'adapte à chaque situation de trafic, mais les calculs restent relatifs à un modèle statistique calibré sur le trafic réel et ne sont donc valables que sur un aéroport précis, dans des situations de trafic connues.

L'optimisation des temps de roulage peut s'effectuer de façon plus approfondie, en fonction de chaque étape pouvant être améliorée. On peut par exemple s'intéresser à l'affectation des parkings et à ses influences sur la congestion des zones de stationnement et sur le flux de passagers [Kim 10]. On peut également optimiser la phase de dégivrage des avions afin de réduire les attentes, en optimisant l'organisation des camions de dégivrage (en prenant en compte les caractéristiques de chaque camion et le temps qu'il faudra pour dégivrer chaque avion) [Norin 09]. Plus globalement, le flux d'avions au départ peut être modélisé et optimisé, par une meilleure gestion des heures de libération du parking et des aires d'attente devant les pistes [Burgain 08, Malika 10].

Il est également possible de réduire les délais en améliorant la coordination entre les contrôleurs, grâce à de nouvelles interfaces contenant des données plus pertinentes et permettant une communication directe entre les contrôleurs [Doble 09].

On peut envisager d'utiliser des données fournies par un AMAN (Arrival MANagement, cf. 1.3.3) au sein d'un SMAN (Surface MANagement, cf. 1.3.3) en se basant sur des règles de logique floue [Böhme 07]. Ces règles permettent de prendre une décision en fonction de différents critères jugés importants ou non tels que l'efficacité d'une action, la priorité d'un avion ou encore la probabilité d'utilisation d'une voie d'accès. Il en va de même pour l'intégration de données d'un DMAN (Departure MANagement, cf. 1.3.3) dans un SMAN en utilisant un concept de *Hot Spot* qui consiste à décider quelle séquence est la plus pertinente entre celle fournie par le DMAN et celle fournie par le SMAN, à une intersection donnée de l'aéroport. Cette décision s'effectue en comparant plusieurs solutions possibles en parallèle à chaque étape. L'objectif des liens entre AMAN-SMAN ou DMAN-SMAN est de pouvoir mieux anticiper le trafic.

Globalement, ces modèles sont prometteurs et donnent des résultats statistiquement satisfaisants, mais la notion de conflits entre avions aux abords des zones de stationnement et des intersections n'est pas prise en compte précisément et peut entraîner localement des différences importantes entre les temps de roulage prévus et les temps de roulage effectifs.

2.2.2 Optimisation des séquences d'avions sur les pistes

La majorité des systèmes actuels utilisent la méthode premier arrivé, premier servi (FCFS : First Come, First Served) pour la gestion des pistes. Il s'avère que l'optimisation des séquences d'avions sur les pistes est de plus en plus à l'étude car il en ressort une nette amélioration de l'utilisation des pistes lorsque le flot d'avions y accédant est correctement agencé.

Cette optimisation implique donc qu'il faut classer les avions pour déterminer leur heure d'utilisation des pistes. On peut alors utiliser des méthodes telles que la recherche *tabou* qui parcourt un ensemble de solutions possibles en tenant compte des mouvements récents sur l'aéroport. Ces solutions définissent les positions d'attente des avions devant la piste ainsi que le chemin emprunté dans les zones d'attente [Atkin 08].

Il est également possible de découper le problème d'ordonnancement des avions en deux sous-problèmes : la gestion de l'attente devant les pistes et le calcul des séquences. Les séquences de pistes sont optimisées dans une première étape. La faisabilité des séquences est alors vérifiée dans la seconde étape qui consiste à minimiser les temps d'attente devant les pistes en optimisant les chemins empruntés par les avions dans les zones d'attente [Atkin 04].

Les séquences de pistes peuvent être calculées en se basant sur différents critères. On peut par exemple chercher la séquence qui respecte au mieux les horaires prévus afin de satisfaire les passagers, celle qui réduit les temps d'attente des avions ou encore celle qui respecte le plus possible les créneaux imposés par le CFMU (cf. 1.2.2). Certains critères pouvant être en opposition, il peut s'avérer difficile de tous les optimiser en même temps. Une pondération entre ces critères peut être mise en place afin de conserver une certaine flexibilité et de pouvoir définir leur ordre d'importance. Il est alors possible d'optimiser la séquence en se basant sur le fonctionnement en FCFS et en effectuant un recuit simulé qui utilise ces pondérations [Pick 07].

2.2.3 Gestion du cheminement des avions

Ce domaine d'étude est probablement le plus complexe des trois. Il consiste à affecter une trajectoire (un chemin et un ou plusieurs points d'attente) à chaque avion sur l'aéroport pour le mener à sa destination et résoudre les conflits lors du roulage. Il faut donc prendre en compte de nombreux paramètres tels que la vitesse de roulage, les sens de circulation, le nombre d'avions en circulation ou encore la position de chacun.

Le calcul des chemins de chaque avion demande la mise en œuvre d'algorithmes complexes qui prennent en compte les divers paramètres de la situation et le nombre plus ou moins grand d'avions en mouvement sur l'aéroport. Il est possible de résoudre ce problème en utilisant des algorithmes de programmation par contraintes [Rathinam 08]. Le critère à optimiser est le temps total de roulage et les variables du problème sont les heures de passage de chaque avion à chaque intersection ainsi que l'ordre des avions. Des contraintes sur les sens de circulation, sur les règles de séparation des avions ou encore sur la priorité des avions sont utilisées. Cette approche permet une diminution significative des délais sur l'aéroport par rapport à une gestion de type premier arrivé, premier servi, mais ne prend pas en compte les incertitudes sur les heures de départ du parking et les vitesses de roulage, ni les problèmes de congestion au niveau des zones de stationnement.

Il est possible d'inclure d'autres phases que le roulage dans la résolution, telles que la phase de push-back (sortie de l'avion de son parking en marche arrière) [Balakrishnan 07b]. En contrôlant l'heure du push-back d'un avion, on peut s'assurer de son heure de départ du parking et ainsi de son temps de passage à certaines intersections. Cela permet de limiter les conflits à résoudre, mais n'assure pas une répartition optimale des avions sur les voies de circulation.

Les méthodes de MIP (Mixed Integer Programming) [Smeltink 05] ou MILP (Mixed Integer Linear Programming) [Roling 08] peuvent donc être utilisées dans le cadre de l'optimisation de la phase de roulage des avions, mais nécessitent en général une simplification du problème de résolution de conflits à résoudre.

Dans l'objectif de trouver des solutions plus globales et plus précises, la résolution des conflits peut alors s'effectuer grâce à des algorithmes évolutionnaires [Gotteland 04]. Ces algorithmes n'assurent pas de trouver une solution optimale mais permettent de prendre en compte les situations de trafic de façon plus réaliste en un temps acceptable, en fonction de paramètres comme l'horizon de prédiction, le pas de rafraîchissement du système ou encore les incertitudes sur les vitesses de roulage.

2.2.4 Problèmes posés

Dans la littérature de nombreux articles traitent de la prédiction des temps de roulage. Le problème est que ce n'est qu'une estimation de ces temps (plus ou moins précise) et qu'elle n'est pas toujours optimisée.

L'étude des séquences de pistes est en plein essor car l'optimisation de ces séquences permet de diminuer significativement les délais. Cependant, le calcul de ces séquences se fait principalement indépendamment de la résolution des conflits au sol. Le lien entre la gestion des pistes et la gestion du roulage est rarement modélisée et reste très complexe car il faut prendre un grand nombre de paramètres en compte et s'adapter à chaque horizon de prédiction (très court pour la résolution des conflits au roulage et plus long pour le calcul des séquences de pistes).

Il apparaît également qu'il subsiste une grande incertitude dans certains paramètres de la gestion du roulage. La plupart des modèles fixent alors leur valeur (la vitesse de roulage des avions par exemple) afin de pouvoir résoudre les situations de trafic. De ce fait, les systèmes proposés peuvent s'avérer inexploitable en temps réel.

2.3 Simulateurs de trafic aéroportuaire

2.3.1 Objectifs

Les simulateurs d'aéroport peuvent avoir différents objectifs. Certains ont pour but d'analyser la quantité de trafic en un point précis de l'aéroport, d'autres étudient les conséquences d'une modification topographique de l'aéroport. Ils ont en revanche tous une base commune qui consiste à reproduire le trafic aéroportuaire en modélisant ses contraintes réelles. Lorsque la modélisation est suffisamment détaillée, ces simulateurs permettent d'étudier les influences de certains facteurs, comme par exemple des modifications topologiques telles que l'ajout d'un taxiway, de parkings, la fermeture d'une piste ou des modifications moins concrètes telles que donner la priorité à une catégorie d'avion, modifier une trajectoire ou encore changer le processus de dégivrage.

Un simulateur permet donc d'analyser des possibilités de scénarios réalisables afin d'en percevoir à l'avance les effets sur la réalité et de prédire comment améliorer la gestion de l'aéroport sans en perturber le fonctionnement.

2.3.2 Exemples de simulateurs

Il existe actuellement quelques simulateurs qui sont utilisés ou en cours de développement. On peut citer par exemple :

- **TAAM** (Total Airspace and Airport Modeler) [Curtis Holden 03] développé par Preston Aviation Solution. C'est un simulateur *gate-to-gate* qui permet de modéliser un vol depuis son départ du parking de son aéroport d'origine jusqu'à son arrivé au parking sur l'aéroport de destination.
- **SIMMOD** (SIMulation MODel) [FAA 02] est également un simulateur *gate-to-gate* mis en œuvre par la FAA (*Federal Aviation Administration*) qui met à disposition de l'utilisateur des informations telles que la consommation de fuel ou les temps de vol.
- **CATS** (CAML All purpose Traffic Simulator) - **OPAS** (Outil de Planification ATM et de Simulation) [DNSA/DTI 07] est en cours de développement par la DSNA/DTI (Direction des Services de la Navigation Aérienne / Direction de la Technique et de l'Innovation). Il possède de nombreux algorithmes de détection et de résolution de conflits. Il existe plusieurs outils dans OPAS, notamment OPAS-SOL qui concerne la gestion du roulage des avions.
- **ATOS** (Airport Traffic Optimization Simulator) [DNSA/DTI/R&D 07] est développé par le domaine R&D de la DSNA/DTI. C'est un simulateur de trafic au sol qui possède trois méthodes de résolution de conflits : une méthode déterministe et deux méthodes par algorithmes évolutionnaires.

2.4 Cadre de travail de la thèse

2.4.1 Questions posées

Comme cela à été mentionné dans la section 2.2, les algorithmes déterministes fixent souvent des paramètres qui sont en réalité incertains. Le simulateur ATOS qui sera utilisé dans cette thèse permet de considérer ces paramètres avec une certaine marge d'erreur et de résoudre les situations de trafic en les intégrant au problème de façon réaliste. ATOS n'a en revanche pas de gestion anticipée des séquences de pistes. L'objectif de cette thèse est donc d'intégrer la gestion des séquences de pistes dans ce simulateur et de lier ces séquences à la résolution des conflits au roulage. Cette opération devrait permettre d'anticiper de façon pertinente le trafic et de réduire les délais générés.

2.4.2 Ordonnancement des avions sur les pistes

Afin d'optimiser le débit d'avions de l'aéroport, les pistes sont souvent utilisées à leur maximum. Comme il a été mentionné dans la section 1.3.1, des règles de séparation doivent être respectées et chaque avion doit attendre un certain temps pour pouvoir utiliser la piste après un autre avion. Un temps d'attente conséquent entre deux avions subsiste alors durant l'utilisation des pistes. Ce temps peut être réduit si le séquençage des avions sur les pistes est optimisé. Un meilleur ordonnancement des avions sur les pistes est donc à l'étude en vue de diminuer les délais et de respecter les créneaux imposés par le CFMU.

2.4.3 Lien entre gestion des pistes et gestion du roulage

L'attente des avions devant la piste avant le décollage pose un véritable problème car la file d'attente peut s'étendre sur les taxiways et bloquer la circulation. Il en va de même lorsqu'un avion qui vient d'atterrir doit traverser une piste. Afin de respecter les créneaux imposés par le CFMU et les séquences de pistes lorsqu'elles sont calculées, on doit également s'assurer de l'ordre d'arrivée des avions devant la piste pour qu'ils ne se bloquent pas l'accès. On constate alors un étroit lien entre l'optimisation de l'utilisation des pistes et la gestion de la phase de roulage. L'idée serait donc de limiter ces files d'attente en modifiant les chemins et les attentes des avions en fonction des séquences

d'avions prévues sur chaque piste. En effet, la politique de fonctionnement qui consiste à acheminer les avions le plus rapidement possible vers la piste ne porte pas ses fruits lors des heures de pointe et entraîne la congestion du trafic. L'étude est donc portée sur le fait de faire attendre les avions en amont et de les ordonnancer dès leur roulage sur les taxiways pour qu'ils arrivent en séquence et à l'heure souhaitée devant les pistes.

Deuxième partie

Séquencement des avions sur les pistes

Chapitre 3

Définition du problème

3.1 Introduction

L'objectif principal de l'étude des séquences de pistes sur les grands aéroports est de parvenir à les anticiper (entre 20 et 60 minutes à l'avance) et de les optimiser, puis de comparer les délais obtenus avec ceux du fonctionnement « premier arrivé, premier servi » ou des séquences réelles : cela permet de quantifier la variation des délais en fonction de la stratégie utilisée et des paramètres de la résolution. Ce chapitre a pour but de formuler le problème et d'expliquer sa modélisation.

La première partie présente une analyse préliminaire du problème tel qu'il se pose aux opérationnels et met en évidence ses principales contraintes. La seconde partie en déduit une modélisation, en formalisant les variables, les contraintes et un critère d'optimisation. Enfin, une analyse de la taille de l'espace de recherche en fonction des paramètres fournit une première estimation de la complexité du problème.

3.2 Analyse préliminaire

3.2.1 Gestion opérationnelle actuelle

L'expertise des contrôleurs révèle que les stratégies d'exploitation des pistes pour les départs dépendent de la densité du trafic : en période creuse, une méthode de fonctionnement de type premier arrivé, premier servi est utilisée pour les départs non régulés et ne semble pas problématique. Lors d'une période chargée, une « réserve » d'avions en attente de décollage se forme naturellement devant la piste. Ces files d'attente sont stratégiquement réparties sur différentes bretelles d'accès en fonction des catégories des avions ou des créneaux à respecter. Les séquences de pistes sont donc construites efficacement en temps réel mais ne sont pas toujours optimales.

Sur les grands aéroports, la gestion des arrivées est réalisée par un AMAN (cf. 1.3.3) et l'ordre des arrivées n'est donc pas modifiable par le contrôle d'aérodrome, hormis en cas très particulier de remise de gaz. En revanche, les heures d'atterrissages peuvent être légèrement modifiées puisque des ordres de changement de vitesse ou de sortie plus ou moins anticipée des volets peuvent être donnés aux avions. Si ces décisions sont prises suffisamment tôt, il est possible d'avancer ou de retarder d'environ trente secondes l'heure d'atterrissage initialement prévue.

Données en entrée	
Notation	Description
N	Nombre total d'avions.
\mathcal{A}	Ensemble des arrivées.
T_{att}^i	Heure d'atterrissage de l'avion i prévue par le AMAN.
δ_{att}	Temps de décalage possible pour les heures d'atterrissage.
\mathcal{D}	Ensemble des départs.
\mathcal{D}_C	Ensemble des départs non régulés.
\mathcal{D}_R	Ensemble des départs régulés.
T_{cfmu}^i	Heure de décollage imposée par le CFMU pour l'avion i .
T_{pmin}^i	Heure minimale de sortie du parking de l'avion i .
D_{rmin}^i	Durée minimale de roulage de l'avion i (sans attente et à vitesse maximale).
T_{min}^i	Heure minimale d'accès à la piste pour l'avion i .
T_{max}^i	Heure maximale d'accès à la piste pour l'avion i .
$Sep_{i/j}$	Temps de séparation minimal entre l'avion i et l'avion j sur la piste.
Sep_{max}	Temps de séparation maximal entre deux avions sur la piste.
H_p	Horizon de prédiction limitant le nombre d'avions pour la résolution.
Δ_p	Pas de décalage de la fenêtre glissante.
M_{att}	Marge (en temps) derrière un atterrissage dans les séquences calculées.
ξ	Espace de recherche.
Variables	
Notation	Description
t_i	Heure prévue d'accès à la piste de l'avion i .

TAB. 3.1 – Table des notations

3.2.2 Heures minimum d'accès à la piste

Afin de pouvoir ordonnancer le passage des avions sur les pistes, il est naturellement indispensable de savoir à partir de quel moment ils pourront effectivement y accéder.

Les avions qui arrivent sur l'aérodrome dans la demi-heure à venir sont en vol et sont pour cette raison beaucoup moins sujets aux incertitudes temporelles que les avions au départ. Un AMAN peut calculer des heures précises d'atterrissage lors de la demi-heure qui précède. L'heure d'accès minimum à la piste des arrivées est donc disponible et relativement fiable.

Le cas des départs est plus délicat à gérer du fait des incertitudes sur l'heure de départ du parking et sur la vitesse de roulage. Ces incertitudes se répercutent inévitablement sur la prévision des heures de décollage. Dans tous les cas, une heure minimale d'accès à la piste peut être calculée, en considérant l'heure de sortie du parking au plus tôt associée au temps de roulage minimal (calculé avec la vitesse de roulage maximale, sans attente) : l'avion ne pourra donc pas atteindre la piste avant cette heure minimale.

3.2.3 Règles de séparation

Dans le manuel d'exploitation de l'aéroport, les règles de séparation entre deux avions sur les pistes sont exprimées en temps et en distances. Elles sont extrêmement détaillées et dépendent de

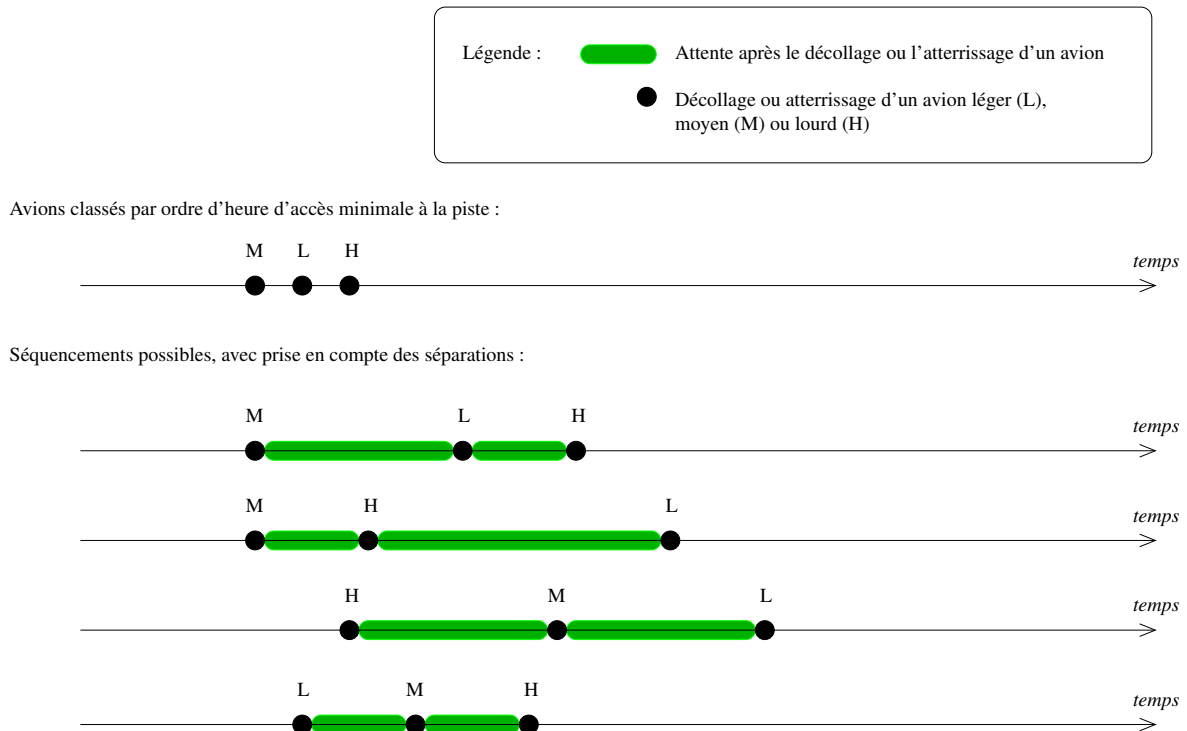


FIG. 3.1 – Séparations entre avions sur la piste

nombreux facteurs tels que les conditions météorologiques, les bretelles d'accès utilisées, les types d'avions et de moteurs impliqués, la SID (*Standart Instrument Departure*) de départ, ou encore la configuration de l'aérodrome [DSNA 08].

Dans le cadre des simulations, toutes ces informations ne sont pas toujours disponibles ni même possibles à intégrer. Les séparations sont donc modélisées de façon plus générale comme un temps de séparation minimum entre deux avions en fonction de leur catégorie de turbulence de sillage (léger, moyen ou lourd). Ce modèle est applicable tant que ces temps de séparation minimaux restent au moins aussi pénalisants que ceux correspondant aux règles de séparation réelles. Toutes les règles précédemment citées peuvent ainsi s'exprimer en temps de séparation. La figure 3.1 illustre l'influence de l'ordre dans lequel trois avions de différentes catégories sont séquencés sur la piste : l'ordre premier arrivé, premier servi est souvent bien plus pénalisant que l'ordre optimal (dernière ligne sur la figure).

En notant t_1 l'heure de décollage ou d'atterrissage d'un premier avion et $Sep_{1/2}$ le temps de séparation nécessaire avec l'avion suivant, l'heure t_2 d'accès à la piste du second avion doit vérifier :

$$t_2 \geq t_1 + Sep_{1/2}$$

3.2.4 Gestion des créneaux imposés par le CFMU

Comme introduit dans la section 1.2.2, une partie des avions au départ se voit imposer un horaire précis de décollage, noté T_{cfmu}^i (pour un avion i). Ces départs doivent décoller entre cinq minutes avant et dix minutes après l'heure fournie, comme illustré figure 3.2.

Lors de la phase de programmation, il est préférable d'éviter de prévoir le décollage d'un départ régulé vers la fin de son créneau afin de s'assurer que ce créneau sera respecté, même en cas de retard additionnel.

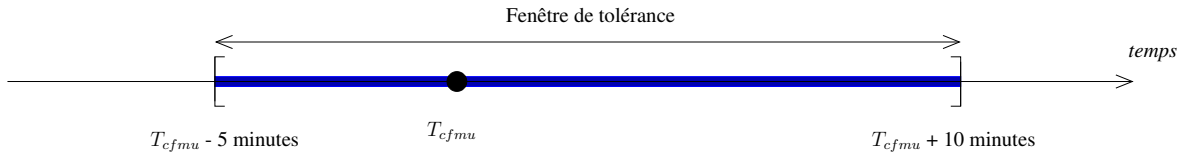


FIG. 3.2 – Fenêtre de tolérance d'un créneau imposé par le CFMU

Dans la réalité, certaines situations de trafic ne permettent pas de respecter tous les créneaux imposés et quelques avions décollent finalement en dehors de leur créneau initial.

Dans le cadre d'une simulation, un avion ne décollera jamais avant le début de son créneau puisqu'il sera toujours possible de le faire attendre au préalable. La situation de trafic pourra néanmoins aboutir au non-respect de certains créneaux, notamment en cas d'heure de sortie du parking tardive.

3.3 Modélisation du problème

3.3.1 Présentation générale

L'optimisation des séquences de pistes consiste donc à classer les avions, ce qui correspond à un problème d'ordonnancement sous contraintes, en optimisant un critère particulier. Cette section a pour but de définir toutes ces notions.

Le choix des variables est simple mais la frontière entre le critère et les contraintes est plus ambiguë. Par exemple, les créneaux imposés par le CFMU ne peuvent pas toujours être respectés. On ne peut donc pas modéliser ces créneaux comme des contraintes strictes, bien que le fait qu'un avion décolle avant le début de son créneau ne représente pas une solution acceptable : l'heure de début du créneau représente donc une contrainte stricte, alors que l'heure de fin doit donc plutôt être intégrée au critère à optimiser.

Il existe différentes méthodes de résolution exactes ou approchées pour résoudre de tels problèmes d'ordonnancement. Dans le cas de Roissy, les pistes de l'aéroport sont considérées indépendantes. Chaque séquence pourra donc être calculée indépendamment des autres.

3.3.2 Variables

Les variables du problème sont les heures de décollage et d'atterrissage des avions, soit :

$$(t_i)_{1 \leq i \leq N}$$

avec N le nombre d'avions.

Le domaine de ces variables est restreint car un avion i ne peut pas accéder à la piste avant son heure minimale T_{min}^i (cf. 3.2.2).

Pour un départ, on note D_{rmin}^i la durée de roulage minimale de l'avion i et T_{pmin}^i son heure de sortie du parking au plus tôt. L'ensemble des départs non-régulés (\mathcal{D}_C) vérifie alors :

$$\forall i \in \mathcal{D}_C, T_{min}^i = T_{pmin}^i + D_{rmin}^i$$

Un avion régulé ne doit pas décoller avant l'heure de début de son créneau. Si l'appareil ne peut pas être sur la piste avant cette heure à cause de son temps de roulage, alors T_{min}^i correspond au même

calcul que pour les départs non régulés (heure de sortie au plus tôt du parking, plus la durée minimale de roulage). Sinon, T_{min}^i correspond à l'heure de début du créneau :

$$\forall i \in \mathcal{D}_{\mathcal{R}}, T_{min}^i = \max(T_{cfmu}^i - 5min, T_{pmin}^i + D_{rmin}^i)$$

avec $\mathcal{D}_{\mathcal{R}}$ l'ensemble des départs régulés

Le domaine des heures d'atterrissage est plus restreint : comme mentionné précédemment, une heure d'arrivée est fixée dans la demi-heure précédente et ne peut être que très légèrement modifiée. Soient δ_{att} le temps de décalage maximal possible des atterrissages (en accélérant ou en ralentissant l'avion) et T_{att}^i l'heure d'atterrissage prévue par le AMAN pour l'avion i :

$$\forall i \in \mathcal{A} \left\{ \begin{array}{l} t_i \in [T_{min}^i; T_{max}^i] \\ T_{min}^i = T_{att}^i - \delta_{att} \\ T_{max}^i = T_{att}^i + \delta_{att} \end{array} \right.$$

avec \mathcal{A} l'ensemble des arrivées.

Le domaine des variables se résume donc par :

$$\left\{ \begin{array}{l} \forall i \in \mathcal{D}, t_i \in [T_{min}^i; +\infty[\\ \forall i \in \mathcal{A}, t_i \in [T_{min}^i; T_{max}^i] \end{array} \right.$$

avec $\mathcal{D} = \mathcal{D}_{\mathcal{C}} \cup \mathcal{D}_{\mathcal{R}}$ l'ensemble des départs.

3.3.3 Contraintes

La limite supérieure des créneaux imposés par le CFMU ne pouvant pas être considérée comme une contrainte du problème, les règles de séparation sur les pistes sont les seules restantes : pour chaque paire (i, j) d'avions, ces contraintes s'expriment par une relation logique faisant intervenir les temps de séparation minimaux entre les deux avions ($Sep_{i/j}$ si i est programmé avant j et $Sep_{j/i}$ sinon). Cependant, le classement des arrivées étant connu à l'avance et fixe, il n'est pas nécessaire de vérifier la séparation de leur ordre inverse car il est impossible qu'il soit choisi :

$$\left\{ \begin{array}{l} \forall (i, j) \in \mathcal{A}^2, i < j \Rightarrow t_i + Sep_{i/j} \leq t_j \\ \forall (i, j) \in \{1, \dots, N\}^2 - \mathcal{A}^2, i \neq j \Rightarrow t_i + Sep_{i/j} \leq t_j \text{ ou } t_j + Sep_{j/i} \leq t_i \end{array} \right.$$

Par ailleurs, l'ajout d'une marge augmentant les temps de séparation entre les avions peut être étudiée, dans le but d'améliorer la pertinence des séquences calculées malgré les incertitudes liées à la phase de roulage des avions. L'ajout de cette marge entre chaque paire de mouvements serait naturellement trop contraignante car elle impliquerait des délais aussi peu réalistes que nécessaires en période de pointe, comme illustré figure 3.3. En revanche, une cause importante de non-respect des séquences prévues provient du temps de libération de l'aire de piste par les arrivées, qui peut parfois dépasser la minute (temps de libération mesurés entre 30 et 90 secondes). Le mouvement suivant ne pouvant pas utiliser la piste tant que l'aire de piste n'est pas libérée, la séparation réglementaire initialement prévue peut s'avérer insuffisante. L'ajout d'une marge spécifique aux séparations derrière les arrivées à faible turbulence de sillage semble donc a priori une stratégie judicieuse (cf. figure 3.3). Cette marge, notée M_{att} apparaît ainsi comme un nouveau paramètre du problème et ajoute de nouvelles contraintes :

Pour chaque arrivée i à faible turbulence de sillage,

$$\forall j \in \{1..N\}, (t_i + Sep_{i/j} + M_{att} \leq t_j)$$

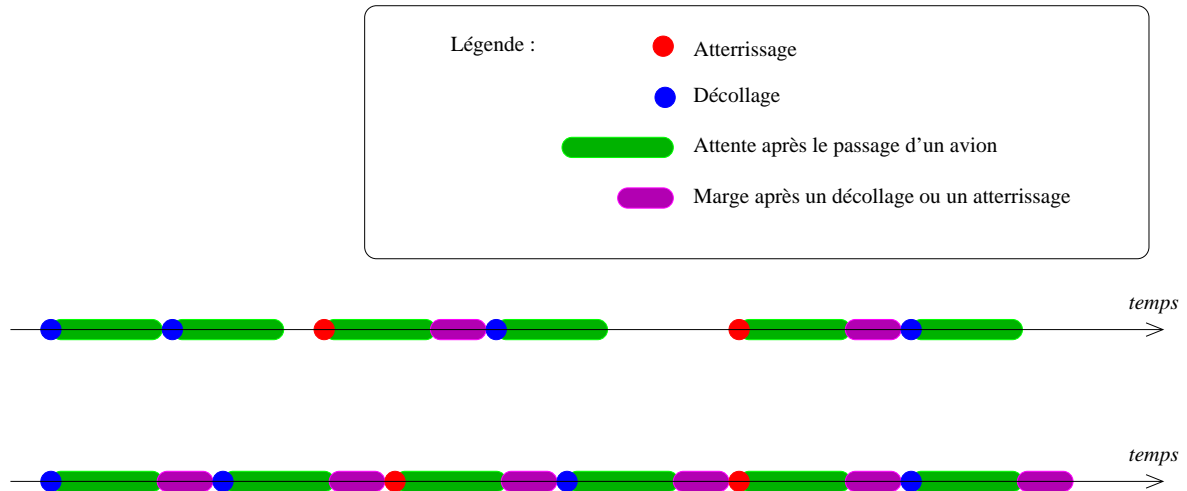


FIG. 3.3 – Insertion d'une marge dans la séquence

3.3.4 Critère d'optimisation

L'optimisation des séquences de pistes peut avoir plusieurs objectifs et peut donc faire intervenir différents critères :

- Les créneaux imposés par le CFMU doivent être respectés au mieux, en évitant autant que possible de programmer les mouvements concernés vers la fin de leur créneau.
- Le retard doit être globalement minimisé, ce qui peut s'exprimer en fonction du retard moyen, du retard maximal, et qui peut également être enrichi par une mesure d'équité entre les différents avions.

Par ailleurs, certaines questions peuvent trouver différentes réponses selon l'aéroport, le type de trafic et les stratégies d'exploitation mises en place :

- Doit-on considérer les long-courriers comme prioritaires ?
- Faut-il prendre en compte la catégorie des avions dans le critère ?
- Faut-il respecter rigoureusement les créneaux imposés par le CFMU, quitte à retarder de façon conséquente l'ensemble des décollages non régulés ?

Ces différentes considérations ont conduit à envisager le critère de façon évolutive, en le définissant globalement comme une pondération entre différentes mesures relatives aux objectifs opérationnels. Dans le cadre de cette thèse, le critère expérimenté ne concerne que les décollages, car les domaines de définition des heures d'atterrissage limitent fortement les retards occasionnés (moins d'une minute par arrivée, selon la valeur du paramètre δ_{att}). Le critère à minimiser s'exprime alors comme une pondération entre le retard des départs non-régulés et les écarts aux créneaux imposés par le CFMU :

$$f((t_i)_{1 \leq i \leq N}) = x_{dc} \sum_{i \in \mathcal{D}_C} (t_i - T_{min}^i) + x_{dr} \sum_{i \in \mathcal{D}_R} f_{cfmu}(t_i, T_{cfmu}^i)$$

avec N le nombre d'avions.

Les coefficients x_{dc} et x_{dr} permettent de donner plus ou moins d'importance aux délais des départs non régulés et aux écarts aux créneaux imposés par le CFMU.

La fonction f_{cfmu} mesure la pénalité correspondant à un écart au créneau imposé (figure 3.4) :

- Entre 5 minutes avant ou après l'heure imposée, aucune pénalité n'est retenue (respect satisfaisant du créneau).

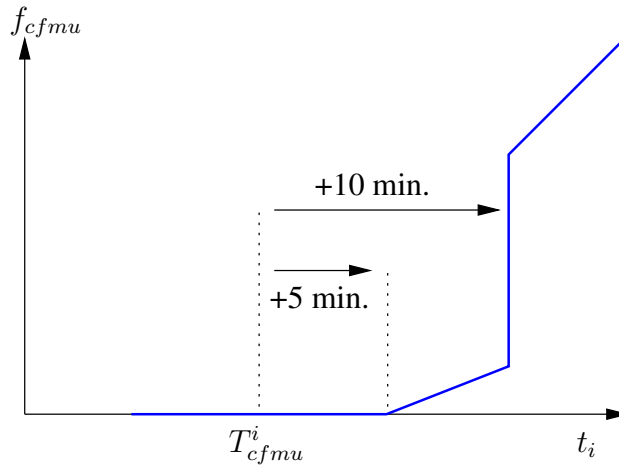


FIG. 3.4 – Critère pour les vols régulés

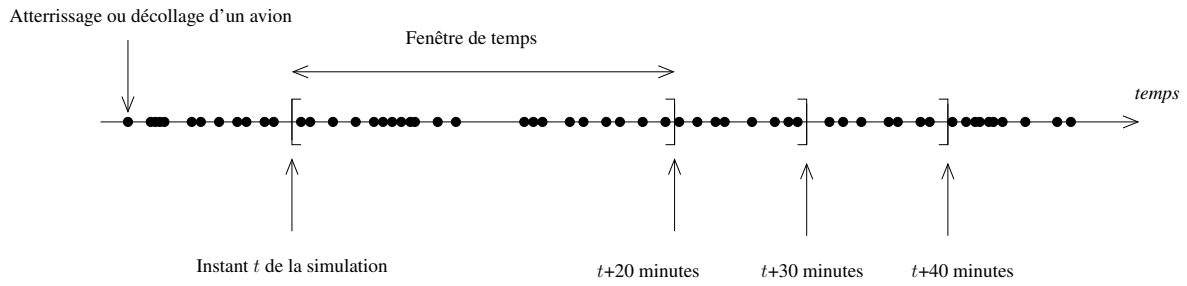


FIG. 3.5 – Modification de l'espace de recherche en fonction de l'horizon de prédiction

- Entre 5 et 10 minutes après l'heure imposée, la pénalité est croissante et modérée (car une telle programmation ne supporte aucun retard additionnel).
- Au delà du créneau, la pénalité est fortement augmentée (discontinuité de la fonction f_{cfmu}) pour rendre le respect des créneaux prioritaire par rapport aux délais des autres avions.

$$\forall i \in \mathcal{D}_{\mathcal{R}} \begin{cases} t_i \in [T_{cfmu}^i - 5min; T_{cfmu}^i + 5min] & \Rightarrow f_{cfmu} = 0 \\ t_i \in [T_{cfmu}^i + 5min; T_{cfmu}^i + 10min] & \Rightarrow f_{cfmu} = t_i - (T_{cfmu}^i + 5min) \\ t_i \in [T_{cfmu}^i + 10min; +\infty[& \Rightarrow f_{cfmu} = 4 \times (t_i - T_{cfmu}^i) \end{cases}$$

3.4 Analyse du problème

3.4.1 Paramètres

La complexité du calcul des séquences dépend de différents paramètres, dont les principaux sont :

- L'horizon de prédiction H_p , qui correspond à la taille de la fenêtre de temps étudiée. Plus la fenêtre est petite, moins il y a d'avions considérés, comme le montre la figure 3.5. Elle fait donc varier le nombre de variables du problème et la qualité de l'optimisation qui sera effectuée.
- La flexibilité des heures d'atterrissage δ_{att} , qui correspond au temps de décalage possible d'un atterrissage. Lorsque ce paramètre varie, les heures d'arrivées peuvent être plus ou moins

décalées dans la séquence. Cela affecte donc le nombre de permutations possibles entre les avions et influence directement la taille de l'espace de recherche.

- La marge M_{att} , dont l'objectif est d'améliorer la pertinence de la séquence. L'augmentation de cette marge réduit le nombre de permutations possibles car elle impose une plus grande séparation entre certains avions.

3.4.2 Espace de recherche

Les variables du problème sont les heures de chaque mouvement sur la piste. Elles sont définies sur des domaines entiers, correspondant au modèle de discrétisation du temps utilisé dans le simulateur, soient des pas de temps élémentaires de cinq secondes.

Afin de pouvoir quantifier la taille de l'espace de recherche, il faut déterminer la borne supérieure des heures à programmer. Par définition, seuls les avions dont l'heure minimale d'accès à la piste est incluse dans l'horizon de prédiction sont considérés :

$$\forall i, T_{min}^i \leq H_p$$

Un des pires cas correspond à une arrivée et un ensemble de départs ayant la même heure minimale d'accès à la piste, en fin d'horizon de prédiction H_p . Dans ce cas, la solution optimale consiste à programmer les décollages à la suite les uns des autres en respectant les règles de séparation. En notant Sep_{max} le plus grand temps de séparation pouvant être imposé entre deux avions, on obtient alors une borne supérieure des heures de décollage :

$$\forall i \in \mathcal{D}, t_i \leq H_p + |\mathcal{D}|Sep_{max}$$

Les domaines de définition des heures d'atterrissage sont des intervalles bornés d'amplitude $2\delta_{att}$. La taille totale de l'espace de recherche peut donc être majorée comme suit :

$$|\xi| \leq (H_p + |\mathcal{D}|Sep_{max})^{|\mathcal{D}|} \times (2\delta_{att})^{|\mathcal{A}|}$$

La taille de l'espace de recherche apparaît donc bien exponentielle par rapport au nombre d'avions et aucune méthode polynomiale n'est connue pour le résoudre : la résolution d'un tel problème peut engendrer des temps de calcul prohibitifs même lorsque le nombre d'avions est petit.

Une autre approche pour formaliser le problème consiste à chercher l'ordre optimal des mouvements sur la piste : les heures de décollage et d'atterrissage des avions peuvent ensuite être déduites séquentiellement, en fonction des temps de séparation et des heures minimales d'accès à la piste. L'ordre des arrivées étant initialement fixé, le problème revient à choisir l'ordre des départs, en les insérant le plus tôt possible et dans cet ordre dans la séquence des arrivées. La taille de l'espace de recherche se réduit ainsi au nombre de permutations possibles des départs :

$$|\xi| = \mathcal{O}(|\mathcal{D}|!)$$

Cette approche permet de réduire la taille de l'espace de recherche, mais par association avec les problèmes d'ordonnancement largement documentés [Woeginger 03], lorsque les contraintes ne sont pas linéaires, le séquencement des avions apparaît comme un problème NP-complet et ne peut être résolu par des méthodes exactes que pour des faibles valeurs de $|\mathcal{D}|$.

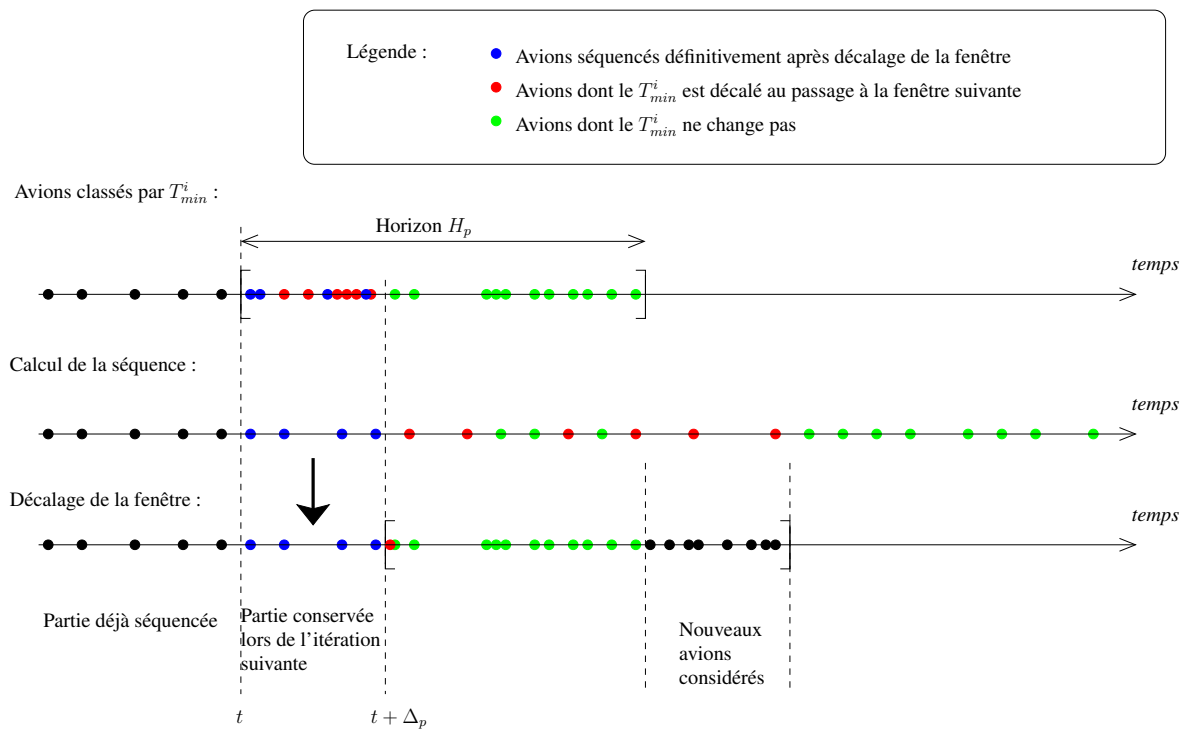


FIG. 3.6 – Fenêtre de temps glissante

3.4.3 Résolution par fenêtre glissante

La limitation du nombre d'avions par l'horizon de prédiction implique une répétition du processus pour les avions qui n'ont pas encore été considérés.

Dans ce but, un système de fenêtre glissante est mis en place : en notant Δ_p le pas de décalage de la fenêtre glissante ($\Delta_p \leq H_p$), la séquence est calculée à un instant t_0 sur l'horizon H_p , puis à l'instant $t_1 = t_0 + \Delta_p$ (à nouveau sur l'horizon H_p). Les heures de décollage ou d'atterrissage des avions prévues dans l'intervalle de temps $[t_0; t_1]$ sont alors considérées figées et ne peuvent plus être modifiées. Le dernier mouvement programmé avant t_1 devient alors une contrainte forte pour le premier avion considéré dans la nouvelle situation, comme l'illustre la figure 3.6.

Le processus est alors réitéré pour $t_2 = t_1 + \Delta_p$ et ainsi de suite.

3.4.4 Mesures sur le trafic réel

Durant la journée du 22 mars 2002, l'aéroport de Roissy Charles de Gaulle a été utilisé par 1454 avions. Parmi ces derniers, on dénombre 735 départs, dont 278 régulés, et 719 arrivées. Ces vols sont répartis sur les pistes comme suit :

- Piste 27 (banalisée) : 313 départs, dont 78 régulés, et 299 arrivées,
- Piste 26R : 422 départs, dont 200 régulés,
- Piste 26L : 420 arrivées.

Les figures 3.7 et 3.8 présentent le nombre maximal d'avions programmés sur chaque piste en fonction de l'horizon de prédiction. Ces mesures sont extraites directement des plans de vol déposés (demandes des compagnies aériennes) et ne tiennent donc pas compte de l'effet d'accumulation pouvant survenir suite aux retards des avions. Dans ce cadre, on mesure une augmentation linéaire du

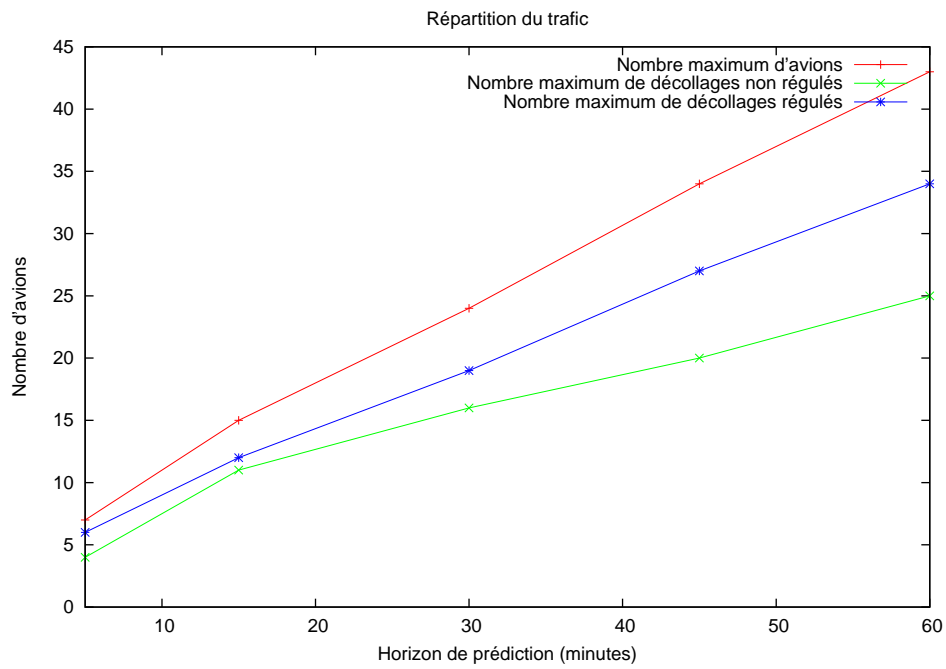


FIG. 3.7 – Nombre maximal d'avions en fonction de la taille de l'horizon, sur la piste 26R

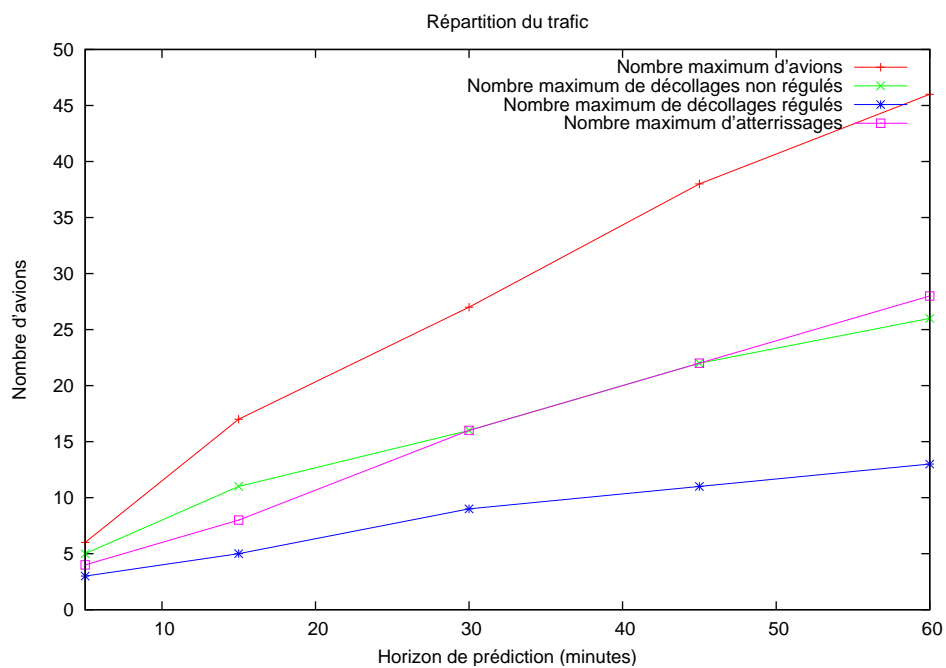


FIG. 3.8 – Nombre maximal d'avions en fonction de la taille de l'horizon, sur la piste 27

nombre d'avions en fonction de la taille de l'horizon de prédiction (de l'ordre de 45 mouvements maximum programmés par heure sur les deux pistes) et on peut remarquer qu'il y a un peu plus de départs régulés que de départs non régulés sur la piste 26R spécialisée, alors qu'il y en a beaucoup moins sur la piste 27 banalisée. On constate également un nombre de décollages équivalent au nombre d'atterrissages sur cette dernière.

Les problèmes de séquences de pistes qui se posent à chaque période de la journée sont de nature variée, faisant intervenir des proportions variables d'atterrissages, de décollages régulés et de décollages non régulés.

3.5 Conclusion

L'optimisation des séquences de pistes sur un aéroport comme Roissy est un problème opérationnel devant être résolu en temps réel. Ce problème se formule assez naturellement comme un problème d'ordonnancement avec des contraintes de séparation en temps. Ce problème est cependant fortement combinatoire avec le nombre d'avions impliqués et ne peut donc pas être résolu directement pour le trafic d'une journée entière.

En revanche, une approche par fenêtres glissantes doit permettre de décomposer le problème global sur chaque piste en sous-problèmes de tailles abordables (correspondant à des périodes de temps inférieures à une heure) pour finalement calculer des séquences de pistes complètes (comportant tous les mouvements prévus sur la journée) et optimisées sur chaque période.

Par ailleurs, la diversité des critères envisageables impose que la pondération choisie reste évolutive et que la méthode de résolution à définir doit éviter d'utiliser des propriétés propres à un choix particulier de critère.

Chapitre 4

Calcul d'une séquence d'avions sur une piste

4.1 Introduction

La formulation du problème lors du précédent chapitre identifie l'ordonnement des avions sur les pistes comme un problème NP-complet. Il existe diverses méthodes pour résoudre de tels problèmes, chacune ayant ses spécificités et ses domaines d'application.

La première section de ce chapitre présente une partie de ces méthodes, dans le but de déterminer celles qui semblent les plus adaptées. Des algorithmes standards existent pour la résolution de tels problèmes mais ne permettent pas toujours d'utiliser leurs propriétés spécifiques. Les solutions trouvées par de tels algorithmes ne sont donc pas toujours optimales (cela dépend de la formulation du problème) et le temps de calcul peut être conséquent. Dans ce contexte, l'utilisation d'une librairie existante est exposée, puis une dernière partie présente un algorithme spécifique pour résoudre le problème.

4.2 Méthodes de résolution envisageables

4.2.1 Algorithmes gloutons

Les algorithmes gloutons consistent à trouver une solution à un problème pas à pas en faisant le choix d'un optimum local à chaque étape. En général, ce choix est définitif et l'algorithme s'arrête dès qu'une solution est obtenue : cette dernière n'est donc pas toujours optimale. De ce fait, ce type d'algorithme est plus souvent utilisé comme *heuristique* (appelée alors *heuristique gloutonne*), fournissant une évaluation rapide d'une sous-partie de l'espace de recherche pour orienter ou limiter l'exploration d'un algorithme cherchant un optimum global.

Lorsqu'un algorithme glouton ne trouve pas de solution acceptable, un retour en arrière (dit *back-track*) est possible pour trouver une solution admissible : cela consiste à reprendre la résolution à une étape précédente et à effectuer un autre choix.

Dans l'optimisation des séquences d'avions par exemple, chaque étape peut correspondre à l'affectation de l'heure de décollage (ou d'atterrissage) d'un avion. Dans un tel algorithme, la séquence d'avions obtenue pourra correspondre à une séquence de type premier arrivé, premier servi. Le premier avion aura donc son heure minimale d'accès à la piste :

$$t_1 = T_{min}^1$$

Le second se verra affecter son heure minimale d'accès, ou l'heure de décollage du premier additionnée du temps de séparation si cela s'avère nécessaire :

$$t_2 = \max(T_{min}^2, t_1 + Sep_{1/2})$$

En étendant ce calcul à chaque avion, on obtient une relation de récurrence :

$$\begin{cases} t_1 = T_{min}^1 \\ \forall i \in [2; N], t_i = \max(T_{min}^i, t_{i-1} + Sep_{i-1/i}) \end{cases}$$

Dans le cas d'une arrivée (dont l'heure d'atterrissage est bornée), il se peut que cette définition des $(t_i)_{1 \leq i \leq N}$ ne soit pas admissible ($T_{max}^i < t_i$) : dans ce cas, l'algorithme glouton doit revenir à l'étape précédente (calcul de t_{i-1}), en remontant le rang de l'arrivée d'une position dans le classement.

Ce type d'algorithme permet ainsi d'obtenir rapidement une solution de type premier arrivé premier servi, mais ne convient pas pour trouver une séquence optimale.

4.2.2 Programmation dynamique

La programmation dynamique se base sur le principe de la décomposition d'un problème en plusieurs sous-problèmes : ce type d'algorithme part du postulat que la résolution optimale de chaque sous-problème amène à une solution globale optimale. Il est donc impossible de résoudre des problèmes qui ne peuvent pas être décomposés avec une telle méthode.

Chaque sous-problème pouvant être lui même divisé en plusieurs problèmes, on obtient un arbre de problèmes, dont les feuilles représentent les problèmes les plus simples et la racine le problème global. Dans la plupart des cas, certains problèmes ont dans leur décomposition un sous-problème commun.

Pour ne pas avoir à résoudre plusieurs fois les mêmes sous-problèmes, il convient donc de mémoriser le résultat de chaque sous-problème déjà résolu, dans des tableaux de mémorisation par exemple. Pour cette raison, la programmation dynamique est utilisée la plupart du temps sous la forme impérative et non récursive, afin d'éviter plusieurs appels récursifs non désirables sur un même sous-problème. [Rust 06]

Le calcul de la suite de Fibonacci est un exemple très utilisé pour illustrer l'intérêt de la récursivité en programmation dynamique. Cette suite est définie par récurrence comme suit :

$$\begin{cases} \mathcal{F}(1) = 1 \\ \mathcal{F}(2) = 1 \\ \mathcal{F}(n) = \mathcal{F}(n-1) + \mathcal{F}(n-2), \forall n > 2 \end{cases}$$

L'algorithme II.1 permet de calculer efficacement les éléments de cette suite, en utilisant un tableau *Fibo* mémorisant chaque valeur calculée : en effet, le problème consistant à calculer la valeur $Fibo[n]$ peut se décomposer en deux sous-problèmes : calculer $Fibo[n-1]$ et $Fibo[n-2]$. On obtient ainsi un arbre très redondant de problèmes ($Fibo[n-2]$ sera réutilisé pour trouver $Fibo[n-1]$) et les feuilles sont les valeurs connues $Fibo[2] = 1$ et $Fibo[1] = 1$. L'utilisation d'un tableau mémorisant chaque valeur permet donc d'éviter de calculer un nombre exponentiel de fois chaque valeur.

Dans le cadre des séquences d'avions sur les pistes, la décomposition du problème est difficilement envisageable : même si la considération de séquences partielles permettrait de réduire considérablement l'espace de recherche (diminution du nombre d'avions), la fusion de deux séquences partielles (même optimales) est un problème aussi complexe que le problème initial (l'ajout d'un seul mouvement à une séquence peut totalement changer l'ordre optimal des avions.)

La programmation dynamique ne semble donc pas appropriée à notre cas.

```

Fibonacci( $n$ ):
  soit Fibo : Tableau de  $n$  éléments
  Fibo[1] ← 1
  Fibo[2] ← 1
  pour  $i$  de 3 à  $n$ , faire
  [ Fibo[ $i$ ] ← Fibo[ $i$  - 1] + Fibo[ $i$  - 2]
  retourner Fibo[ $n$ ]

```

Algorithme II.1: Fibonacci : calcul de la suite de Fibonacci par programmation dynamique.

4.2.3 Programmation linéaire

La programmation linéaire consiste à maximiser une fonction linéaire $\mathcal{L}(\mathcal{X}) = {}^t C \mathcal{X} = \sum_{i=1}^n c_i x_i$ avec $0 \leq \mathcal{X}$ et un ensemble de contraintes linéaires : $A\mathcal{X} \leq B$.

D'un point de vue géométrique, les contraintes linéaires forment un polytope dans l'espace de recherche. Les optima locaux sont toujours atteints sur un sommet de ce polytope (ils peuvent s'étendre sur les arêtes ou même les faces dans certains cas). L'objectif est donc de trouver un sommet du polytope qui soit un optimum global.

Il existe de nombreux algorithmes de résolution de problèmes de programmation linéaire, dont les principaux sont :

- L'algorithme du simplexe [Talbot 08] : l'exploration commence par une solution admissible correspondant à un sommet du polytope puis continue de sommets en sommets voisins, en passant donc d'un optimum local à un autre qui améliore la fonction objectif. S'il existe plusieurs sommets adjacents qui améliorent la fonction objectif, une fonction appelée *règle de pivotage* (définie au cas par cas) en choisit un. Lorsqu'il n'existe plus de sommet adjacent qui améliore l'objectif, le sommet courant est un optimum global.
- La méthode du point intérieur [Gondzio 01] : cette méthode consiste à partir d'un point intérieur au polytope (généralement proche du centre) et à se rapprocher d'un sommet optimal grâce à différents mécanismes : on peut par exemple utiliser la méthode *affine scaling*, qui consiste à maximiser le critère en réduisant le domaine à un ellipsoïde centré sur le point représentant la solution courante (la nouvelle solution trouvée sera alors le centre d'un autre ellipsoïde et ainsi de suite jusqu'à l'optimum).
- La méthode de génération de colonnes [Desaulniers 05, Huart 09] : cet algorithme de résolution de problèmes de grande taille consiste à décomposer l'ensemble des contraintes en sous-ensembles et à considérer seulement une partie des variables et des contraintes du problème. Au fur et à mesure de la résolution, les variables et contraintes non prises en compte sont réintégrées une à une en réitérant la résolution à chaque fois pour affiner la solution.

Dans le cadre du calcul d'une séquence d'avions sur une piste, ni le critère défini dans la partie 5.2, ni les contraintes sont linéaires : cela réduit donc les possibilités d'utilisation de ce type d'algorithmes pour résoudre le problème.

4.2.4 Programmation par contraintes

Dans sa formulation générale, la programmation par contraintes [Papadimitriou 82, Horst 95] est définie par un triplet $(\mathcal{X}, \mathcal{D}, \mathcal{C})$:

- $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$: l'ensemble des variables du problème,

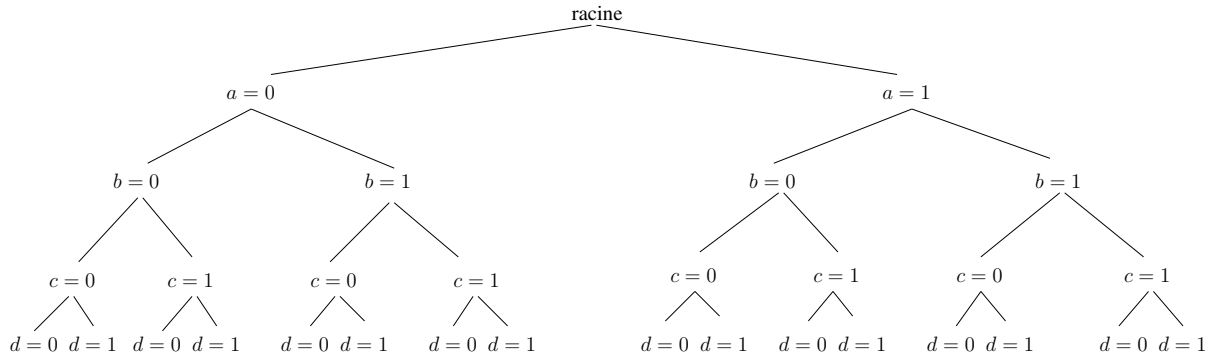


FIG. 4.1 – Exemple d'arbre de solutions en programmation par contraintes

- $\mathcal{D} = D_1 \times D_2 \times \dots \times D_n$: les domaines respectifs de ces variables,
- $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$: un ensemble de contraintes entre les variables.

Chaque contrainte est une relation logique entre certaines variables : elle peut se représenter comme une paire (t, R) avec t un sous-ensemble de variables et R les domaines restreints par la contrainte pour ces variables.

L'objectif est de déterminer une valeur pour chaque variable qui respecte toutes les contraintes : la recherche de solutions admissibles peut être représentée par un parcours d'arbre dont chaque niveau de profondeur correspond à l'affectation d'une valeur à une variable. Posons par exemple :

$$\begin{aligned}\mathcal{X} &= \{a, b, c, d\} \\ \mathcal{D} &= \{0, 1\}^4 \\ \mathcal{C} &= \{a \neq b, a + c \leq b, cd = 1\}\end{aligned}$$

On obtient l'arbre de choix représenté par la figure 4.1. La résolution consiste à parcourir cet arbre en vérifiant le respect des contraintes à chaque nœud et en *propageant* les contraintes aux nœuds suivants pour réduire les valeurs possibles des variables non encore traitées. Dans l'exemple, le nœud représentant « $a = 0, b = 0$ » (deuxième nœud dans un parcours en profondeur) ne respecte pas la contrainte « $a \neq b$ » et l'exploration remonte alors d'un niveau (étude de la branche « $a = 0, b = 1$ »).

Afin d'accélérer la recherche, les contraintes peuvent être *relaxées* (ignorées) : si on relaxe la contrainte « $cd = 1$ » de l'exemple cité, la solution $\{a, b, c, d\} = \{0, 1, 1, 0\}$ devient acceptable. Lorsque l'arbre de recherche est très grand, la relaxation de contraintes permet de trouver des affectations aux variables proches de l'affectation optimale dans un temps réduit. Une fois que des valeurs proches de la solution ont été trouvées, les contraintes sont *resserrées* (à nouveau prises en compte dans la résolution) progressivement afin d'obtenir une solution optimale.

La formulation d'un problème de **minimisation sous contraintes** peut s'effectuer par l'ajout d'une variable supplémentaire c représentant le critère à minimiser. Cette variable est alors associée à une contrainte représentant une borne supérieure : $c \leq B$. Le problème est alors résolu par itérations en modifiant la borne :

- la borne est réduite lorsqu'une solution est trouvée,
- elle est augmentée dans le cas contraire.

La variation de cette borne peut s'effectuer soit avec un pas élémentaire fixe, soit par dichotomie.

Ce type de résolution (parcours d'arbre, réduction d'arbre avec utilisation de bornes) est appelé algorithme de *branch & bound* [Miquel 01] : le principe général de ces algorithmes consiste à trouver une solution admissible dans l'arbre d'exploration des solutions, en utilisant éventuellement des *heuristiques* pour choisir quelle(s) branche(s) explorer à chaque étape.

4.3. RÉOLUTION À L'AIDE D'UNE LIBRAIRIE DE PROGRAMMATION PAR CONTRAINTES 35

Dans le cas d'une séquence d'avions sur une piste, deux formulations sont possibles :

- soit par l'intermédiaire des heures de décollage et d'atterrissage à trouver ;
- soit par l'intermédiaire du rang de chaque avion dans la séquence.

Dans ces deux cas, les contraintes sont relatives aux temps de séparations réglementaires entre les avions.

La programmation par contraintes semble donc être adaptée au problème. Le nombre de contraintes et l'espace de recherche étant toutefois très grands, l'utilisation de bibliothèques standards de programmation par contraintes peut s'avérer insuffisante : comme souvent, la mise au point d'un algorithme de branch & bound spécifique au problème peut permettre de réduire considérablement le temps de calcul d'une solution.

4.3 Résolution à l'aide d'une bibliothèque de programmation par contraintes

4.3.1 Principes des bibliothèques de programmation par contraintes (PPC)

Les bibliothèques de programmation par contraintes sont des solveurs paramétrables permettant à l'utilisateur de formaliser facilement un problème par ses variables, leurs domaines, les contraintes et éventuellement un critère à minimiser et des heuristiques de recherche. On peut citer, parmi les plus connues :

- Ilog Cplex [IBM 10] : un moteur de programmation mathématique haute performance multi-plate-formes développé par IBM. Il est principalement optimisé pour résoudre des problèmes de programmation linéaire (PL) ou de programmation linéaire en nombre entier (PLNE). Les bibliothèques sont disponibles afin d'intégrer directement les solveurs dans un code *C*, *C++*, *Java* ou encore *.NET*. Cependant un environnement complet (CPLEX Studio Integrated Development Environment) est mis à disposition des utilisateurs afin d'utiliser un langage OPL (Optimization Programming Language) de plus haut niveau pour résoudre leurs problèmes.
- Choco [EMN 10] : une bibliothèque de programmation par contraintes libre d'accès, développée en Java par l'École des Mines de Nantes. Elle propose diverses méthodes de résolution telles que des méthodes de satisfaction (calcul d'une solution, de toutes les solutions ou par itération sur l'ensemble des solutions) ou encore des méthodes d'optimisation (maximisation ou minimisation). Choco est donc une bibliothèque générale de programmation par contraintes (en comparaison avec CPLEX qui est optimisé pour la PL et la PLNE).
- FaCiLe [CENA 05] : une bibliothèque de programmation par contraintes libre d'accès, développée en Ocaml à l'ENAC. Sa description fait l'objet de la section suivante.

4.3.2 Description générale de la bibliothèque Facile

FaCiLe est une bibliothèque de programmation par contraintes sur les domaines finis (entiers ou ensembles d'entiers). Elle intègre les fonctionnalités habituelles de création et de manipulation de variables, d'expressions arithmétiques, de contraintes (éventuellement non linéaires), de contraintes globales (contraintes s'exprimant sur l'ensemble des variables, comme la différence, la cardinalité, le tri, etc.) et d'heuristiques de recherche.

Lorsque cette bibliothèque est intégrée directement dans son langage de développement, elle tire parti de tous les avantages du compilateur Ocaml :

- inférence de type et typage statique strict,
- polymorphisme,
- production de code natif optimisé (pour la plupart des plates-formes),

- gestion automatique de la mémoire.

Cette librairie permet donc d'effectuer rapidement du prototypage et des expérimentations sur des problèmes de programmation par contraintes. Pour toutes ces raisons, et parce que le simulateur de trafic aéroportuaire est également écrit en OCaml, cette librairie a été retenue pour la première implémentation du problème des séquences d'avions sur une piste.

4.3.3 Implémentation du problème des séquences d'avions

La première étape consiste à définir les variables du problème, en leur associant un domaine fini. Cela nécessite la définition d'une borne supérieure T_{sup} des heures à programmer : cette borne peut être définie comme celle calculée au chapitre précédent pour estimer la taille de l'espace de recherche (partie 3.4.2) :

$$T_{sup} = H + |\mathcal{D}|Sep_{max}$$

avec \mathcal{D} l'ensemble des départs et Sep_{max} le temps de séparation maximal pouvant être imposé entre deux avions. Les **variables** $(t_i)_{1 \leq i \leq N}$ (heures de décollage ou d'atterrissage) sont ainsi définies sur un domaine correspondant à un intervalle fini d'entiers :

$$\begin{cases} \forall i \in \mathcal{A}, t_i \in [T_{min}^i; T_{max}^i], \text{ comme cela a été défini au 3.3.2} \\ \forall i \in \mathcal{D}, t_i \in [T_{min}^i; T_{sup}] \end{cases}$$

Les **contraintes** de séparation ne diffèrent pas de celles qui sont définies au 3.3.3, l'ordre des arrivées étant connu, la vérification de leurs séparations s'effectue unilatéralement. En revanche cette vérification reste bilatérale pour les départs :

$$\begin{cases} \forall (i, j) \in \mathcal{A}^2, i < j \Rightarrow t_i + Sep_{i/j} \leq t_j \\ \forall (i, j) \in \{1, \dots, N\}^2 - \mathcal{A}^2, i \neq j \Rightarrow t_i + Sep_{i/j} \leq t_j \text{ ou } t_j + Sep_{j/i} \leq t_i \end{cases}$$

Cela représente donc au total :

$$\frac{N(N-1)}{2} \text{ contraintes}$$

Le **critère** à minimiser est défini comme la somme des pénalités associées aux heures de décollage : cela peut être implémenté en ajoutant, pour chaque départ $i \in \mathcal{D}$, une variable c_i associée à des contraintes de définition :

- pour un départ i non régulé :

$$c_i = t_i - T_{min}^i$$

- pour un départ i régulé, en définissant $T_0 = \max(T_{min}^i, T_{cfmu}^i + 5min.)$:

$$\begin{cases} t_i \leq T_0 \Rightarrow c_i = 0 \\ t_i \leq T_{cfmu}^i + 10min. \Rightarrow c_i = t_i - T_0 \\ t_i > T_{cfmu}^i + 10min. \Rightarrow c_i = 4(t_i - T_0) \end{cases}$$

Une fois toutes ces variables et contraintes définies, le solveur de FaCile peut être utilisé directement pour minimiser la somme des c_i . Cette implémentation est très rapide à effectuer mais n'utilise pas d'heuristique particulière pour orienter la recherche : les temps de calcul d'une solution risquent donc d'être rapidement trop grand lorsque le nombre d'avions augmente, même si la librairie garantit toujours l'obtention d'un optimum global du problème.

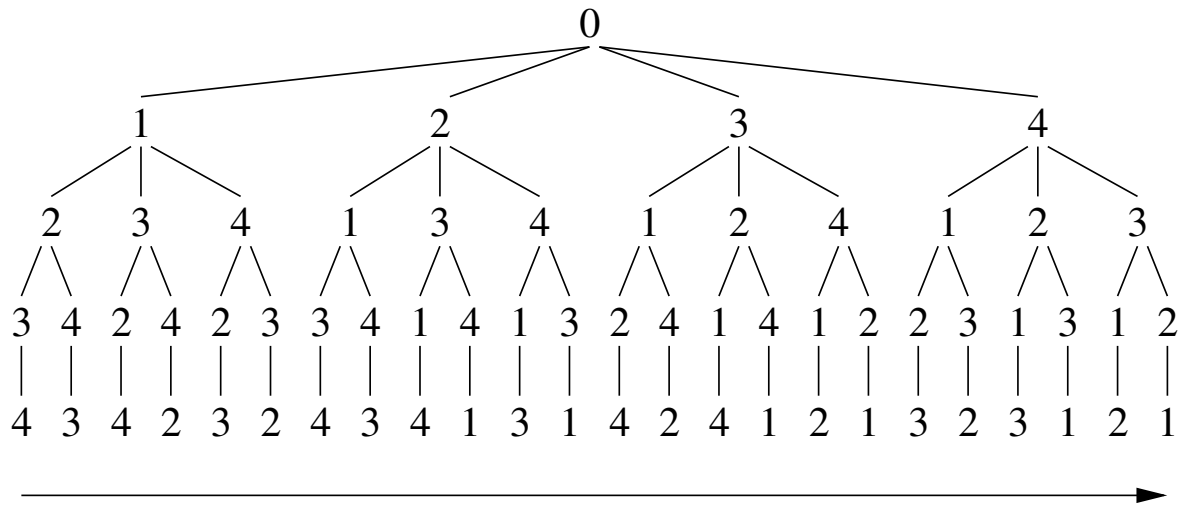


FIG. 4.2 – Arbre de permutations

4.4 Élaboration d'un algorithme spécifique

4.4.1 Principes

L'approche par classement (comme définie au 3.4.2) change la formulation du problème : les variables ne sont plus les heures de décollage et d'atterrissage mais le rang de chaque avion dans la séquence. L'arbre de recherche est donc réduit aux permutations possibles des avions comme l'illustre la figure 4.2 pour quatre mouvements. Les heures de décollage et d'atterrissage peuvent être déduites du classement en positionnant chaque avion le plus tôt possible après celui qui le précède.

La définition d'un algorithme de *branch & bound* spécifique doit ainsi permettre d'explorer rapidement des classements intéressants en utilisant des heuristiques adaptées au problème.

4.4.2 Première mise en œuvre de l'algorithme

Chaque nœud de l'arbre des permutations correspond à une séquence d'avions déjà positionnés (au cours de l'exploration des nœuds supérieurs) et à un ensemble d'avions non encore traités (ceux qui le seront dans le reste de la branche à parcourir). Pendant l'exploration de l'arbre, chaque nœud peut donc être associé au coût de la séquence partielle construite à ce point. Ce coût fournit un minorant du coût des séquences pouvant être construites à partir de ce nœud (car l'ajout des autres avions dans la séquence ne fera qu'augmenter son coût) : si ce coût s'avère supérieur ou égal au coût de la meilleure solution déjà trouvée, la suite de la branche n'a pas besoin d'être explorée.

En conséquence, l'exploration de l'arbre sera considérablement réduite si des solutions proches de l'optimum sont rapidement trouvées : l'ordre dans lequel les différents classements sont explorés a une grande importance sur le nombre de nœuds qui seront effectivement visités. Un ordre de parcours intuitif consiste à partir du classement premier arrivé, premier servi, car dans le cas où ce classement respecte les contraintes de séparation, il constitue le meilleur classement possible : les délais des départs non régulés sont nuls et les créneaux sont respectés au mieux. Aucune pénalité n'étant associée aux heures d'atterrissages, il semble par ailleurs judicieux de chercher initialement à les positionner le plus tard possible pour pouvoir insérer un maximum de décollages plus tôt. Un classement initial

peut ainsi être défini en fonction des $(t_i)_{1 \leq i \leq N}$ suivants :

$$\begin{cases} \forall i \in \mathcal{D}, t_i = T_{min}^i \\ \forall i \in \mathcal{A}, t_i = T_{max}^i \end{cases}$$

De plus, cette définition du classement initial permet de réduire l'ensemble des avions pouvant effectivement être positionnés au rang suivant, à chaque étape du parcours : en effet, les arrivées étant initialement classées en fonction de l'heure maximale d'atterrissage, aucun décollage classé ultérieurement ne peut être inséré plus tôt (car son heure minimale de décollage est supérieure à l'heure maximale d'atterrissage de l'arrivée). Ainsi, si l'ensemble des avions restant à insérer dans la séquence reste classé dans l'ordre initial, un sous-ensemble d'avions *candidats* au rang suivant peut être extrait rapidement (complexité linéaire) puisqu'il s'agit des premiers d'entre eux (jusqu'au prochain atterrissage).

Enfin, une amélioration supplémentaire de l'algorithme consiste à vérifier que le prochain atterrissage restant à traiter (s'il y en a un) peut toujours être inséré, en le comparant au dernier mouvement positionné. Cette vérification peut être effectuée pendant le calcul des avions candidats au rang suivant et permet d'abandonner plus rapidement de nombreuses branches sans solution.

```

Sequence( $\mathcal{E}$ ) :                               ( $\mathcal{E}$  : avions à séquencer, dans l'ordre initial)
┌
│ Candidats( $\mathcal{S}, \mathcal{R}$ ) :                       ( $\mathcal{S}$  : séquence partielle,  $\mathcal{R}$  : avions restant à séquencer)
│   si  $\mathcal{R}$  contient au moins une arrivée alors
│     soient  $(r_i)_{1 \leq i \leq n}$  les éléments de  $\mathcal{R}$ 
│     soit  $a$  indice de la première arrivée de  $\mathcal{R}$ 
│     si  $\mathcal{S} \neq \emptyset$  alors
│       soit  $d$  : dernier élément de  $\mathcal{S}$  et  $t_d$  : son heure associée
│       si  $T_{max}^a < t_d + Sep_{d/r_a}$  alors retourner  $\emptyset$ 
│       sinon retourner  $(r_i)_{1 \leq i \leq a}$ 
│     sinon retourner  $(r_i)_{1 \leq i \leq a}$ 
│   sinon retourner  $\mathcal{R}$ 
│ soit  $C_{max} : +\infty$                                (Coût de la meilleure solution trouvée)
│ soit  $Solution : \emptyset$                              (Meilleure solution trouvée)
│ Explorer( $\mathcal{S}, \mathcal{R}$ ) :                               ( $\mathcal{S}$  : séquence partielle,  $\mathcal{R}$  avions restant à séquencer)
│   soit  $C_S$  : coût de la séquence partielle  $\mathcal{S}$ 
│   si  $C_S < C_{max}$  alors
│     si  $\mathcal{R} = \emptyset$  alors
│        $C_{max} \leftarrow C_S$ 
│        $Solution \leftarrow \mathcal{S}$ 
│     sinon pour chaque  $a \in \text{Candidats}(\mathcal{S}, \mathcal{R})$ , faire Explorer( $\mathcal{S} + a, \mathcal{R} - a$ )
│ Explorer( $\emptyset, \mathcal{E}$ )
│ retourner  $Solution$ 
└
Avec |  $\mathcal{S} + a$  : séquence obtenue en ajoutant  $a$  à la fin de  $\mathcal{S}$ 
     |  $\mathcal{R} - a$  : liste  $\mathcal{R}$  privée de  $a$ 

```

Algorithme II.2: Sequence : calcul de la meilleure séquence d'avions sur une piste

L'algorithme II.2 correspond ainsi à une première version du parcours de l'arbre des permutations en profondeur, prenant en compte les restrictions introduites ci-dessus. Cet algorithme sert de base à l'algorithme final, qui utilise d'autres propriétés particulières au problème pour réduire encore l'exploration, comme cela est détaillé dans la partie suivante. Dans sa version initiale, l'algorithme se décompose en deux fonctions :

- La fonction `Candidats` extrait de la liste \mathcal{R} des avions restant à traiter ceux qui peuvent être positionnés immédiatement à la suite de la séquence partielle \mathcal{S} . Si le prochain atterrissage à traiter s'avère incompatible avec la séquence partielle, la fonction retourne l'ensemble vide.
- La fonction `Explorer` visite récursivement tous les nœuds de l'arbre, en descendant toutes les branches devant être explorées : à chaque étape, le coût de la séquence partielle est comparée au coût C_{max} de la meilleure solution trouvée à ce point. Tant que le coût partiel est inférieur à cette borne, chaque avion de la liste des candidats (maintenue dans l'ordre initial) est successivement ajouté à la séquence partielle (et supprimé de la liste des avions non traités) pour former un nouveau nœud à explorer.

4.4.3 Amélioration de l'algorithme

Comme souvent dans les problèmes d'ordonnancement, l'étude de propriétés particulières au problème permet de réduire l'espace de recherche à parcourir. Dans le cas d'une séquence d'avions sur une piste, trois différents types d'améliorations de l'algorithme ont été envisagées et sont présentées dans cette partie.

Affinement du minorant du coût

Le minorant utilisé dans l'algorithme II.2 pour évaluer le coût des séquences pouvant être construites à partir de chaque nœud de l'arbre peut être affiné, car le coût f^i associé à chaque avion i est toujours une fonction croissante de son heure de programmation t_i : ainsi, un minorant c_{min}^i du coût associé à chaque avion restant i peut être calculé : il correspond au coût qui serait associé à cet avion s'il était positionné le plus tôt possible après le dernier avion a de la séquence partielle \mathcal{S} :

$$\forall i \in \mathcal{R}, c_{min}^i = f^i(\max(T_{min}^i, t_a + Sep_{a/i}))$$

Un minorant C_{min} du coût de toutes les séquences pouvant être construites à partir d'un nœud de l'arbre, caractérisé par la séquence partielle \mathcal{S} et la liste d'avions non encore positionnés \mathcal{R} est alors donné par :

$$C_{min} = C_S + \sum_{i \in \mathcal{R}} c_{min}^i$$

Lorsque ce minorant est supérieur ou égal au coût de la meilleure solution trouvée, la branche n'a pas besoin d'être explorée.

Détection de symétries du problème

Les contraintes et le critère du problème sont définis en fonction des trois types de mouvements devant être séquencés (arrivées, départs non régulés, départs régulés) et de leurs catégories de turbulences de sillage (légère, moyenne ou haute). L'ordre entre certaines paires d'avions « équivalents » (même type et même catégorie) peut donc être fixé définitivement, sans que cela n'empêche d'obtenir une séquence optimale, puisque les contraintes et le critère associé aux deux avions sont identiques :

- Tout d'abord, l'ordre des arrivées ne peut être modifié (mais cette contrainte est déjà intégrée dans la version initiale de l'algorithme).
- Deux décollages non régulés de même catégorie de turbulence de sillage peuvent être systématiquement programmés dans l'ordre de leurs heures minimales de décollage (car dans l'ordre inverse, le délai résultant ne pourrait en aucun cas être strictement inférieur). Le classement initial défini au 4.4.2 peut donc être considéré comme définitif dans ce cas.

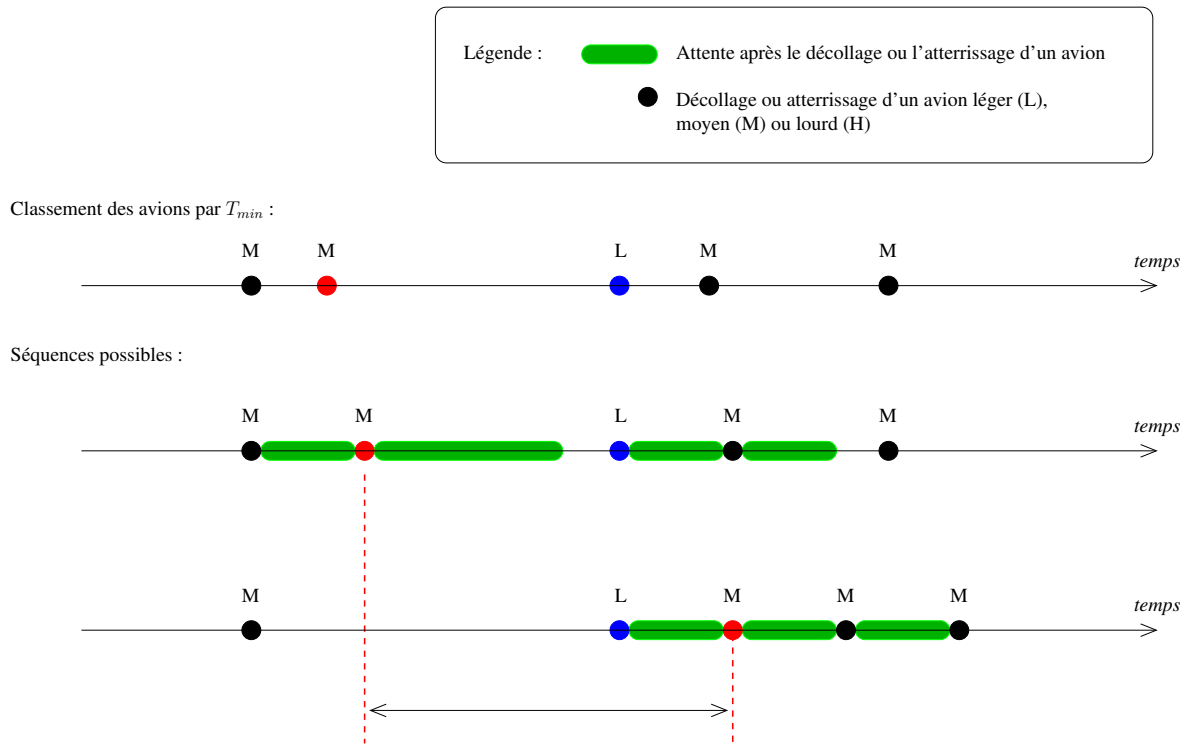


FIG. 4.3 – Branches sous-optimales pouvant être supprimées

- Il en est de même entre deux décollages régulés de même catégorie, si leurs créneaux CFMU sont classés dans le même ordre que leurs heures minimales de décollage, car le critère associé à ce type de mouvement est la même fonction croissante, définie en fonction de l'heure de décollage imposée par le CFMU.

Suppression de branches sous-optimales

Au cours de l'exploration de chaque nœud de l'arbre, une dernière amélioration de l'algorithme concerne les avions non encore traités, lorsque l'un d'entre eux, noté i , peut être positionné au rang $n + 1$ (soit après le dernier avion d de la séquence partielle S), sans en retarder un second, noté j :

$$\exists(i, j) \in \mathcal{R}^2, t_i + Sep_{i/j} \leq T_{min}^j, \text{ avec } t_i = \max(T_{min}^i, t_d + Sep_{d/i})$$

Le coût associé à l'avion i étant une fonction croissante de son heure de décollage ou d'atterrissage, il ne peut être bénéfique de positionner l'avion j au rang $n + 1$ après la séquence partielle S , car cela retarderait l'avion i sans avancer l'avion j , comme illustré figure 4.3. Dans ce cas, l'avion j peut donc être retiré de l'ensemble des candidats au rang $n + 1$ (bien que l'ordre entre l'avion i et l'avion j dans la séquence optimale ne soit pas connu à ce stade).

4.4.4 Version finale de l'algorithme

Les améliorations décrites dans la partie précédente sont mises en place dans l'algorithme II.3, par l'ajout de deux nouvelles fonctions à l'algorithme II.2 :

```

Sequence_Opt( $\mathcal{E}$ ) :
  Heuristique( $\mathcal{S}, \mathcal{R}$ ) : (minorant du coût engendré par les avions de  $\mathcal{R}$  après  $\mathcal{S}$ )
  | retourner  $\sum_{a \in \mathcal{R}}$  coût de  $a$  juste après  $\mathcal{S}$ 
  Ameliorable( $\mathcal{S}, i, j$ ) : (teste si  $j$  peut être mieux que  $i$  juste après  $\mathcal{S}$ )
  | soit  $t_i$  : heure au plus tôt de  $i$  juste après  $\mathcal{S}$ 
  | si  $t_i + Sep_{i/j} \leq T_{min}^j$  alors retourner Faux
  | sinon si  $i$  est une arrivée alors retourner Faux
  | sinon si  $j$  est une arrivée alors retourner Vrai
  | sinon si catégorie( $i$ )  $\neq$  catégorie( $j$ ) alors retourner Vrai
  | sinon si  $i$  et  $j$  sont deux départs non régulés alors retourner Faux
  | sinon si  $i$  et  $j$  sont deux départs régulés alors retourner  $T_{cfmu}^j < T_{cfmu}^i$ 
  | sinon retourner Vrai
  Candidats( $\mathcal{S}, \mathcal{R}$ ) : (avions de  $\mathcal{R}$  pouvant candidater juste après  $\mathcal{S}$ )
  | retourner  $\{r_j \in \mathcal{R} \mid \forall i < j, \text{Ameliorable}(\mathcal{S}, r_i, r_j)\}$ 
  soit  $C_{max} : +\infty$  (Coût de la meilleure solution trouvée)
  soit  $Solution : \emptyset$  (Meilleure solution trouvée)
  Explorer( $\mathcal{S}, \mathcal{R}$ ) : ( $\mathcal{S}$  : séquence partielle,  $\mathcal{R}$  : avions restant à séquencer)
  | soit  $C_S$  le coût de la séquence partielle  $\mathcal{S}$ 
  | si  $C_S + \text{Heuristique}(\mathcal{S}, \mathcal{R}) < C_{max}$  alors
  | | si  $\mathcal{R} = \emptyset$  alors
  | | |  $C_{max} \leftarrow C_S$ 
  | | |  $Solution \leftarrow \mathcal{S}$ 
  | | sinon pour chaque  $a \in \text{Candidats}(\mathcal{R})$ , faire Explorer( $\mathcal{S} + a, \mathcal{R} - a$ )
  Explorer( $\emptyset, \mathcal{E}$ )
  Avec |  $\mathcal{S} + a$  : séquence obtenue en ajoutant  $a$  à la fin de  $\mathcal{S}$ 
  |  $\mathcal{R} - a$  :  $\mathcal{R}$  privée de  $a$ 

```

Algorithme II.3: Sequence_Opt : calcul optimisé de la meilleure séquence d'avions

- La fonction *Heuristique* permet de calculer le minorant du coût restant à ajouter pour obtenir une séquence complète à partir de la séquence partielle en cours.
- La fonction *Améliorable* permet de sélectionner les avions devant être retenus pour être candidat au rang suivant dans l'exploration en profondeur (afin d'éliminer les cas symétriques et de supprimer les branches sous-optimales de l'arbre).

4.5 Conclusion

L'utilisation d'une librairie standard de programmation par contraintes permet d'obtenir aisément un algorithme fonctionnel, en décrivant simplement les variables, les contraintes et le critère du problème. Cependant, le programme obtenu de cette manière risque d'aboutir à des temps de calculs rapidement prohibitifs lorsque le nombre d'avions augmente, car aucune propriété particulière du problème n'est étudiée pour réduire l'exploration.

L'élaboration d'un algorithme spécifique permet au contraire de modéliser une approche par classement des avions, a priori plus efficace : le classement initial des avions doit permettre d'une part l'obtention de bonnes solutions dès le début de la recherche, mais également la détection des cas symétriques et la suppression de nombreux sous-espace non optimaux.

La performance et les résultats de ces deux algorithmes sont comparés au prochain chapitre.

Chapitre 5

Applications et résultats

5.1 Introduction

Ce chapitre présente les résultats obtenus dans le cadre de l'optimisation de séquences d'avions sur les pistes. Il a pour but de comparer et d'évaluer les deux méthodes de résolution élaborées au chapitre précédent et de préciser les valeurs envisageables pour chaque paramètre.

En premier lieu, les deux méthodes de résolution sont appliquées à des séquences d'avions relatives à des situations de trafic réelles et chargées. Leur efficacité est comparée en terme de temps de calcul par rapport au nombre d'avions, puis les délais et le respect des créneaux imposés par le CFMU sont mesurés en fonction du critère et des paramètres. La dernière partie introduit l'utilisation d'une fenêtre glissante pour le calcul de séquences sur une journée complète, afin d'étudier l'effet de l'horizon de prédiction et de mesurer plus globalement les effets du critère et des différents paramètres.

5.2 Exemples de calculs de séquences

5.2.1 Présentation

L'objectif de cette première partie est d'évaluer l'efficacité des méthodes de résolution en terme de temps de calcul, ainsi que les valeurs envisageables pour les différents paramètres (coefficients définissant le critère, flexibilité des heures d'atterrissage et marge derrière un atterrissage). Dans ce cadre, la variation de chaque paramètre est étudiée individuellement, en conservant les autres valeurs constantes. Chaque paramètre est donc associé à une valeur *par défaut*, en fonction des pratiques ou des objectifs opérationnels :

- Coefficients définissant le critère (cf. 3.3.4) : $x_{dc} = x_{dr} = 1$ (soit une pénalisation équivalente des délais des départs non régulés et des départs régulés) ;
- Flexibilité des heures d'atterrissage : ± 30 secondes ;
- Marge derrière un atterrissage (en cas de libération de piste tardive) : 5 secondes supplémentaires (soit 65 secondes minimum entre un atterrissage et le mouvement suivant).

Afin d'être comparables, les résultats présentés dans cette partie sont calculés à partir des mêmes données, issues des enregistrements de trafic écoulé sur l'aéroport de Paris Charles de Gaulle, la journée du 22 mars 2002 : cette journée est sélectionnée car elle correspond à un trafic chargé, avec une exploitation en mode banalisé (décollages et atterrissages) de la piste 27 et une exploitation en mode spécialisé (uniquement des décollages) de la piste 26R, ce qui assure différents types de séquences d'avions à optimiser. Les données décrivent :

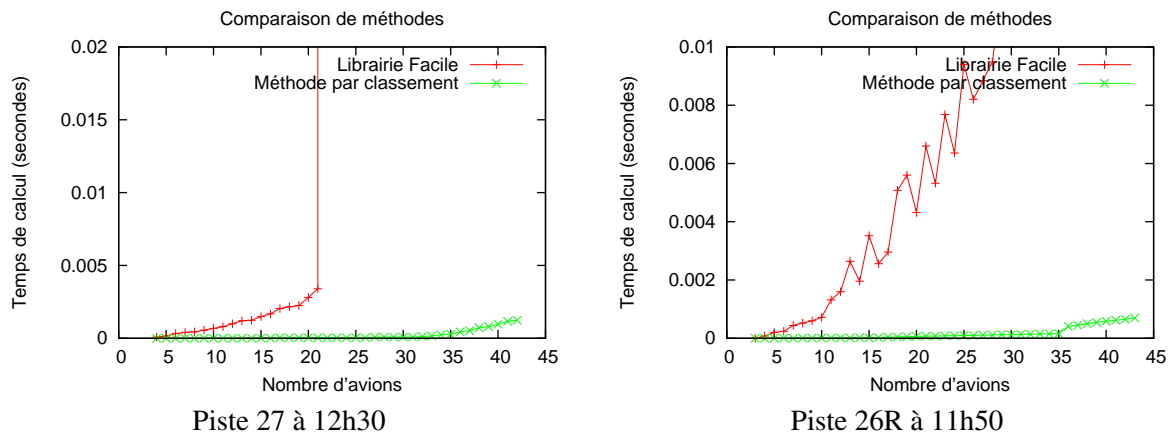


FIG. 5.1 – Temps de calcul des différentes méthodes

- l'heure à laquelle chaque départ prévu est sensé être prêt à libérer son parking ;
- le temps de roulage minimal restant pour chaque départ (prévu ou en cours), calculé à partir des voies de circulation de l'aéroport, dans sa configuration réelle ;
- les heures d'atterrissage prévues par l'AMAN ;
- les temps de séparation imposés entre avions sur la piste, définis en fonction des types et des catégories des avions utilisés.

Pour chacune des deux pistes 27 et 26R, une situation correspondant à un nombre maximal d'avions est choisie :

- La situation à 12h30 pour la piste 27, avec 42 mouvements programmés dans l'heure qui suit : 12 départs non régulés, 15 départs régulés et 15 arrivées.
- La situation à 11h50 pour la piste 26R, avec 43 départs programmés dans l'heure : 16 départs non régulés et 27 départs régulés.

5.2.2 Comparaison des deux méthodes

La figure 5.1 présente les temps de calcul des deux méthodes de résolution en fonction du nombre d'avions considérés pour la piste 27 (à gauche) et pour la piste 26R (à droite). On constate que la méthode par classement, élaborée spécifiquement pour ce problème obtient des temps de calcul bien meilleurs que ceux obtenus en utilisant directement la librairie FaCiLe. Cela montre l'intérêt des propriétés particulières utilisées dans la méthode par classement pour réduire le nombre de permutations explorées.

Lorsque les deux méthodes aboutissent, les séquences trouvées ne sont pas toujours les mêmes mais correspondent bien au même optimum du critère (comme illustré figure 5.2) : l'utilisation, même sans heuristique particulière, de la librairie FaCiLe permet ainsi de vérifier que la méthode par classement, bien que n'explorant qu'une partie réduite de l'espace de recherche, trouve bien un optimum.

Les temps de calcul obtenus avec la librairie FaCiLe peuvent certainement être largement améliorés en guidant l'exploration par la mise en place d'heuristiques, soit générales (largement documentées pour les problèmes classiques d'ordonnancement de tâches [Minton 93, Laborie 07]), soit spécifiques telles que la prise en compte de la séquence premier arrivé, premier servi dans notre cas). Les temps de calcul obtenus avec la méthode par classement étant largement satisfaisants pour un nombre d'avions

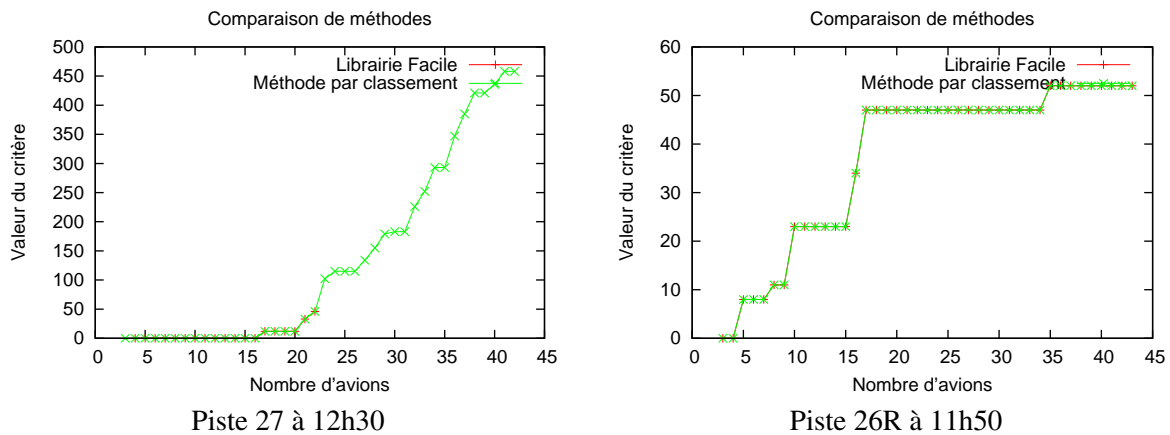


FIG. 5.2 – Valeur du critère pour chaque méthode

en rapport avec les besoins opérationnels, l'utilisation de la librairie Facile n'a pas été approfondie dans le cadre de cette thèse et toutes les séquences d'avions étudiées par la suite sont calculées avec l'algorithme par classement.

5.2.3 Influence du critère

Trois pondérations des délais affectés aux heures de décollage sont envisagées pour le critère global, en affectant différentes valeurs aux coefficients définissant le critère (cf. 3.3.4) :

- La première, notée C_e correspond à celle fixée par défaut, qui pénalise de façon équivalente les délais des départs régulés et non régulés :

$$C_e : x_{dc} = x_{dr} = 1$$

- La seconde, notée C_r , consiste à pénaliser davantage la programmation d'heures de décollage en fin de créneau imposé par le CFMU, afin de limiter le nombre de ces cas et d'éviter que des créneaux ne soient finalement ratés pour cause de retard additionnel au roulage :

$$C_r : x_{dc} = 1, x_{dr} = 2$$

- À l'inverse, la dernière pondération, notée C_d consiste à pénaliser plus fortement le délai des départs non régulés, pour donner priorité à la minimisation des délais des départs non régulés, au risque que certains décollages soumis à créneau soient programmés en fin de période de tolérance :

$$C_d : x_{dc} = 2, x_{dr} = 1$$

Les délais infligés aux départs non régulés sont présentés en fonction de chaque critère sur la figure 5.3 et sont comparés avec les délais infligés par la séquence premier arrivé, premier servi (notée FCFS) :

- En optimisant la séquence, le délai des départs sur la piste 26R représente moins d'un tiers du délai engendré par la séquence premier arrivé, premier servi.
- La piste 27 étant exploitée en mode banalisé, la réduction des délais est moins importante (les heures d'atterrissage sont peu flexibles et fixent partiellement la séquence).

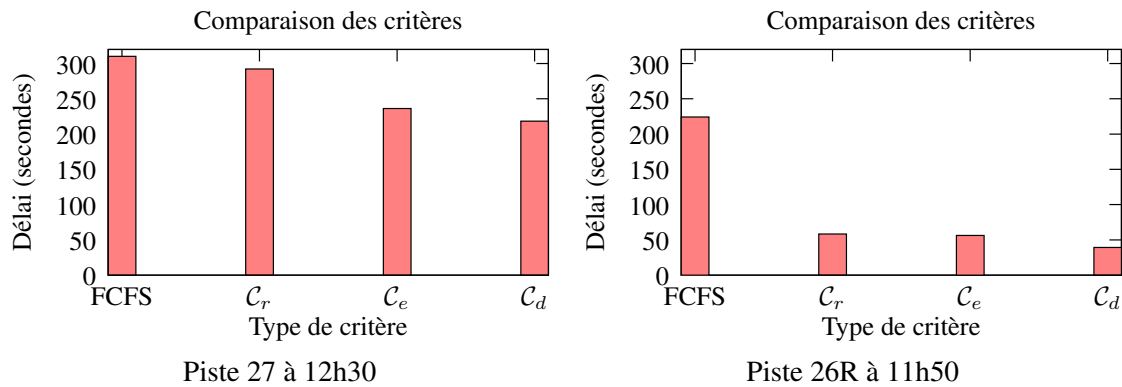


FIG. 5.3 – Délai total des départs en fonction du critère

Piste	Critère	Programmation satisfaisante	Programmation tardive	Créneaux CFMU ratés
27	FCFS	11	2	2
	C_r	11	2	2
	C_e	10	3	2
	C_d	10	3	2
26R	FCFS	25	0	2
	C_r	26	0	1
	C_e	25	1	1
	C_d	24	2	1

FIG. 5.4 – Programmation des créneaux en fonction du critère

- Parmi les différentes pondérations étudiées, on vérifie que le critère C_d est celui qui réduit le plus les délais des départs non régulés. Cela s'explique par le fait que les décollages programmés en fin de créneau imposé (entre +5 et +10 minutes après l'heure imposée) sont moins pénalisés, ce qui permet aux départs non régulés d'être programmés plus tôt.

Lorsqu'il y a un grand nombre de départs régulés (comme sur la piste 26R), les créneaux de décollage imposés par le CFMU sont naturellement moins bien respectés dans la séquence premier arrivé, premier servi, comme le montre la figure 5.4.

De plus, on constate que le nombre de créneaux non respectés est le même pour chaque pondération étudiée. En revanche, le nombre de créneaux programmés tardivement augmente avec les critères C_e et C_d .

La variation des délais et du respect des créneaux en fonction du critère pour une situation précise apparaît relativement faible : le compromis entre la réduction des délais et le respect des créneaux imposés par le CFMU semble ainsi acceptable pour les trois critères définis et le critère initial (pondération équivalente entre départs régulés et non régulés) peut donc être conservé.

5.2.4 Influence de la flexibilité et de la marge

Les paramètres de flexibilité des heures d'atterrissage et de marge derrière un atterrissage n'interviennent que pour la piste 27 en mode banalisé : l'objectif de ces mesures est de quantifier la variation du délai et du respect des créneaux imposés par le CFMU.

La figure 5.5 montre le délai imposé aux départs non régulés dans la situation étudiée. On vérifie

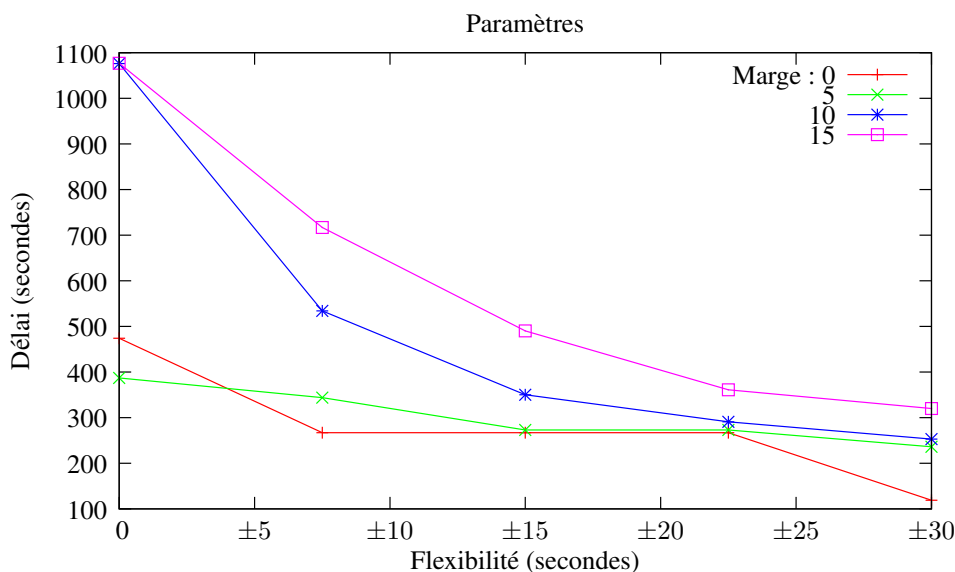


FIG. 5.5 – Délai des départs en fonction de la marge et de la flexibilité (piste 27 à 12h30)

naturellement que plus la flexibilité est grande et la marge faible, plus le délai est réduit : la marge augmente certaines séparations et la flexibilité augmente les décalages possibles des atterrissages, ce qui donne plus de possibilités pour optimiser la programmation des décollages.

Il faut alors faire un compromis entre la réduction du délai et le réalisme des séquences d'avions ainsi programmées : une flexibilité au delà de ± 30 secondes est opérationnellement difficile à mettre en œuvre et une marge nulle risque d'aboutir trop souvent à des séquences non réalisables.

La figure 5.6 montre la façon dont sont programmés les décollages régulés en fonction de la marge et de la flexibilité. On constate que le nombre de créneaux ratés est plus élevé lorsque la flexibilité est nulle : les possibilités d'insertion de départs (régulés ou non) entre les atterrissages deviennent insuffisantes. Lorsque la flexibilité est supérieure à ± 15 secondes, le nombre de créneaux ratés (quelque soit la marge) reste constant mais les délais générés diminuent encore sensiblement (cf. figure 5.5). Il est donc profitable d'augmenter la flexibilité autant que possible, en fonction des possibilités opérationnelles. La valeur de $\pm 30s$ indiquée par défaut permet de réduire significativement les délais.

La marge influence peu le nombre de créneaux ratés mais augmente fortement les délais (cf. figure 5.5) : à moins de vouloir absolument privilégier la pertinence des séquences calculées au détriment des délais générés, une marge supérieure à 5 secondes semble inappropriée.

5.3 Calcul d'une séquence sur une journée complète

5.3.1 Principes

Les mesures effectuées dans la partie précédente sont relatives à une situation particulière sur chaque piste : il apparaît nécessaire de les confirmer en cherchant à optimiser les séquences d'avions sur une journée complète de trafic et d'étudier leur sensibilité à différents horizons de prédiction.

Dans ce cadre, les séquences d'avions sont construites par itérations sur des périodes glissantes de la journée. Ces itérations sont effectuées en fonction des deux paramètres supplémentaires définis au 3.4.3 :

Marge	Flexibilité	Programmation satisfaisante	Programmation tardive	Créneaux CFMU ratés
0	0	9	3	3
	±7,5	10	3	2
	±15	10	3	2
	±22,5	10	3	2
	±30	11	2	2
5	0	9	2	5
	±7,5	10	3	3
	±15	10	3	3
	±22,5	10	3	3
	±30	10	3	3
10	0	7	3	5
	±7,5	9	4	2
	±15	10	3	2
	±22,5	10	3	2
	±30	10	3	2
15	0	7	3	5
	±7,5	7	5	3
	±15	8	5	2
	±22,5	10	3	2
	±30	10	3	2

FIG. 5.6 – Programmation des créneaux en fonction de la marge et de la flexibilité (piste 27 à 12h30)

- L’horizon de prédiction H_p correspond à la taille des périodes sur lesquelles chaque séquence est optimisée par la méthode par classement. Sa valeur par défaut est fixée à 30 minutes.
- Le pas de résolution correspond au décalage $\Delta_p < H_p$ de la période, entre chaque calcul de séquence (en pratique, on prendra $\Delta_p < \frac{H_p}{2}$, pour des valeurs de H_p variant de 5 minutes à 60 minutes).

Les calculs sont effectués en utilisant les plans de vol réels de la même journée (22 mars 2002 à Roissy Charles de Gaulle) : on retrouve donc la piste 27 en mode banalisé, la piste 26R en mode spécialisé (pour les départs) et les paramètres (coefficients du critère, flexibilité et marge) conservent leurs valeurs par défaut.

5.3.2 Influence de l’horizon

La figure 5.7 présente le délai moyen et le délai maximal par avion (sur la journée), en fonction de l’horizon de prédiction. On vérifie que plus l’horizon de prédiction est grand, plus les délais sont réduits, mais la réduction du délai moyen reste très faible (un horizon de prédiction de 5 minutes ne génère que deux secondes de délai moyen supplémentaire, comparé à un horizon de 30 minutes). Le véritable apport d’un horizon de prédiction de l’ordre de 30 minutes est donc relatif au délai maximal (qui est réduit de plusieurs minutes) ou à l’écart-type des délais (qui est réduit de 83.2 à 71.6, comme l’indique le tableau 5.1).

La figure 5.8 illustre la distribution des créneaux en fonction de l’horizon. On constate que l’horizon n’influence pas significativement cette répartition. Les créneaux ratés s’expliquent par le fait que certains avions ne peuvent pas respecter leur créneau en raison d’une sortie tardive du parking et du temps de roulage minimal (c’est le cas des 14 créneaux considérés ratés qui sont en fait impossibles à respecter).

En réalité, prédire une séquence d’avions sur les pistes 60 minutes en avance s’avère incertain et

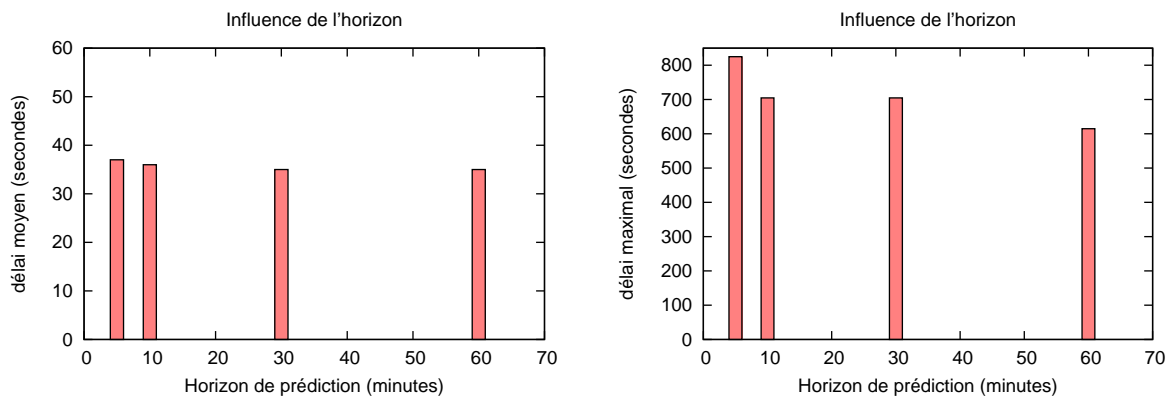


FIG. 5.7 – Délai moyen et maximal des départs (toutes pistes) sur la journée, en fonction de l’horizon de prédiction

Horizon de prédiction (minutes)	Délai moyen (secondes/avion)	Délai maximal (secondes)	Écart type (secondes)	Créneaux ratés
5	37	825	83.2	14
10	36	705	75.5	14
30	35	705	71.6	14
60	35	615	70.7	14

TAB. 5.1 – Délai des départs (toutes pistes) sur la journée, en fonction de l’horizon de prédiction

Horizon	Programmation satisfaisante	Programmation tardive	Créneaux CFMU ratés
5	243	21	14
10	241	23	14
30	243	21	14
60	243	21	14

FIG. 5.8 – Programmation des créneaux (toutes pistes), en fonction de l’horizon de prédiction

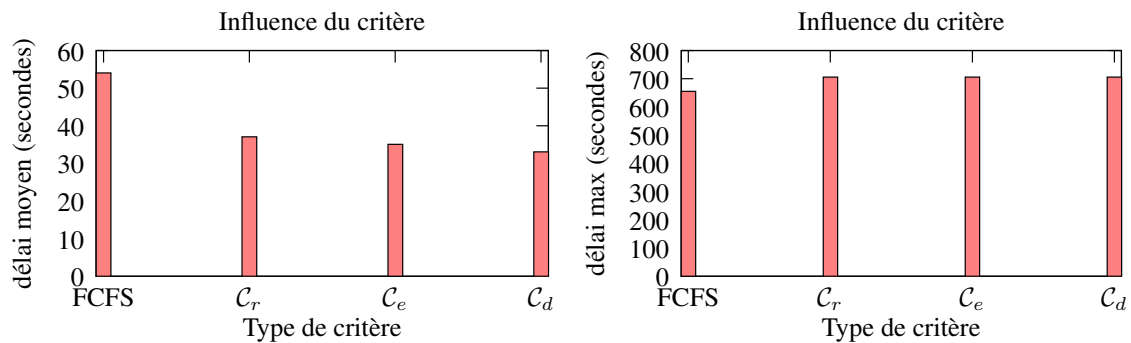


FIG. 5.9 – Délai moyen et maximal des départs (toutes pistes) sur la journée, en fonction du critère

Type de critère	Délai moyen (secondes/avion)	Délai maximal (secondes)	Écart type (secondes)	Créneaux ratés
FCFS	54	655	94.1	16
C_r	37	705	76.5	14
C_e	35	705	71.6	14
C_d	33	705	70.1	14

TAB. 5.2 – Délai des départs (toutes pistes) sur la journée, en fonction du critère

peu fiable en raison des imprévus du trafic au sol. On retient alors l’horizon de prédiction de 30 minutes, qui aboutit déjà à des délais moyens minimaux et qui correspond aux exigences opérationnelles.

5.3.3 Influence des différents critères

L’impact des différentes pondérations du critère sur les délais et le respect des créneaux est mesurée sur la journée complète.

La figure 5.9 présente le délai moyen et le délai maximal par avion (sur la journée), en fonction des pondérations du critère. On vérifie que le critère C_d minimise les délais. On confirme également que l’optimisation des séquences d’avions sur les pistes réduit fortement les délais par rapport aux séquences premier arrivé, premier servi : seul le délai maximal est plus faible avec une séquence premier arrivé premier servi, car chaque avion utilise la piste le plus rapidement possible.

La figure 5.10 montre les différentes programmations des décollages régulés en fonction du critère appliqué. On vérifie là encore que le critère C_r minimise le nombre de décollages programmés tardivement.

Critère	Programmation satisfaisante	Programmation tardive	Créneaux CFMU ratés
FCFS	244	18	16
C_r	246	18	14
C_e	243	21	14
C_d	240	24	14

FIG. 5.10 – Programmation des créneaux (toutes pistes), en fonction du critère

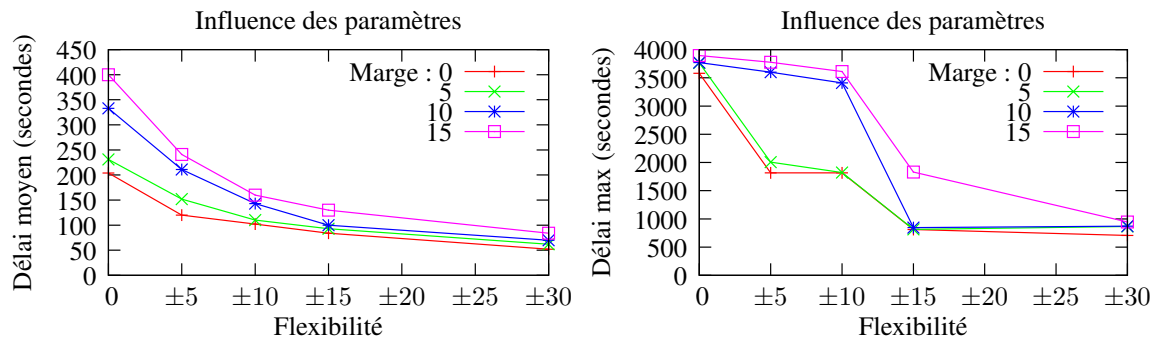


FIG. 5.11 – Délai des départs (sur la piste 27) sur la journée, en fonction de la marge et de la flexibilité

De la même façon que lors du calcul d'une séquence particulière, on constate que la variation des délais et du respect des créneaux est relativement faible. Le critère C_e peut donc être conservé.

5.3.4 Influence de la flexibilité et de la marge

L'étude de l'influence des paramètres sur les délais générés pour une situation a montré qu'une marge de 5 secondes et une flexibilité de ± 30 secondes représentent un bon compromis entre les pratiques actuelles, la minimisation des délais et la programmation tardive des créneaux.

La figure 5.11 fournit le délai moyen et maximal, pour une journée complète, en fonction de ces paramètres : ces résultats confirment l'importance de la flexibilité des heures d'atterrissages (forte diminution du délai moyen lorsque la flexibilité passe de 0 à ± 15 secondes) et l'augmentation importante du délai avec la marge.

La figure 5.13 présente la programmation des décollages régulés en fonction de la marge et de la flexibilité. On constate que l'augmentation de la flexibilité ainsi que la réduction de la marge diminuent significativement le nombre de créneaux programmés tardivement ou même ratés.

La flexibilité des heures d'atterrissage peut donc être fixée à ± 30 secondes et la marge derrière un atterrissage à 5 secondes.

5.4 Conclusion

L'algorithme spécifique par classement élaboré au chapitre précédent permet de calculer des séquences d'avions optimales sur 30 minutes avec un temps de calcul inférieur à la seconde : dans les séquences obtenues, les délais des départs non régulés, ainsi que le nombre de créneaux ratés sont diminués de façon significative par rapport aux séquences de type premier arrivé, premier servi.

La mise en place d'une marge derrière un atterrissage et de la flexibilité des heures d'atterrissage permet de plus d'obtenir un compromis entre la pertinence des séquences calculées et leur caractère optimal. Le paramètre de flexibilité admet cependant une limite opérationnelle et ne peut donc être supérieure à ± 30 secondes. Les résultats obtenus confirment également qu'un horizon de prédiction de l'ordre de la demi-heure (correspondant aux exigences actuelles des opérationnels) est suffisant pour minimiser les délais générés par les pistes.

Ces possibilités d'optimisation des séquences d'avions au niveau des pistes incitent à étudier plus précisément la phase de roulage des avions, car cette dernière doit pouvoir être largement améliorée si

Marge (secondes)	Flexibilité (secondes)	Délai moyen (secondes/avion)	Délai maximal (secondes)	Écart type (secondes)	Créneaux ratés
0	0	204	3580	385.9	6
	±5	120	1815	209.9	4
	±10	102	1815	193.6	4
	±15	84	810	139.7	3
	±30	52	705	91.1	3
5	0	231	3756	426.1	9
	±5	152	2005	273.2	4
	±10	110	1820	205.0	4
	±15	93	815	115.3	3
	±30	62	865	117.8	3
10	0	333	3770	608.6	9
	±5	211	3600	415.6	6
	±10	143	3410	317.6	4
	±15	100	845	161.9	4
	±30	70	870	131.9	3
15	0	400	3895	706.1	9
	±5	241	3775	450.0	9
	±10	160	3610	356.2	4
	±15	130	1830	248.7	4
	±30	84	945	160.8	3

FIG. 5.12 – Mesures pour les départs sur la piste 27, en fonction de la marge et de la flexibilité





























































Marge	Flexibilité	Programmation satisfaisante	Programmation tardive	Créneaux CFMU ratés
0	0	53 	19 	6 
	±5	62 	12 	4 
	±10	62 	12 	4 
	±15	63 	12 	3 
	±30	68 	7 	3 
5	0	51 	18 	9 
	±5	57 	17 	4 
	±10	62 	12 	4 
	±15	64 	11 	3 
	±30	65 	10 	3 
10	0	50 	19 	9 
	±5	55 	17 	6 
	±10	61 	13 	4 
	±15	61 	13 	4 
	±30	65 	10 	3 
15	0	49 	20 	9 
	±5	51 	18 	9 
	±10	59 	15 	4 
	±15	59 	15 	4 
	±30	64 	11 	3 

FIG. 5.13 – Programmation des créneaux sur la piste 27, en fonction de la marge et de la flexibilité

elle intègre l'objectif de réalisation des séquences. Le problème se pose alors de relier ces séquences optimales à la phase de roulage, dans un contexte opérationnel : comment viser ces séquences lors de la résolution des conflits au roulage ? Réciproquement, comment adapter correctement les séquences à réaliser à chaque situation de trafic ?

Ces questions sont la principale motivation de la partie suivante.

Troisième partie

Résolution du trafic au sol

Chapitre 6

Formulation du problème

6.1 Introduction

La gestion de la phase de roulage sur les grands aéroports constitue une tâche très complexe du fait du nombre d'avions et des divers facteurs à prendre en compte (tels que les règles de circulation, la configuration de l'aéroport, les incertitudes sur les vitesses de roulage ou encore les conditions météorologiques). Globalement, l'objectif est de résoudre les conflits entre avions en déterminant les chemins qu'ils doivent emprunter et les éventuelles positions d'attente à leur imposer. La résolution de ces conflits fait intervenir la situation courante des avions, leurs intentions et leurs trajectoires prévues et éventuellement les séquences d'avions souhaitées sur les pistes. Il est donc nécessaire de modéliser chaque situation de trafic, en intégrant précisément toutes ces notions.

La mise au point de méthodes d'optimisation permettant de résoudre les conflits au roulage a déjà été étudiée [Gotteland 04], mais sans chercher à anticiper et à réaliser des séquences d'avions sur les pistes. La partie précédente a néanmoins démontré qu'il peut être bénéfique de lier la gestion du roulage aux séquences optimales d'avions sur les pistes. La difficulté principale pour établir ce lien provient de la différence entre les ordres de grandeur des horizons de prédiction (l'horizon de 30 minutes utilisé et souhaité pour l'optimisation des séquences est incompatible avec les incertitudes liées à la phase de roulage des avions). Ce chapitre a donc pour but de formuler le problème de résolution de conflits en intégrant des séquences d'avions sur les pistes calculées à plus long terme que ces conflits.

Dans un premier temps, une modélisation de la prévision de trajectoires au roulage avec incertitudes sur la vitesse est présentée, puis les notions de conflits au roulage et de détection de ces conflits sont introduites : un modèle de simulation intégrant deux horizons de prédiction différents est alors défini. Le premier horizon (plus grand, de l'ordre de la demi-heure) concerne le calcul des séquences optimales, tandis que le second (inférieur à 10 minutes) est utilisé pour la détection et la résolution des conflits au roulage. Par la suite, le problème de résolution de conflits au roulage est formalisé par ses variables et ses contraintes. Deux critères à minimiser lui sont alors associés : l'un correspondant à une gestion à court terme de la phase de roulage des avions (minimisation des délais de roulage), l'autre intégrant en plus des séquences d'avions à réaliser sur les pistes.

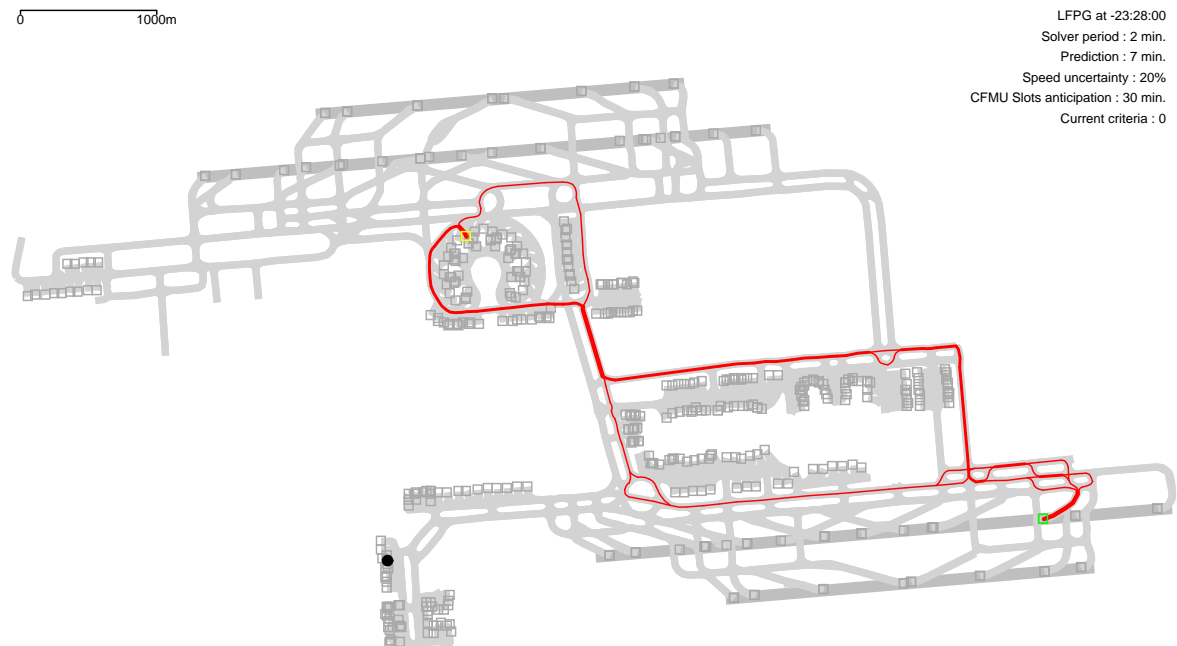


FIG. 6.1 – Calcul des chemins possibles depuis un parking vers une piste dans ATOS

6.2 Prévision de trajectoires au roulage

6.2.1 Affectation de chemins

La description détaillée des voies de circulation de l'aéroport (cf. figure 6.1) permet de le représenter par un premier graphe orienté :

- Chaque nœud correspond à un élément ponctuel de l'aéroport : un parking, un point de report après push-back, une intersection, ou encore un point d'entrée ou de sortie de l'aire de piste.
- Les arcs représentent les portions de voies de circulation permettant à un avion de se rendre d'un nœud à un autre : un taxiway, une bretelle d'accès à une piste ou encore une voie d'entrée ou de sortie d'un parking. Chaque arc est associé à un coût, représentant nominalement le temps nécessaire à un avion pour le parcourir.

Ce graphe correspond logiquement à la description topologique de l'aéroport mais pose certains problèmes pour la mise en œuvre des contraintes liées au cheminement des avions (telles que l'impossibilité de changer de sens de circulation et l'adaptation de la vitesse de roulage en fonction de la portion suivante).

Pour ces raisons, le simulateur ATOS exploite un graphe construit à partir du graphe dual de celui précédemment décrit en l'adaptant aux enchaînements autorisés [Gotteland 04] :

- Les **nœuds** du graphe représentent les liaisons entre les éléments ponctuels de l'aéroport, ainsi que les parkings et les points situés sur un axe de piste. Ces derniers permettent de représenter un chemin complet d'un avion avec son origine et sa destination.
- Les **arcs** sont définis par des couples de liaisons $(\mathcal{L}_1, \mathcal{L}_2)$ ou des couples formés d'un point (un parking ou un point d'un axe de piste) et d'une liaison : (p, \mathcal{L}) ou (\mathcal{L}, p) . L'arc $(\mathcal{L}_1, \mathcal{L}_2)$ n'existe

que 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. De même, les arcs de type (p, \mathcal{L}) et (\mathcal{L}, p) n'existent que si le point p correspond à une extrémité de la liaison \mathcal{L} .

- Le **coût** associé à un arc dépend de la nature des nœuds impliqués et représente le temps nécessaire à un avion pour réaliser l'enchaînement correspondant, parfois pondéré par un coefficient multiplicatif permettant de modéliser les sens préférentiels de circulation.

Ce graphe permet d'associer à chaque avion un ensemble de chemins envisageables. L'énumération de tous les chemins possibles étant trop combinatoire à Roissy, seuls quelques uns d'entre eux doivent être sélectionnés : pour cela, un compromis est défini entre le coût des chemins et leur diversité. Un algorithme spécifique [Gotteland 04] permet d'extraire les chemins les plus pertinents de l'ensemble des chemins possibles : son principe est d'itérer une énumération des k plus courts chemins, en augmentant progressivement le coût des arcs empruntés par les chemins précédemment trouvés. La figure 6.1 illustre un ensemble de chemins trouvés par cette méthode, entre deux points de l'aéroport de Roissy Charles de Gaulle.

Chaque chemin est ensuite discrétisé en une suite de positions ponctuelles, en fonction des vitesses nominales associées à chaque portion de voie de circulation et du pas de discrétisation du temps : la liste de positions ainsi obtenue représente la trajectoire de l'avion, en roulant à la vitesse maximale en chaque point (et sans attente).

6.2.2 Modélisation des incertitudes

Dans un souci de réalisme et d'une éventuelle intégration dans un système en temps réel, la prise en compte de l'incertitude relative aux vitesses de roulage est primordiale [Balakrishnan 07a]. En effet, le fait de considérer une vitesse moyenne pour les avions n'assure pas la fiabilité des solutions apportées puisque les prévisions du système pourraient ne pas correspondre à la réalité : les vitesses réelles observées à partir des traces radar des avions montrent une grande diversité des vitesses sur certaines portions de voies de circulation, sans corrélation possible avec les autres données disponibles (court/long courrier, type d'avion, compagnie, etc.)

Pour cette raison, un taux d'incertitude δ_v sur la vitesse de roulage est défini. La vitesse $v(p)$ prévue d'un avion sur chaque future position p est alors délimitée par un intervalle de vitesses possibles, défini en fonction de la vitesse maximale $v_{max}(p)$ en ce point :

$$(1 - \delta_v)v_{max}(p) \leq v(p) \leq v_{max}(p)$$

La prédiction des futures positions d'un avion intègre donc ce taux d'incertitude : la position prévue d'un avion n'est plus ponctuelle mais correspond à un ensemble de positions possibles, appelé *zone d'incertitude* de l'avion, comme l'illustre la figure 6.2. Les zones d'incertitude des avions augmentent linéairement avec le temps : plus la prédiction de trajectoire s'effectue sur une longue période, moins elle est précise.

Dans le modèle de temps discrétisé, le nombre maximal de positions possibles $n_{pos}(t)$ pour un avion à l'instant t dépend du taux d'incertitude δ_v mais également du pas de rafraîchissement du système, noté Δ_r : en effet, l'étude d'une situation de trafic doit toujours s'effectuer en avance sur la réalité et cette avance doit être supérieure au pas de rafraîchissement. De ce fait, la position initiale des avions au début de l'horizon de prédiction (à l'instant noté t_0) est déjà sujette aux incertitudes. Le nombre maximal de positions de la zone d'incertitude d'un avion à l'instant t s'exprime alors comme suit :

$$n_{pos}(t) = 1 + \lfloor \delta_v(\Delta_r + t - t_0) \rfloor$$

Données en entrée	
Notation	Description
N	Nombre total d'avions.
\mathcal{A}	Ensemble des arrivées.
\mathcal{D}	Ensemble des départs.
\mathcal{D}_C	Ensemble des départs non régulés.
\mathcal{D}_R	Ensemble des départs régulés.
\mathcal{P}	Ensemble des paires d'avions en conflit.
\mathcal{C}	Ensemble des clusters d'avions en conflit.
T_{cfmu}^i	Heure de décollage imposée par le CFMU pour l'avion i .
T_{pmin}^i	Heure minimale de sortie du parking de l'avion i .
T_{min}^i	Heure minimale d'accès à la piste pour l'avion i .
D_{rmin}^i	Durée de roulage minimale (sans attente et à vitesse maximale) de l'avion i .
H_p	Horizon de prédiction pour le calcul des séquences de pistes.
H_r	Horizon de prédiction pour la phase de roulage.
H_e	Horizon de prédiction étendu pour la résolution des conflits après H_r .
Δ_r	Pas de rafraîchissement.
t_0	Heure de début de la situation courante étudiée.
$v_{max}(p)$	Vitesse maximale d'un avion à la position p .
$v(p)$	Vitesse d'un avion à la position p .
$n_{pos}(t)$	Nombre de positions possibles d'un avion à l'instant t de la prédiction.
δ_v	Taux d'incertitude sur la vitesse de roulage des avions.
t_i	Heure d'accès à la piste de l'avion i prévue dans la séquence de pistes.
n_{att}	Nombre d'attentes maximal pour chaque avion sur l'horizon de prédiction. (paramétrable)
n_{ch}	Nombre de chemins maximal pour chaque avion. (paramétrable)
n_{ch}^i	Nombre de chemins disponibles pour l'avion i .
ω_i	Attente stratégique calculée pour l'avion i .
Variables	
Notation	Description
η_i	Chemin choisi pour l'avion i .
θ_i	Niveau de priorité de l'avion i .
d_i	Délai de l'avion i .
ϵ_i	Détour en temps de l'avion i .
n_{att}^i	Nombre d'attentes effective pour l'avion i
τ_a^i	Durée de la $a^{\text{ième}}$ attente de l'avion i . ($1 \leq a \leq n_{att}^i$)
p_i	Pénalité pour l'avion i .

TAB. 6.1 – Table des notations

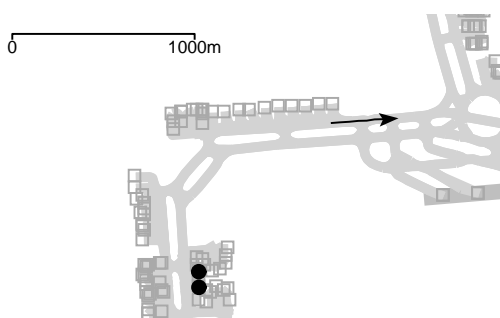


FIG. 6.2 – Représentation des avions avec incertitudes sur la vitesse dans ATOS

La zone d'incertitude d'un avion est naturellement modifiée lorsque ce dernier se voit imposer une attente, car l'avion ne peut pas dépasser la position d'attente avant l'heure de fin d'attente : les ordres d'attentes imposés aux avions permettent ainsi de réduire leurs zones d'incertitude.

6.3 Détection de conflits

6.3.1 Définition d'un conflit

Chaque zone de l'aéroport (parking, taxiway, piste) possède des règles de séparation spécifiques :

1. Deux avions ne peuvent naturellement pas occuper un **parking** au même moment, mais les avions au parking sont considérés isolés du reste du trafic et ne peuvent donc pas être en conflit avec les avions au roulage ou garés dans d'autres parkings.
2. Lors de la phase de **roulage**, les pilotes ont la responsabilité de maintenir une distance de sécurité avec les autres avions (cette distance restant à leur appréciation). Dans le simulateur, cette règle est modélisée par une norme de séparation en distance, fixée à 75 mètres (car cela correspond à la taille des plus gros avions).
3. Sur les **pistes**, en plus de la norme de séparation en distance, les règles suivantes sont imposées :
 - les règles de séparation en temps entre chaque mouvement (décollage ou atterrissage) décrites en 3.3.3 doivent être respectées,
 - un avion ne peut pas occuper l'aire de piste s'il se trouve devant un décollage ou un atterrissage, mais peut en revanche l'occuper s'il se trouve derrière (ce qui permet notamment l'alignement des avions suivants et les traversées de l'aire de piste derrière le mouvement en cours).

Deux avions sont dits en conflit lorsqu'au moins une de ces règles n'est pas respectée dans la prédiction de leurs trajectoires. Chaque conflit peut être associé à :

- la paire d'avions impliqués,
- l'instant du conflit,
- la nature du conflit (la règle qui n'est pas respectée).

La détection des conflits nécessite donc de simuler les déplacements prévus des avions sur l'horizon de prédiction et de mesurer le respect des règles pour chaque paire d'avions, à chaque instant.

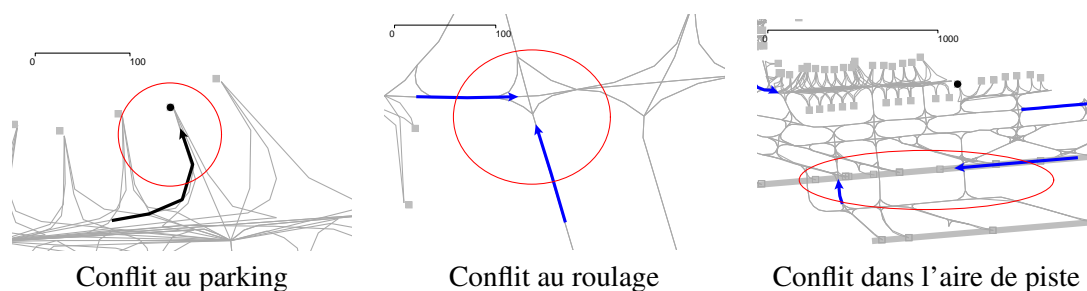


FIG. 6.3 – Détection des conflits avec incertitude

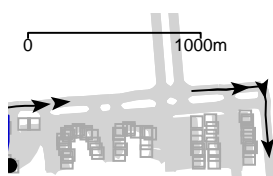


FIG. 6.4 – Chevauchements possibles de zones d'incertitude

6.3.2 Conflits liés à l'horizon de prédiction limité

Le taux d'incertitude sur la vitesse des avions au roulage (entre 20% et 50% pour des vitesses maximales autour de 10m/s) rend inefficace l'utilisation d'un horizon de prédiction supérieur à 10 minutes [Gotteland 04] : au delà, les zones d'incertitude des avions dépassent 1200 mètres et de nombreux conflits sont détectés alors qu'ils n'auront finalement pas lieu. Il est donc nécessaire de considérer un horizon de prédiction H_r limité, entre 5 et 10 minutes selon le taux d'incertitude considéré, pour la détection et la résolution des conflits.

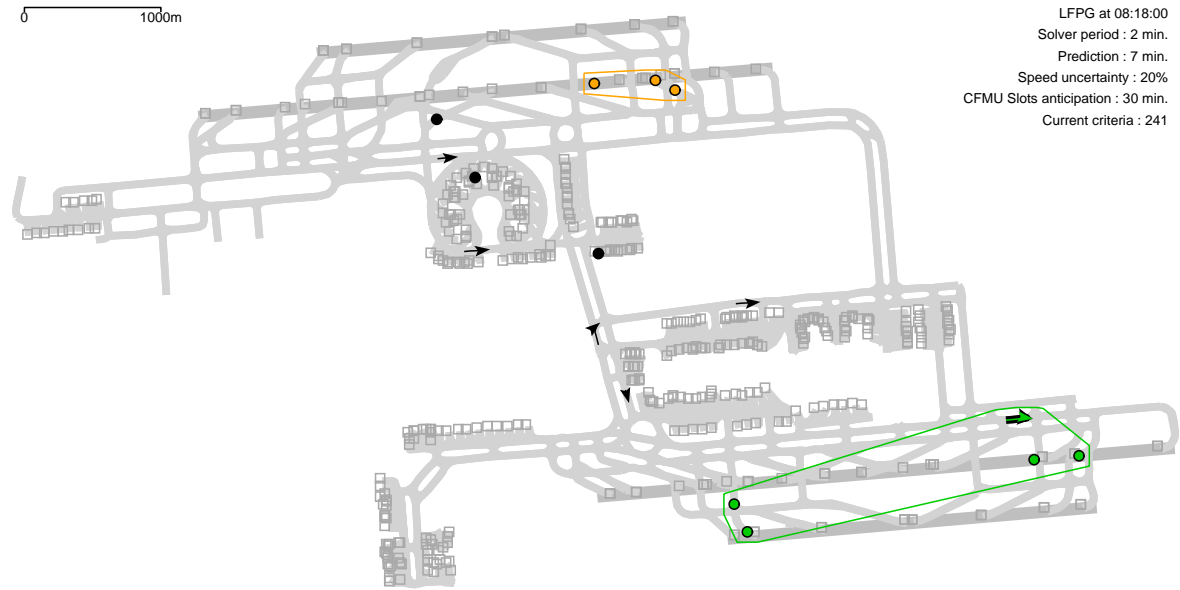
Cette limite de l'horizon de prédiction implique que des conflits sans solution peuvent apparaître dans une situation, car ils n'ont pas été détectés dans la situation précédente (face-à-face sur les longues lignes droites, ou occupation prolongée de la piste alors qu'un atterrissage est programmé). En vue d'éviter de telles situations, un horizon de prédiction *étendu* $H_e > H_r$ est défini pour détecter en avance les cas où elles risqueraient de survenir. Cet horizon étendu est donc associé à deux nouveaux types de conflits détectés de façon préventive :

- les risques de face-à-face en fin d'horizon,
- les occupations d'aires de pistes utilisables par des atterrissages en fin d'horizon.

6.3.3 Conflits avec incertitude sur les vitesses

Les incertitudes sur la vitesse de roulage nécessitent d'adapter la notion de conflit, en l'élargissant à des zones d'incertitude (et non plus des positions ponctuelles) : les règles de séparation décrites doivent donc être respectées entre toutes les positions possibles des avions. La figure 6.3 (extraite de [Gotteland 04]) illustre les conflits pouvant survenir dans chaque zone de l'aéroport en tenant compte des incertitudes sur les vitesses.

Un cas particulier doit cependant être considéré : deux avions qui se suivent peuvent avoir des positions possibles non séparées sans être réellement en conflit, car sinon cela contraindrait le second avion à rouler à la vitesse minimale. Dans la réalité, le pilote du second appareil adapte sa vitesse à celle de l'avion qui le précède. L'ensemble des positions possibles des deux avions n'est donc pas

FIG. 6.5 – Groupement des avions en *clusters*

modifié mais il devient conditionnel (dépendant de la vitesse qui sera adoptée par le premier avion). Ce cas se présente uniquement sur les taxiways et n'est possible que si le second avion est passé par la position la plus reculée du premier : ceci assure que les avions sont l'un derrière l'autre et que leur ordre ne peut être autrement (même dans le cas où le premier avion aurait roulé à la vitesse minimale). La figure 6.4 illustre le chevauchement possible des zones d'incertitude d'avions au roulage, sans conflit.

6.3.4 Groupes d'avions en conflit

Dans le but de décomposer autant que possible le problème de résolution de conflit en sous-problèmes indépendants, tout en cherchant une solution globalement optimale, une fermeture transitive doit être appliquée sur l'ensemble \mathcal{P} des paires d'avions en conflit pour former les groupes d'avions en conflit (appelés *clusters*), notés $(C_i)_{1 \leq i \leq n}$:

$$\forall i \in \{1, \dots, n\}, \forall a \in C_i, \begin{cases} \exists b \in C_i, (a, b) \in \mathcal{P} \\ \forall c \in \mathcal{C} - C_i, (a, c) \notin \mathcal{P} \end{cases}$$

Chaque cluster représente alors un ensemble d'avions en conflit qui peut être initialement considéré indépendamment des autres. La figure 6.5 illustre la formation de ces clusters devant les pistes.

Lors de la résolution des conflits d'un cluster, il est possible que les nouvelles trajectoires affectées aux avions génèrent des conflits avec des avions n'appartenant pas au même cluster. Il faut donc, après chaque résolution des clusters, recalculer les paires d'avions en conflit et fusionner les clusters conflictuels (s'il y en a) puis réitérer le processus. Si une solution optimale est trouvée pour chaque cluster sans que de nouveaux conflits n'apparaissent, l'ensemble des solutions trouvées forme une solution optimale au problème complet.

6.4 Simulation du trafic au sol

6.4.1 Principes

Chaque situation de trafic à étudier correspond à une heure de début t_0 , associée à deux horizons de prédiction : l'un pour le calcul des séquences d'avions sur les pistes (H_p), l'autre pour la résolution des conflits au roulage (H_r), avec $H_r < H_p$. Un certain nombre d'avions sont alors considérés *actifs* sur l'aéroport : soit en mouvement sur les voies de circulation, soit au parking avec une heure de sortie du parking prévue avant la fin de l'horizon de prédiction le plus grand (H_p), soit en l'air avec un atterrissage prévu avant la fin du même horizon de prédiction.

Dans ce contexte, chaque situation de trafic au roulage est étudiée en fonction des paramètres suivants :

- Le taux d'incertitude δ_v , qui définit les tailles des zones d'incertitude des avions ;
- Le nombre maximal d'attentes n_{att} pouvant être imposées à chaque avion : opérationnellement, il est préférable de faire attendre un avion plus longtemps mais à un seul endroit afin de simplifier les ordres à donner aux avions et d'éviter les erreurs de compréhension. En pratique, on prendra $n_{att} = 2$ pour un horizon de prédiction entre 5 et 10 minutes.
- Enfin, en fonction de sa position et de sa destination, chaque avion i peut être associé à un nombre variable de chemins possibles, noté n_{ch}^i . Ces chemins sont indicés du plus court au plus long (en temps), et un paramètre n_{ch} définit le nombre maximal de chemins envisagés pour chaque avion : $n_{ch}^i \in [1; n_{ch}]$.

6.4.2 Heures minimales de décollage

Dans le contexte de chaque situation de trafic, l'heure minimale de décollage T_{min}^i de chaque départ non régulé i ne s'exprime plus directement en fonction de son heure de sortie du parking T_{pmin}^i mais peut dépendre de t_0 et de sa durée minimale de roulage restante D_{rmin}^i :

$$\forall i \in \mathcal{D}_C, T_{min}^i = \max(t_0, T_{pmin}^i) + D_{rmin}^i$$

Il en est de même pour un départ régulé : son heure minimale de décollage s'exprime en fonction de son temps de roulage restant s'il est déjà sorti de son parking, et de la contrainte relative au début de son créneau imposé :

$$\forall i \in \mathcal{D}_R, T_{min}^i = \max(\underbrace{\max(t_0, T_{pmin}^i) + D_{rmin}^i}_{\text{Heure minimale de décollage hors créneau}}, T_{cfmu}^i - 5min.)$$

6.4.3 Notions de détour et de délai

Les critères envisageables pour l'optimisation de la phase de roulage des avions font naturellement référence aux détours et aux délais des avions :

- La notion de **détour** est définie par l'augmentation du temps de roulage lorsqu'un avion est dévié de son plus court chemin. Ainsi, si on note $l_i(k)$ le temps de parcours (sans attente) du $k^{ième}$ chemin possible pour l'avion i (avec $k \in [1; n_{ch}^i]$), le détour ϵ_i de l'avion lorsqu'il suit un chemin η_i se définit par :

$$\epsilon_i = l_i(\eta_i) - l_i(0)$$

- La notion de **délai** représente toute augmentation du temps de roulage lorsqu'un avion est à l'arrêt ou qu'il change de chemin. Le délai d_i de l'avion i correspond donc à la somme de ses temps d'attentes et de son détour :

$$\forall i \in [1; N], d_i = \sum_{a=1}^{n_{att}^i} \tau_a^i + \epsilon_i$$

avec $\left\{ \begin{array}{l} N \text{ le nombre d'avions} \\ \tau_a^i \text{ la durée de la } a^{i\text{ème}} \text{ attente de l'avion } i \\ n_{att}^i \text{ le nombre d'attentes de l'avion } i \end{array} \right.$

En règle générale, les détours sont considérés plus coûteux que les attentes : à délai égal, il est en effet toujours préférable de prévoir une attente pour un avion plutôt qu'un détour, d'une part parce que l'avion reste immobile, d'autre part parce que le temps prévu d'attente peut être réduit par la suite (lors d'une résolution ultérieure, avec une connaissance plus précise de la position des avions), alors qu'un détour implique le plus souvent un délai définitif.

6.4.4 Attente stratégique

Les créneaux de décollage imposés par le CFMU correspondent à des heures de décollage retardées par rapport à l'heure demandée initialement par la compagnie. Ceci implique que les départs régulés sont souvent en mesure d'atteindre la piste avant le début de leur créneau : cela se produit à chaque fois que l'heure du créneau se situe bien après l'heure à laquelle l'avion est prêt à libérer son parking :

$$\underbrace{\max(t_0, T_{pmin}^i) + D_{rmin}^i}_{\text{Heure minimale de décollage hors créneau}} < T_{cfmu}^i - 5min.$$

Heure minimale de décollage hors créneau

Dans ce cas, il devient alors nécessaire de faire attendre l'avion, en essayant de minimiser la gêne occasionnée aux autres avions. Cette attente diffère des autres types d'attentes imposables par le contrôle car elle doit s'effectuer le plus tôt possible, pour éviter que l'avion concerné n'interfère avec le trafic aux abords des pistes (l'attente au parking est naturellement la plus intéressante, lorsqu'elle est possible). Ceci incite à introduire la notion d'*attente stratégique*, notée ω_i , qui correspond à l'écart entre le début du créneau et l'heure minimale de décollage hors créneau :

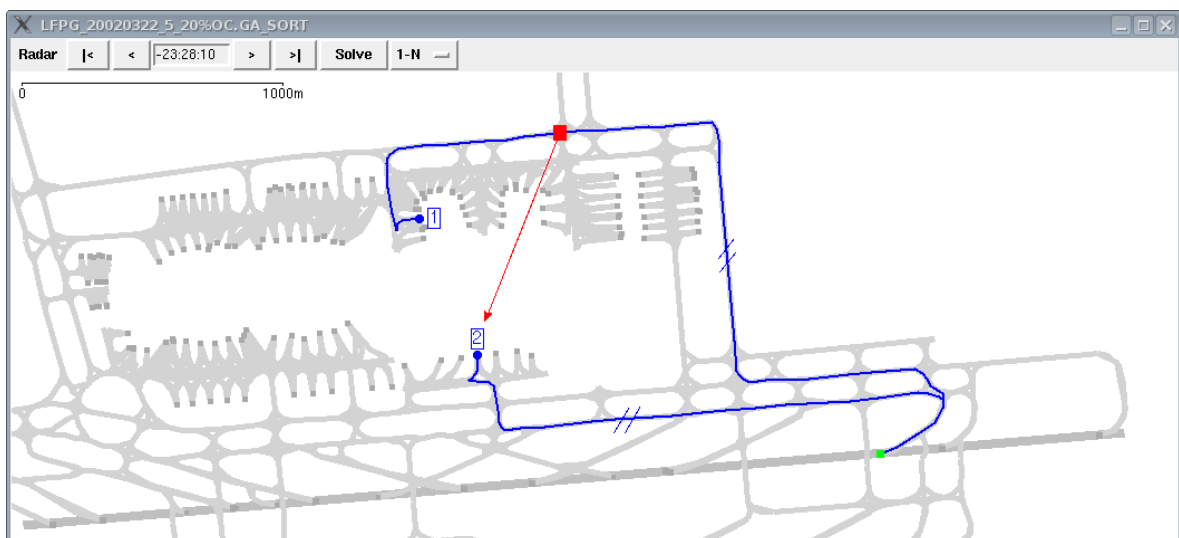
$$\forall i \in \mathcal{D}_{\mathcal{R}}, \omega_i = \max(0, \underbrace{T_{cfmu}^i - 5min. - (\max(t_0, T_{pmin}^i) + D_{rmin}^i)}_{\text{Avance sur le créneau}})$$

Dans le cadre de séquences d'avions à réaliser au niveau des pistes, le centre du problème consiste à ordonner correctement l'ensemble des départs liés à une même piste (dans l'ordre indiqué par la séquence), et ce le plus tôt possible pendant leurs phases de roulage. La notion d'attente stratégique peut dans ce cas être étendue à tous les départs : chacun d'eux possède une heure souhaitée de décollage et peut donc être amené à devoir attendre en amont. À chaque fois que ces attentes peuvent être effectuées au parking, cela doit de plus permettre de réduire le nombre d'avions en mouvement sur les voies de circulation, donc de fluidifier globalement le trafic au roulage et d'améliorer la sécurité.

Le calcul de l'attente stratégique de chaque avion dépend alors de la séquence dans laquelle il apparaît et doit faire intervenir son temps de roulage minimal restant (pour être également applicable entre départs situés dans des zones éloignées de l'aéroport) : un avion doit attendre tant que son temps



L'avion 1 se situe avant l'avion 2 dans la séquence de piste



Départ du parking de l'avion 2 seulement si l'on est assuré que l'avion 1 sera devant lui au roulage

FIG. 6.6 – Heure stratégique de départ du parking

de roulage minimal restant lui permet d'arriver devant la piste avant l'un de ceux qui sont programmés avant lui. Pour deux départs i et j liés à la même piste, il faut donc assurer :

$$t_i < t_j \Rightarrow \max(t_0, T_{pmin}^i) + D_{rmin}^i + \omega_i \leq \max(t_0, T_{pmin}^j) + D_{rmin}^j + \omega_j$$

avec

	t_i, t_j les heures de décollage programmées des avions i et j dans la séquence
	t_0 l'heure de la situation considérée
	D_{rmin}^i, D_{rmin}^j les durées minimales de roulage restantes
	ω_i, ω_j les attentes stratégiques des avions

En cherchant malgré tout à minimiser ces attentes stratégiques, la relation précédente permet d'obtenir une définition par récurrence, pour n départs $(u_i)_{1 \leq i \leq n}$ classés dans l'ordre de la séquence qui les relie :

$$\left\{ \begin{array}{l} \omega_1 = 0 \\ \omega_i = \max\left(0, \underbrace{\max(t_0, T_{pmin}^{u_{i-1}}) + D_{rmin}^{u_{i-1}} + \omega_{u_{i-1}}}_{\text{Heure de décollage de } u_{i-1}} - \underbrace{(\max(t_0, T_{pmin}^{u_i}) + D_{rmin}^{u_i})}_{\text{Heure minimale de décollage de } u_i}\right), \forall i > 1 \end{array} \right.$$

Dans l'exemple de la figure 6.6, l'avion 2 (encore au parking) devra ainsi attendre que l'avion 1 (programmé plus tôt dans la séquence) dépasse le point représenté par le carré rouge pour sortir du parking, afin de limiter les risques qu'il ne se retrouve devant lui.

6.4.5 Boucle de simulation

Pour simuler le trafic d'une journée, les situations sont étudiées par fenêtres glissantes, sur des périodes successives. Ce modèle de simulation permet également une éventuelle intégration dans un système en temps réel : la résolution peut par exemple s'effectuer chaque minute pour les dix minutes suivantes.

Du fait que l'heure minimale de décollage peut changer à chaque situation, les séquences optimales d'avions sur les pistes doivent être recalculées pour chaque nouvelle situation (avant de procéder à la résolution des conflits).

Chaque situation est alors traitée de la façon suivante (comme illustré figure 6.7) :

1. Les séquences de pistes optimales sont calculées sur l'horizon H_p (de l'ordre de 30 minutes) en fonction de la situation courante du trafic.
2. Les conflits au roulage sont résolus sur l'horizon H_r en intégrant éventuellement les séquences optimales à réaliser.
3. Les trajectoires résultantes sont appliquées sur les Δ_r premières minutes de la situation étudiée (avec Δ_r entre une et deux minutes), pour former la situation suivante.

Dans un environnement en temps réel, le passage à la situation suivante nécessite un rafraîchissement du système, en tenant compte des positions réelles des avions. Dans la simulation, ce rafraîchissement est remplacé par la réduction des zones d'incertitude de chaque avion : une position ponctuelle possible de l'avion est choisie et la prévision de la trajectoire de l'avion se fait à partir de cette nouvelle position. Cette opération détermine ainsi les nouvelles zones d'incertitude des avions pour le début de la nouvelle situation étudiée.

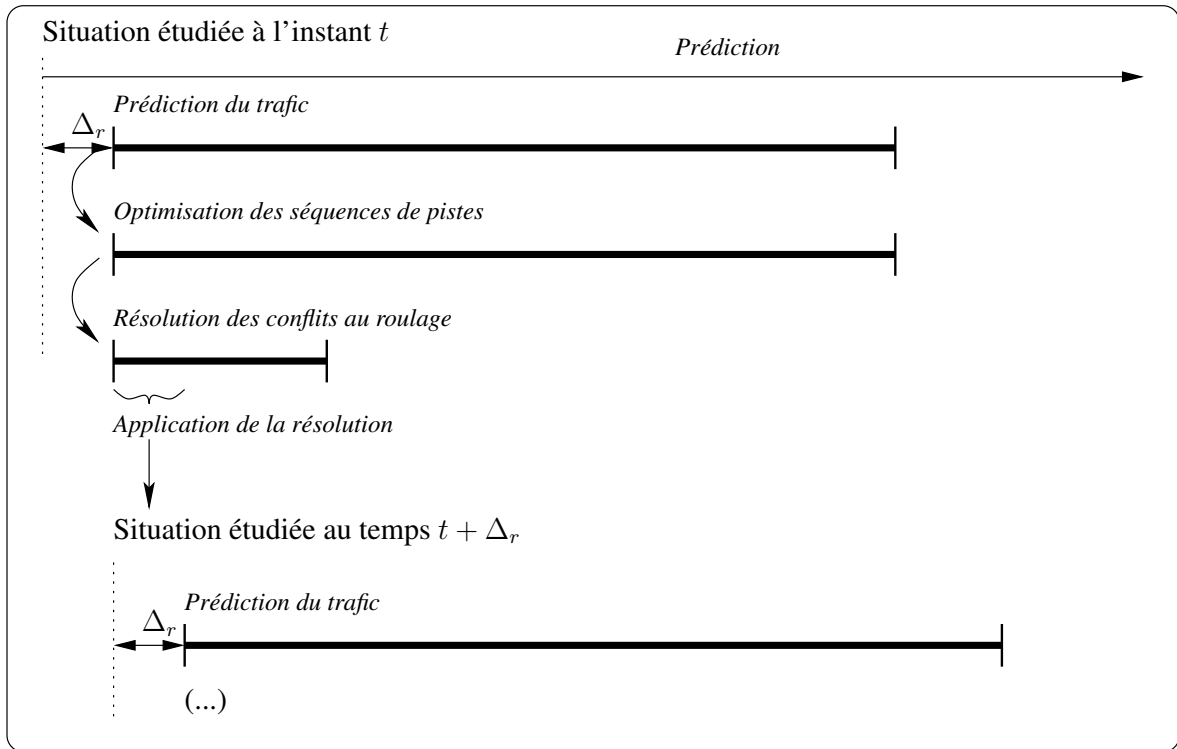


FIG. 6.7 – Simulation par fenêtres glissantes

6.5 Formulation en problème d'optimisation

6.5.1 Présentation

La gestion de la phase de roulage consiste à résoudre tous les conflits entre avions sur l'horizon de prédiction, en leur attribuant des chemins (choisis parmi la liste des chemins possibles), et éventuellement des positions et des temps d'attentes sur ces chemins.

Les travaux précédents ([Gotteland 04]) ont montré que les conflits au roulage sont résolus plus efficacement lorsque les attentes imposées découlent de niveaux de priorité entre avions : dans cette approche, l'objectif est alors d'affecter un chemin et un niveau de priorité à chaque avion, et les trajectoires précises (attentes comprises) des avions sont déduites, dans l'ordre décroissant des niveaux de priorité : l'algorithme III.1 décrit dans la partie 7.2 permet de déterminer les attentes les plus adaptées à imposer à chaque avion, en fonction de son classement.

Cette partie formalise cette approche, en précisant les variables et les contraintes du problème à résoudre, puis en présentant les deux définitions du critère qui seront comparées : l'une sans notion de séquence optimale d'avions à réaliser, l'autre avec.

6.5.2 Variables

Dans l'approche par niveaux de priorité, les variables du problème décrivent le chemin et le rang de chaque avion, de sorte que le classement des avions par priorité est modéré par le choix des chemins : un détour peut ainsi être ordonné à un avion plus prioritaire, pour résoudre un conflit avec un avion qu'il l'est moins, mais qui n'aurait aucune solution sinon.

Le domaine de la variable η_i représentant le chemin de l'avion i sera alors l'intervalle entier $[1; n_{ch}^i]$.

Le rang de l'avion i est décrit par la variable θ_i , dont le domaine est donc l'intervalle $[1; N]$, pour un conflit regroupant N avions.

L'ensemble des variables pour la résolution d'un conflit à N avions est donc décrit par :

$$\forall i \in [1; N] \begin{cases} \eta_i \in [1; n_{ch}^i] \\ \theta_i \in [1; N] \end{cases}$$

6.5.3 Contraintes

Les principales contraintes du problème restent les règles de séparation imposées entre avions, décrites au 6.3.1 :

- une distance minimale de séparation entre tous les avions en mouvement ;
- des temps minimaux de séparation entre chaque mouvement sur chaque piste.

Ces contraintes ne peuvent pas être reliées formellement aux variables du problème : leur vérification nécessite la simulation du trafic sur l'horizon de prédiction avec les chemins et les niveaux de priorités correspondants.

Les autres contraintes sont relatives aux rangs, pour que ces derniers définissent toujours une permutation des N avions (deux avions ne doivent pas avoir le même niveau de priorité) :

$$\forall (i, j) \in [1; N]^2, i \neq j \implies \theta_i \neq \theta_j$$

6.5.4 Critère sans séquence d'avions à réaliser sur les pistes

Le critère à minimiser mis en place dans [Gotteland 04] sert de base à celui défini dans le cadre de cette thèse. Il est défini par la somme des *pénalités* associées à chaque avion. Trois catégories de mouvements sont distinguées pour cela : les arrivées, les départs régulés et les départs non régulés :

- **Pour une arrivée**, la pénalité est une pondération particulière du délai et du détour :

$$\forall i \in \mathcal{A}, p_i = x_{a1}d_i + x_{a2}\epsilon_i$$

Le détour est ainsi au moins aussi pénalisant que les attentes (lorsque $1 \leq x_{a1}$ et $0 \leq x_{a2}$), puisqu'il intervient dans les deux termes.

- **Pour un départ non régulé**, la pénalité p_i se définit par une autre pondération (plus importante, pour favoriser les départs sur les arrivées) des mêmes mesures :

$$\forall i \in \mathcal{D}_C, p_i = x_{c1}d_i + x_{c2}\epsilon_i$$

avec :

$$\begin{cases} x_{c1} > x_{a1} \\ x_{c2} > x_{a2} \end{cases}$$

- **Pour un départ régulé**, l'objectif du respect du créneau imposé par le CFMU vient s'ajouter à la minimisation du délai : la pénalité p_i (pour un départ régulé i) doit être définie non seulement en fonction de son délai et de son détour mais également en fonction de l'écart entre son heure de décollage résultante et son créneau T_{cfmu}^i , en pénalisant fortement l'avance sur le créneau. Il faut alors considérer l'*attente stratégique* ω_i définie au 6.4.4, pour mesurer l'écart entre le délai imposé à l'avion et cette attente stratégique.

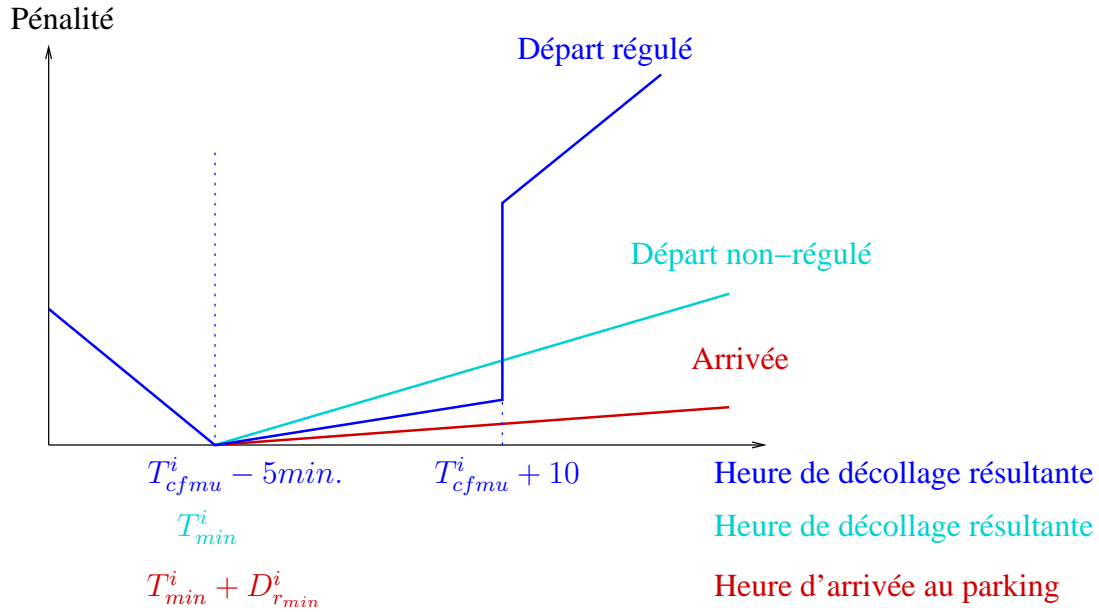


FIG. 6.8 – Fonction de calcul de la pénalité sans prise en compte des séquences de pistes

La pénalité pour l'avion est alors calculée comme suit :

$$\forall i \in \mathcal{D}_{\mathcal{R}} \begin{cases} \text{Si le créneau est respecté : } & p_i = x_{r1}|d_i - \omega_i| + x_{r2}\epsilon_i \\ \text{Sinon :} & p_i = x_{r3}|d_i - \omega_i| + x_{r4}\epsilon_i \end{cases}$$

Pour que le respect des créneaux imposés soit prioritaire sur le délai généré aux avions, il faut donc fixer les valeurs des coefficients x_{r3} et x_{r4} à des valeurs plus grandes que celles des coefficients x_{r1} , x_{r2} , x_{c1} et x_{c2} .

La figure 6.8 donne l'allure générale de la pénalité selon le type de mouvement. En pratique, pour rendre les délais des arrivées moins pénalisants que ceux des départs, et donner priorité au respect des créneaux de décollage imposés par le CFMU, les coefficients peuvent être définis comme suit :

$$\begin{aligned} \text{Pour les arrivées : } & x_{a1} = 1, & x_{a2} & = 1 \\ \text{Pour les départs non-régulés : } & x_{c1} = 4, & x_{c2} & = 4 \\ \text{Pour les départs régulés respectés : } & x_{r1} = 2, & x_{r2} & = 2 \\ \text{Pour les départs régulés ratés : } & x_{r3} = 10, & x_{r4} & = 10 \end{aligned}$$

Le **critère global** P à minimiser s'obtient alors en faisant la somme des pénalités des avions impliqués :

$$P = \sum_{i=1}^N p_i \text{ avec } N, \text{ le nombre d'avions.}$$

6.5.5 Critère visant à réaliser des séquences d'avions sur les pistes

Pour réaliser des séquences d'avions sur les pistes, le critère doit pénaliser l'écart entre l'heure de décollage résultant de la résolution des conflits et l'heure de décollage prévue dans les séquences. De

la même façon que pour les avions régulés dans la partie précédente, l'attente stratégique ω_i définie au 6.4.4 doit être incorporée au critère. Cependant, le délai de chaque avion doit toujours être considéré, afin de maintenir un débit de trafic suffisant : dans le cas contraire, l'arrêt d'un départ en avance par rapport à son heure prévue de décollage ne serait pas pénalisé, ce qui pourrait provoquer rapidement des situations de congestion sur l'aéroport.

En conséquence, le critère visant à réaliser les séquences optimales d'avions est défini par différentes pondérations du délai, du détour et de l'éventuel retard sur les heures de décollage programmées pour les départs. Comme dans la partie précédente, les pondérations dépendent du type de mouvement :

- **Pour une arrivée**, l'heure d'atterrissage est fortement contrainte (flexibilité maximale de ± 30 secondes), donc le respect des séquences n'intervient pas dans le critère :

$$\forall i \in \mathcal{A}, p_i = x_{a1}d_i + x_{a2}\epsilon_i$$

avec x_{a1} et x_{a2} les coefficients déterminants l'importance du délai par rapport au détour.

- **Pour un départ non régulé**, le critère doit prendre en compte l'écart entre l'heure de décollage résultante de la solution étudiée et l'heure de décollage t_i programmée dans la séquence : le retard sur l'heure programmée peut notamment être mesuré par la différence entre les deux, lorsqu'elle est positive.

Le critère s'exprime alors en fonction de cet écart et de la différence entre le délai d_i imposé à l'avion i et son attente stratégique ω_i :

$$\forall i \in \mathcal{D}_C, p_i = \underbrace{x_{c1}|d_i - \omega_i| + x_{c2}\epsilon_i}_{\text{Délai et détour}} + \underbrace{x_{c3} \max(0, T_{min}^i + d_i - t_i)}_{\text{Retard sur l'heure programmée dans la séquence}}$$

Les coefficients x_{c1} et x_{c2} déterminent comme précédemment la pondération entre le détour et le délai, tandis que le coefficient x_{c3} permet de fixer l'importance du respect des séquences à réaliser par rapport aux délais générés.

- **Pour un départ régulé**, comme pour le critère sans prise en compte des séquences, la pénalité doit être fonction du respect du créneau : tant que l'avion n'est pas en retard sur son créneau, la pénalité est une pondération comparable à celle définie pour un départ non régulé ; dans le cas contraire, la pénalité ne dépend plus de l'heure programmée dans la séquence mais de l'heure imposée par le CFMU, afin de donner priorité au créneau : elle s'exprime alors par l'écart au créneau imposé :

- Si l'avion i peut décoller avant la fin de son créneau :

$$p_i = \underbrace{x_{r1}|d_i - \omega_i| + x_{r2}\epsilon_i}_{\text{Délai et détour}} + \underbrace{x_{r3} \max(0, \max(t_0, T_{pmin}^i) + D_{rmin}^i + d_i - t_i)}_{\text{Retard sur l'heure programmée dans la séquence}}$$

- Si l'avion i est en retard sur son créneau :

$$p_i = x_{r4} \underbrace{(\max(t_0, T_{pmin}^i) + D_{rmin}^i + d_i - T_{cfmu}^i)}_{\text{Retard sur l'heure imposée par le CFMU}}$$

Les délais et les détours sont pondérés par les coefficients x_{r1} , x_{r2} et x_{r3} et le coefficient x_{r4} fixe l'importance du respect des créneaux imposés par le CFMU.

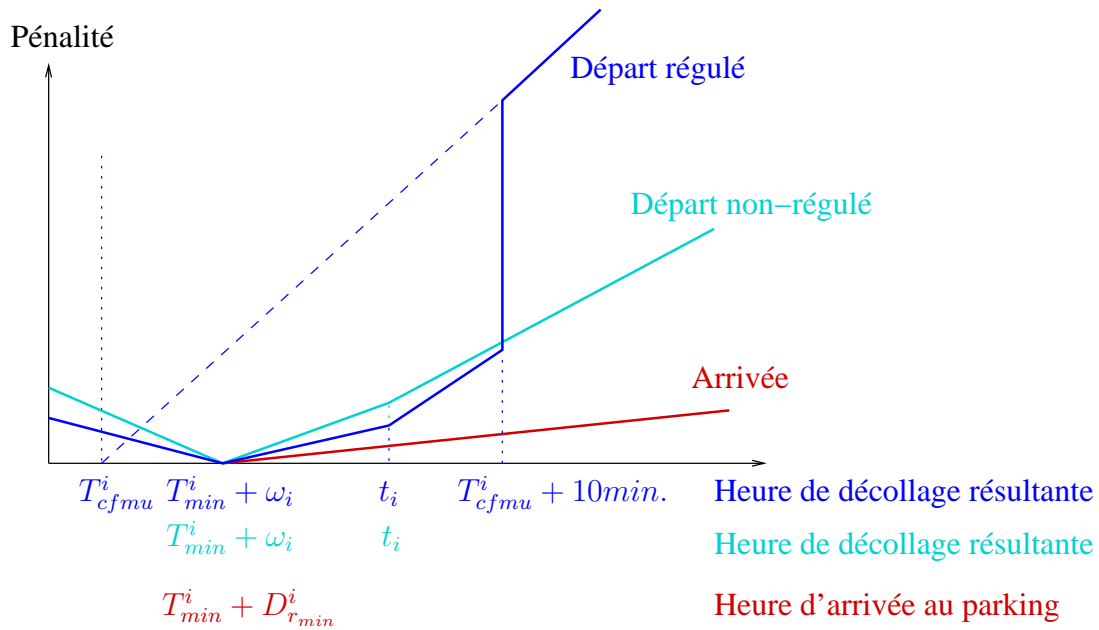


FIG. 6.9 – Fonction de calcul de la pénalité avec prise en compte des séquences de pistes

La figure 6.9 illustre le calcul de la pénalité en fonction du type de mouvement. En pratique, toujours pour rendre les délais des arrivées moins pénalisants que ceux des départs et pour donner priorité à l'application des heures de décollage calculées pour les départs régulés, les coefficients peuvent être définis comme suit :

$$\begin{aligned}
 \text{Pour les arrivées : } & x_{a1} = 1, \quad x_{a2} = 1 \\
 \text{Pour les départs non-régulés : } & x_{c1} = 4, \quad x_{c2} = 4, \quad x_{c3} = 1 \\
 \text{Pour les départs régulés respectés : } & x_{r1} = 2, \quad x_{r2} = 2, \quad x_{r3} = 2 \\
 \text{Pour les départs régulés ratés : } & x_{r4} = 4,
 \end{aligned}$$

Le **critère global** à minimiser s'exprime toujours par la somme des pénalités associées aux avions :

$$P = \sum_{i=1}^N p_i \text{ avec } N, \text{ le nombre d'avions.}$$

6.5.6 Taille de l'espace de recherche

Pour un problème à N avions et avec l'approche par classement, une solution correspond à une affectation de chemins aux avions (n_{ch}^N possibilités au maximum) et à un choix de permutation ($N!$ possibilités). La taille de l'espace de recherche s'exprime par :

$$|\xi| \leq n_{ch}^N \times N!$$

Le problème est donc fortement combinatoire avec le nombre d'avions :

$$\text{Pour } \begin{cases} N & = 15 \text{ avions} \\ n_{ch} & = 10 \text{ chemins par avion} \end{cases}$$

On obtient : $|\xi| < 10^{27}$ possibilités.

6.6 Conclusion

Ce chapitre montre que la formulation du problème de résolution de conflits au roulage nécessite un contexte de simulation du trafic au sol, intégrant une description détaillée des voies de circulation de l'aéroport, un modèle de prédiction de trajectoires capable de prendre en compte les incertitudes sur les vitesses de roulage, et enfin un modèle de détection de conflits permettant d'identifier les groupes d'avions dont les trajectoires interfèrent.

Dans ce contexte, l'approche par classement des avions permet de formaliser deux types de variables pour le problème : d'une part celles correspondant à l'affectation de chemins aux avions, d'autres part celles décrivant leurs niveaux de priorité. Deux critères sont alors mis en place afin d'être comparés : le premier consiste à minimiser les délais des avions non régulés en essayant de respecter au mieux les créneaux imposés aux avions régulés ; tandis que le second vise à réaliser des séquences pré-calculées d'avions sur les pistes, avec l'objectif de réduire davantage les délais et d'améliorer le respect des créneaux imposés, ainsi que la prévisibilité des heures de décollage.

Chapitre 7

Résolution des conflits au roulage

7.1 Introduction

Le problème de résolution des conflits au roulage tel qu'il est formulé au chapitre précédent est associé à un espace de recherche dont la taille est fortement combinatoire avec le nombre d'avions et le nombre de chemins considérés par avion. Combinée à la prise en compte des incertitudes sur les vitesses de roulage (qui complexifie la détection des conflits entre ces avions), la résolution optimale des conflits par une méthode exacte n'est pas envisageable, même sur un horizon de prédiction limité de l'ordre de 5 minutes.

Les méthodes de résolution de conflits présentées dans ce chapitre ont donc pour objectif d'affecter une trajectoire sans conflit à chaque avion, même si cela ne représente pas forcément une solution optimale au problème formulé dans sa forme la plus générale. Les deux critères présentés au chapitre précédent doivent cependant permettre de comparer les solutions obtenues avec et sans prise en compte de séquences optimales d'avions sur les pistes. Deux méthodes de résolution sont présentées pour cela : la première est déterministe mais s'applique à une simplification du problème initial, l'autre est stochastique (basée sur un algorithme évolutionnaire) mais s'applique au problème global.

7.2 Résolution par une méthode déterministe

7.2.1 Classement des avions

Pour pouvoir trouver des solutions aux conflits entre avions avec une méthode de résolution déterministe, il est nécessaire de simplifier le problème formulé au chapitre précédent. Une simplification intuitive consiste à considérer les avions les uns après les autres : dans ce contexte, le problème élémentaire à résoudre est celui d'un seul avion devant en éviter d'autres (dont les trajectoires ont déjà été calculées et deviennent des contraintes du problème à résoudre pour l'avion). La résolution du problème à N avions se fait alors par itérations de la résolution élémentaire pour un avion, dans un ordre prédéterminé.

Le classement initial des avions correspond ainsi à des niveaux de priorité : le premier avion suit sa trajectoire optimale, tandis que les suivants ont de plus en plus de contraintes. Ce classement initial joue donc un rôle majeur pour cette méthode de résolution de conflits : beaucoup de classements correspondent à un problème sans solution pour un avion, si par exemple un avion est moins bien classé qu'un avion se trouvant derrière lui.

Données en entrée	
Notation	Description
N	Nombre total d'avions.
D_{rmin}^i	Durée minimale de roulage de l'avion i (sans attente et à vitesse maximale).
H_s	Horizon de prédiction pour le calcul des séquences de pistes.
H_r	Horizon de prédiction pour la phase de roulage.
Δ_r	Pas de rafraîchissement.
t_i	Heure d'accès à la piste de l'avion i prévue dans la séquence de piste.
n_{ch}^i	Nombre de chemins disponibles pour l'avion i .
n_{att}	Nombre maximal d'attentes pour chaque avion.
ω_i	Attente stratégique calculée pour l'avion i .
Variables	
Notation	Description
η_i	Chemin choisi pour l'avion i .
θ_i	Niveau de priorité de l'avion i .
p_i	Pénalité pour l'avion i
Algorithme génétique	
Notation	Description
T_p	Taille de la population.
f_q^i	Fitness globale de l'individu i
f_p^a	Fitness partielle de l'avion a au sein d'un individu.
K	Constante ajouté à la pénalité d'un avion lorsqu'il est en conflit avec un autre.
Méthode déterministe	
Notation	Description
θ'_i	Niveau de priorité pour l'avion i exprimé en heure absolue.
δ_{prio}	Réduction du niveau de priorité des arrivées après atterrissage (en minutes).

TAB. 7.1 – Table des notations

Dans le contexte de séquences d'avions à réaliser sur les pistes, cette méthode possède un grand avantage : le lien entre les séquences d'avions à réaliser sur les pistes et la phase de roulage peut s'effectuer simplement, en prenant un classement initial en cohérence avec ces séquences. Dans le cas d'un aéroport avec plusieurs pistes comme Roissy, les séquences souhaitées sur les pistes ne définissent pas directement de relation d'ordre total entre les avions, mais un classement total peut être obtenu, en classant les avions par heures de décollage et d'atterrissage programmées.

Une modification de ce classement peut de plus être opérée, afin de réduire le niveau de priorité des arrivées, une fois qu'elles ont atterri : en effet, lors de la phase de roulage, les arrivées sont souvent considérées moins prioritaires que les départs, car les retards imputés en début de vol (avant même le décollage) sont considérés plus problématiques par les opérationnels. Une fois qu'elles ont libéré l'aire de piste, les arrivées peuvent donc avoir un niveau de priorité inférieur à celui de l'ensemble des départs en cours : pour ce faire, un paramètre δ_{prio} détermine la perte du niveau de priorité des arrivées, une fois posées.

Le classement initial des avions est donc défini en fonction des heures $(\theta'_i)_{1 \leq i \leq N}$ suivantes :

$$\begin{cases} \forall i \in \mathcal{D}, \theta'_i = t_i \\ \forall i \in \mathcal{A}, \begin{cases} t_i < t_0 \Rightarrow \theta'_i = t_i + \delta_{prio} \\ t_0 \leq t_i \Rightarrow \theta'_i = t_i \end{cases} \end{cases}$$

Une fois ce classement défini, la résolution de l'ensemble des conflits peut s'effectuer par l'intermédiaire de deux algorithmes de type branch & bound :

- le premier consiste à explorer l'arbre des trajectoires possibles de l'avion le long de l'un de ses chemins ;
- la second correspond à des itérations du premier, sur chaque chemin et pour chaque avion.

7.2.2 Calcul de la trajectoire d'un avion sur l'un de ses chemins

La recherche de la meilleure trajectoire pour un avion devant en éviter d'autres, en suivant l'un de ses chemins, peut s'exprimer par un parcours d'arbre :

- La racine de l'arbre correspond à la position (ou zone d'incertitude) initiale de l'avion, au début de l'horizon de prédiction).
- Chaque nœud correspond à une zone d'incertitude datée de l'avion sur son chemin, associée au nombre de positions d'attentes déjà effectuées (car le nombre d'attentes possibles reste limité). Il se représente par un quadruplet (z, t, n_a, λ) :
 - z décrit la zone d'incertitude de l'avion (elle est représentée par l'indice des positions extrêmes (p_{min}, p_{max}) dans le chemin) ;
 - t est l'heure à laquelle cette zone d'incertitude est atteinte ;
 - n_a représente le nombre d'attentes déjà effectuées par l'avion,
 - λ indique si l'avion avance ($\lambda = 1$) ou est en position d'attente ($\lambda = 0$).
- Chaque nœud peut avoir au maximum deux fils, représentant les zones d'incertitudes possibles et non conflictuelles de l'avion au pas de temps suivant, en fonction de l'action effectuée par l'avion à l'instant t [Gotteland 04] :
 1. soit l'avion attend, à condition que le nombre maximal d'attentes ne soit pas atteint et que l'attente ne génère pas de conflit avec les avions déjà traités ;
 2. soit l'avion avance, si cela ne génère pas de conflit.
- Les feuilles dont l'heure correspond à la fin de l'horizon de prédiction représentent une solution admissible : elles correspondent à une trajectoire sans conflit sur l'horizon de prédiction, depuis la position initiale de l'avion.

```

Choix_Attentes( $\mathcal{M}, m, \eta, P_{max}$ ) :
    ( $\mathcal{M}$  : avions devant être évités)
    ( $m$  : avion dont la trajectoire doit être calculée)
    ( $\eta$  : chemin étudié)
    ( $P_{max}$  : pénalité maximale souhaitée)
    (Trajectoire trouvée)
    (Pénalité maximale souhaitée sur le chemin  $\eta$ )
    soit  $\gamma : \emptyset$ 
    soit  $Borne : Penalite\_Chemin(P_{max}, m, \eta)$ 
    Exploration( $z, t, n_a, \lambda$ )
    si Sans_Conflit( $\mathcal{M}, z, t$ ) et  $Penalite(m, t) < Borne$  alors
        (Nouvelle solution trouvée)
        si  $t = H_r$  alors
             $\gamma \leftarrow Trajectoire(m)$ 
             $Borne \leftarrow Penalite(m)$ 
        sinon si  $Attente(m) < Attente\_Strategique(m)$  et  $n_a < n_{att}$  alors
            Exploration( $Attendre(z, t), t + 1, n_a + \lambda, 0$ )
            Exploration( $Avancer(z, t), t + 1, n_a, 1$ )
        sinon
            Exploration( $Avancer(z, t), t + 1, n_a, 1$ )
            si  $n_a < n_{att}$  alors Exploration( $Attendre(z, t), t + 1, n_a + \lambda, 0$ )
    Affecter_Chemin( $m, \eta$ )
    Exploration( $z_0, 0, 0, 1$ )
    retourner  $\gamma$ 

```

Algorithme III.1: Choix_Attentes : meilleure trajectoire pour un avion, sur l'un de ses chemins

L'exploration de cet arbre s'effectue par un parcours en profondeur d'abord, et peut être limitée par une valeur de pénalité maximale pour l'avion. Dans le cas où l'avion est associé à une attente stratégique (cf. 6.4.4), il doit attendre dès que possible : les nœuds fils correspondants à l'attente de l'avion sont alors explorés en premier. Dans le cas contraire (ou dès que l'attente stratégique est atteinte), l'avion doit chercher à avancer et les nœuds correspondants sont alors explorés en premier.

Lorsque l'algorithme trouve une solution, la pénalité de l'avion correspondant à cette dernière est définie comme une borne pour les itérations suivantes. Lors de l'exploration de chaque nœud, la pénalité minimale résultante de la trajectoire courante est comparée à cette borne et la branche n'est pas explorée si la borne est dépassée. Lorsqu'une meilleure solution est trouvée, la borne est mise à jour.

Ce mode d'exploration est mis en œuvre dans l'algorithme récursif III.1 :

- La fonction `Penalite_Chemin` renvoie la pénalité maximale relative au chemin η de l'avion ;
- Les fonctions `Avancer` et `Attendre` calculent la zone d'incertitude de l'avion au pas de temps suivant, suivant l'action effectuée.
- La fonction `Attente` renvoie le temps d'attente cumulé imposé à l'avion sur l'horizon de prédiction.
- La fonction `Attente_Strategique` renvoie l'attente stratégique ω_i calculée pour l'avion i .

7.2.3 Résolution des conflits d'un groupe d'avions

Le second algorithme correspond également à un parcours d'arbre de type branch & bound : il consiste à appliquer l'algorithme III.1 pour chaque chemin possible de chaque avion afin de trouver les meilleures trajectoires possibles, dans l'ordre décrit au 7.2.1. Chaque nœud de l'arbre correspond à l'affectation d'un chemin à un avion et chaque niveau de profondeur de l'arbre correspond donc à un nouvel avion à traiter.

<pre> Choix_Chemin(\mathcal{M}, m) : soit $\gamma : \emptyset$ soit $P_{max} : +\infty$ pour chaque η, chemin de m faire soit $\gamma' : \text{Choix_Attentes}(\mathcal{M}, m, \eta, P_{max})$ si $\gamma' \neq \emptyset$ alors $\gamma \leftarrow \gamma'$ $P_{max} \leftarrow \text{Penalite}(m, H_r)$ retourner γ </pre>	<p>(Meilleur chemin pour m devant éviter \mathcal{M}) (meilleure solution trouvée)</p>
<pre> Resoudre($(m_i)_{1 \leq i \leq N}$) : soit $i : 1$ tant que $i \leq N$ soit $\gamma : \text{Choix_Chemin}((m_j)_{1 \leq j < i}, m_i)$ si $\gamma = \emptyset$ alors soit $j : \text{Max}\{j < i \mid m_j \text{ est en conflit avec } m_i\}$ Insérer m_i devant m_j dans $(m_k)_{j \leq k \leq N}$ si $(m_i)_{1 \leq i \leq N}$ a déjà été exploré alors lever ECHEC sinon $i \leftarrow j$ sinon $i \leftarrow i + 1$ </pre>	<p>(Résout un conflit à N avions)</p>

Algorithme III.2: Resoudre : calcule les trajectoires sans conflit des avions

L'exploration d'un nœud consiste à parcourir l'ensemble des chemins possibles (du plus court au plus long) jusqu'à trouver une solution acceptable. Lorsqu'une trajectoire sans conflit est trouvée pour un avion, la solution obtenue fournit une borne sur le délai imposé à cet avion. Cette borne est alors utilisée pour limiter l'exploration des chemins suivants de l'avion (ces derniers sont envisagés avec la pénalité maximale correspondant à la borne, en fonction du chemin exploré).

Si aucune solution n'est trouvée pour un avion i , un nouveau classement est construit :

- L'avion j le moins bien classé et en conflit avec l'avion i est identifié (parmi les avions plus prioritaires que i).
- Un nouveau classement est formé, en insérant l'avion i juste avant l'avion j .
- Si le classement ainsi formé n'a jamais été exploré, la méthode de résolution est appliquée sur ce classement. Sinon, cela correspond à un cas d'échec de cette méthode de résolution de conflit.

Cette exploration est mise en œuvre dans l'algorithme III.2.

7.3 Algorithmes évolutionnaires

7.3.1 Principes

Les algorithmes génétiques se basent sur la théorie de l'évolution génétique darwinienne afin de faire *évoluer* des solutions d'un problème pour les faire *s'adapter* au critère à optimiser [Mitchell 96, Eiben 03].

Concrètement, l'espace de recherche est parcouru par une population d'individus. Chaque individu est associé à un codage de données spécifique au problème à résoudre. Ce codage peut être une chaîne d'entiers, de réels ou encore une combinaison des deux [Janikow 91] et représente le *génotype* de

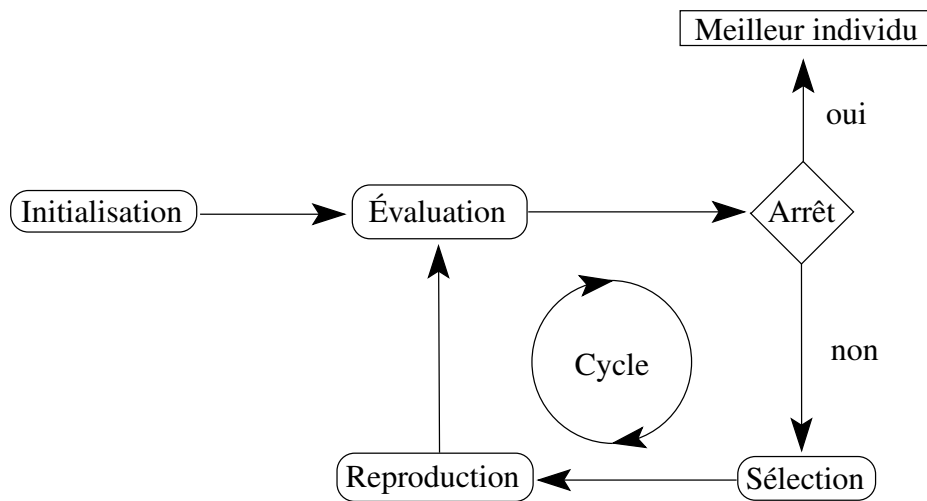


FIG. 7.1 – Processus d’itération d’un algorithme génétique

l’individu. Chaque individu représente un point de l’espace de recherche, appelé *phénotype* (dédit du génotype). Plusieurs génotypes peuvent donc correspondre à un même phénotype.

Les individus vont alors se reproduire afin de former de nouveaux individus, qui ont des chances d’être plus adaptés grâce à l’héritage génétique transmis par leurs *parents*. Certaines *mutations* peuvent également être effectuées aléatoirement, sur quelques individus afin de diversifier la population et ainsi d’étendre l’exploration de l’espace des solutions.

Au fil des générations, les individus les plus adaptés se reproduisent en plus grand nombre et la population peut alors converger vers les régions de l’espace de recherche où se trouvent les optima recherchés.

La *fonction d’évaluation* est essentielle à l’algorithme : elle permet de juger si un individu est adapté ou non. Elle correspond à une note attribuée à chaque individu afin de quantifier son adaptation au critère.

Ce mode d’exploration permet de parcourir de vastes espaces de recherche qui ne sont pas toujours abordables par des méthodes exactes. Les algorithmes génétiques ont donc la particularité de pouvoir explorer ces grands espaces mais n’assurent pas de trouver un optimum global.

7.3.2 Cycle d’évolution

La figure 7.1 illustre les itérations d’un algorithme génétique :

- L’**initialisation** consiste à définir la population initiale, en général de façon aléatoire (un biais pouvant être envisagé selon des heuristiques établies, pour avoir de meilleures chances d’obtenir une génération initiale de bonne qualité). Son hétérogénéité est un caractère important, car cette population sera la *graine* de toutes les générations descendantes. Il est donc important d’assurer que tout point de l’espace de recherche ait une chance non nulle d’être représenté dans la population initiale.
- La simulation du **cycle** de l’évolution est constituée des phases suivantes :
 - L’**évaluation** consiste à attribuer une note à chaque individu. Cette note est aussi appelée *fonction d’adaptation* ou *fitness* et doit être cohérente avec le critère du problème : les individus représentant des solutions optimales doivent posséder les meilleures notes.

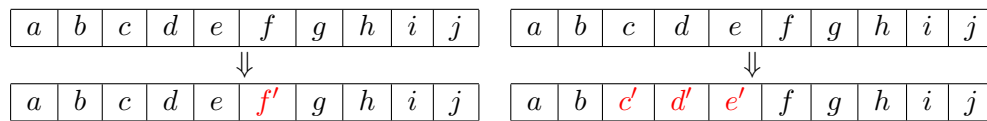


FIG. 7.2 – Exemple de mutation

- Lorsque l'évaluation est effectuée, il y a une possibilité d'**arrêt** du cycle. Différents critères d'arrêt peuvent être mis en place selon le problème. On distingue notamment : arrêt lorsque la meilleure note possible est obtenue ou lorsqu'un nombre maximal de générations est atteint. Si le cycle s'arrête, le ou les meilleur(s) individu(s) de la population courante est(sont) extrait(s). Il est souvent utile d'appliquer une méthode d'optimisation locale en fin de convergence pour affiner les meilleures solutions trouvées, car les algorithmes génétiques sont efficaces pour trouver certaines régions intéressantes mais n'assurent pas que la solution trouvée soit un optimum, même local.
- La **sélection** est la phase où les individus qui vont être utilisés comme parents ou qui vont simplement être conservés (avec ou sans mutation) dans la génération suivante sont choisis. Si la taille de la population est fixe, certains individus (les moins bons) sont exclus de la population au cours de la sélection. Il existe de nombreuses méthodes pour choisir quels individus doivent être retenus. On distingue, parmi les méthodes les plus répandues décrites dans [Goldberg 89] :
 - La sélection par *roulette* : on symbolise cette sélection par une roue dont chaque individu possède une portion proportionnelle à son adaptation : un tirage aléatoire est alors effectué sur cette roue, ce qui désigne un individu. Chaque individu a donc une chance proportionnelle à sa fitness d'être sélectionné.
 - La sélection par *rang* : les individus sont triés par fitness croissante et la sélection s'effectue par roulette, appliquée à leurs rangs dans ce classement et non plus à leurs fitness.
 - La sélection par *tournoi* : les individus sont confrontés deux à deux. Le meilleur de chaque paire est alors sélectionné.
 - La sélection *uniforme* : le choix des individus se fait totalement aléatoirement. Chacun des n individus a donc une probabilité $\frac{1}{n}$ d'être sélectionné.
 Un principe d'*élitisme* est également fréquemment introduit dans le processus de sélection, consistant à conserver systématiquement le ou les meilleurs individus de la population courante dans la génération suivante, sans lui faire subir de croisement ou de mutation. Ce principe confère à l'algorithme la propriété de croissance monotone de la fitness du meilleur individu.
- La **reproduction** utilise les individus choisis précédemment pour générer de nouveaux individus soit par croisement entre deux parents, soit par mutation d'un seul ou encore par simple copie, comme cela est décrit dans partie suivante.

7.3.3 Opérateurs de variation

Le terme d'*opérateurs de variation* (aussi appelés *opérateurs génétiques*) désigne les fonctions de croisement et de mutation des individus. Ces fonctions dépendent avant tout de la structure de données utilisée pour coder chaque individu.

La **mutation** d'un individu modifie aléatoirement tout ou une partie du génotype d'un individu. Cette modification a pour but de diversifier la population afin d'assurer qu'elle ne se retrouve pas

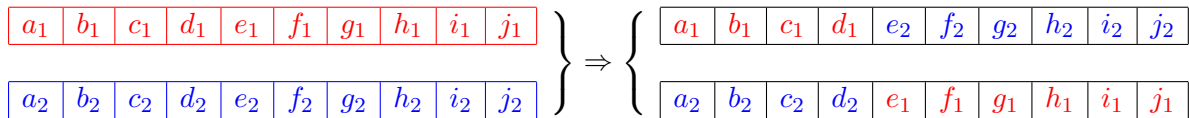


FIG. 7.3 – Exemple de croisement

enfermée dans un optimum local et que la recherche de solutions puisse toujours s'étendre. Comme le montre la figure 7.2, la mutation peut s'opérer sur différentes parties du génotype d'un individu. Sur cette figure, le génotype de l'enfant correspond au génotype du parent avec un changement aléatoire de la valeur de f dans le premier cas, et un changement aléatoire de la valeur de c , d et e dans le second cas.

Le **croisement** de deux individus génère deux nouveaux individus dont les caractéristiques sont obtenues par un mélange entre celles des deux parents. De nombreux types d'opérateurs de croisement peuvent être élaborés, en fonction du codage des données utilisés. La figure 7.3 illustre un exemple de croisement : les valeurs des $(a_i, \dots, j_i)_{1 \leq i \leq 2}$ ne changent pas mais la distribution différente de ces valeurs permet de définir de nouveaux individus.

Les opérateurs de mutation et de croisement peuvent également être définis en fonction des contraintes du problème : l'optimum recherché se trouve le plus souvent à la frontière de ce dernier et les opérateurs peuvent donc favoriser ou se restreindre à l'exploration de cette frontière [Schoenauer 97].

Les taux de croisement et de mutation représentent la proportion de la population qui sera ainsi renouvelée à chaque génération. La somme de ces deux taux est appelé *taux de renouvellement*. Si le taux de renouvellement n'est pas de 100% et que la taille de la population est fixe, une proportion d'individus est conservée sans modification pour la génération suivante.

7.3.4 Fonction d'évaluation

La fonction d'évaluation (ou *fitness*) joue un rôle majeur dans la convergence de l'algorithme génétique : la note fournie par cette fonction permet d'orienter la sélection, en vue de conserver les individus qui correspondent le plus au critère visé.

Lorsque les contraintes du problème ne s'expriment pas simplement en fonction des variables, les individus générés au fil des générations ne les respectent pas forcément : dans ce cas, la fonction d'évaluation doit mesurer (en plus du critère) les violations de contraintes [Michalewicz 96] et les pénaliser. Une méthode souvent utilisée consiste à définir un seuil d'évaluation pour distinguer les individus situés en dehors du domaine admissible : l'évaluation de ces derniers est systématiquement inférieure au seuil défini, et peut être inversement proportionnelle au nombre de violations de contraintes. L'avantage de cette méthode est de pouvoir conserver des individus ne respectant pas les contraintes, mais pouvant parfois générer des enfants intéressants (car les optima recherchés sont souvent en bordure du domaine admissible).

7.3.5 Améliorations classiques

La convergence d'un algorithme génétique peut être fortement améliorée par des méthodes de *scaling* et de *sharing* :

Scaling

Le scaling [Michalewicz 92] est une mise à l'échelle de la fonction d'évaluation. Le but est d'augmenter ou de diminuer artificiellement les écarts d'évaluation entre les individus afin d'ajuster la sélection. De nombreuses fonctions de scaling peuvent être envisagées. Les deux catégories de scaling les plus utilisées sont :

1. Le scaling linéaire, défini comme une fonction affine de la fitness initiale. Pour chaque individu, on aura :

$$f_i^s = a f_i + b$$

$$\text{avec } \left\{ \begin{array}{l} f_i \text{ la fitness de l'individu } i, \\ f_i^s \text{ la nouvelle fitness de l'individu } i, \text{ après scaling,} \\ a > 0 \end{array} \right.$$

Le paramètre a permet d'augmenter ou de réduire la pression sélective des individus : une exploration plus large de l'espace de recherche est effectuée dans le cas où $a < 1$ alors que si $a > 1$, les individus ayant les meilleures fitness sont favorisés. Les valeurs de a et b peuvent varier au fil des générations.

2. Le scaling exponentiel se définit par une fonction exponentielle de la fitness :

$$f_i^s = (f_i)^k$$

$$\text{avec } \left\{ \begin{array}{l} f_i \text{ la fitness de l'individu } i, \\ f_i^s \text{ la nouvelle fitness de l'individu } i, \text{ après scaling.} \\ k > 0 \end{array} \right.$$

- Lorsque k tend vers 0, l'algorithme génétique effectue une exploration de plus en plus aléatoire de l'espace de recherche.
- Lorsque k se rapproche de 1, le scaling devient neutre ;
- Lorsque k augmente, la sélection des individus ayant la meilleure fitness est renforcée.

Sharing

Le sharing [Goldberg 87] a pour objectif de diversifier la population afin d'éviter qu'elle ne converge trop rapidement vers un optimum local. Pour cela, la fitness des individus est modifiée en fonction de leur densité dans l'espace de recherche : plus les individus sont concentrés au même endroit, plus ils sont pénalisés. Le sharing nécessite donc la définition d'une distance d entre les points de l'espace de recherche, pour quantifier l'éloignement des individus.

Une première façon de définir le sharing consiste à diviser la fitness de chaque individu par une valeur représentant la densité de la population dans son voisinage :

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

$$\text{avec } : \left\{ \begin{array}{ll} S(d) = 1 - \left(\frac{d(x_i, x_j)}{\sigma_{share}}\right)^\alpha & \text{si } d(x_i, x_j) < \sigma_{share} \\ S(d) = 0 & \text{si } d(x_i, x_j) \geq \sigma_{share} \end{array} \right.$$

$$\text{et } \left\{ \begin{array}{l} \sigma_{share} : \text{distance d'influence d'un individu} \\ \alpha : \text{intensité du sharing} \end{array} \right.$$

- 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 est fortement corrélée à la définition de la distance d pour le problème traité).
- Le paramètre α définit l'intensité du sharing : plus sa valeur est élevée, plus les individus proches se pénalisent.

En pratique, ce type de sharing est efficace mais nécessite le calcul des distances entre chaque paire d'individus de la population : sa complexité en $\mathcal{O}(N^2)$ peut s'avérer trop pénalisante, par rapport aux autres opérations de l'algorithme dont la complexité est en général en $\mathcal{O}(N)$.

Sharing clusterisé

Le sharing clusterisé [Yin 93] est une adaptation du sharing précédent, plus économique en temps de calcul, car pouvant être implémenté en $\mathcal{O}(N \log N)$ opérations : la population est partitionnée en groupes d'individus (appelées *clusters*) et la fitness de chaque individu est modifiée en fonction de la taille du cluster auquel il appartient et de sa distance au centre du cluster. Ce type de sharing nécessite donc la définition de barycentres entre les individus (en plus d'une distance d) :

$$f_i^s = \frac{f_i}{n_i \left(1 - \left(\frac{l_i}{2d_{max}}\right)^\alpha\right)}$$

avec

avec	f_i la fitness de l'individu i ,
	f_i^s la nouvelle fitness de l'individu i après sharing,
	n_i le nombre d'individus dans le même cluster que i .
	l_i la distance entre l'individu i et le centre de son cluster.
	α l'intensité du sharing.

Le partitionnement en clusters est défini en fonction de deux paramètres d_{min} et d_{max} :

- Initialement, chaque individu *est* un cluster.
- Le paramètre d_{max} représente la distance minimale entre un individu et le centre du cluster le plus proche. Lorsqu'un individu est trop proche d'un cluster, il y est incorporé (et le centre du cluster est alors recalculé).
- Le paramètre d_{min} représente la distance minimale entre deux centres de clusters : lorsque deux clusters sont trop proches, ils sont fusionnés et le centre du cluster formé devient le barycentre des deux centres précédents.

La définition des paramètres d_{min} et d_{max} est donc là encore fortement corrélée à la distance d utilisée sur l'espace de recherche.

Sharing clusterisé adaptatif

Le sharing clusterisé adaptatif [Alliot 96] est une technique de sharing clusterisé dans laquelle les paramètres d_{min} et d_{max} s'adaptent automatiquement à la population et à la distance définie : le principe est d'augmenter le nombre de clusters lorsque que beaucoup d'entre eux contiennent de *bons* individus et de diminuer leur nombre dans le cas contraire. Ce type de sharing se définit en fonction de trois nouveaux paramètres, dont les valeurs sont à fixer entre 0 et 1, mais dont le réglage s'avère plus généralisable que celui de d_{min} et d_{max} :

- **Le taux de sharing** τ_s caractérise les *bons* individus et les *bons* clusters. Les *bons* individus sont définis comme ceux dont l'évaluation relative à celle du meilleur individu de la population est supérieure au taux. Les *bons* clusters sont ceux qui contiennent au moins un *bon* individu.

7.4. ALGORITHME GÉNÉTIQUE POUR LA RÉOLUTION DES CONFLITS AU ROULAGE 85

- **Deux autres seuils** $S_1 < S_2$ déterminent les cas où les valeurs de d_{min} et d_{max} doivent être adaptées : lorsque la proportion de *bons* clusters est inférieure au seuil S_1 , les valeurs de d_{min} et d_{max} sont augmentées pour favoriser la réunion de clusters ; lorsqu’au contraire la proportion de *bons* clusters dépasse le seuil S_2 , les valeurs de d_{min} et d_{max} sont diminuées pour augmenter le nombre de clusters.

En pratique, les valeurs suivantes se sont souvent montrées efficaces sur les problèmes de résolution de conflits [Granger 02] :

$$\begin{cases} \tau_s = 0.9 \\ S_1 = 0.75 \\ S_2 = 0.85 \end{cases}$$

7.3.6 Problèmes partiellement séparables

Certains types de problèmes d’optimisation possèdent la propriété d’être *partiellement séparables* [Durand 94, Durand 98]. Ces problèmes sont caractérisés par un critère à minimiser en fonction de n variables et ce critère peut s’exprimer comme la somme de m fonctions $(F_i)_{1 \leq i \leq m}$, chacune ne dépendant que d’un sous-ensemble de variables $\{x_{j_{i1}}, x_{j_{i2}}, \dots, x_{j_{ik}}\}$ parmi les n :

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^m F_i(x_{j_{i1}}, x_{j_{i2}}, \dots, x_{j_{ik}})$$

Pour $l \in \{1, \dots, n\}$, on définit alors l’ensemble E_l des indices i tels que x_l est une variable de F_i :

$$\forall l \in \{1, \dots, n\}, E_l = \{i \in \{1, \dots, m\} \mid x_l \in \{x_{j_{i1}}, x_{j_{i2}}, \dots, x_{j_{ik}}\}\}$$

Dans le cas où le génotype d’un individu est composé directement des n variables (x_1, x_2, \dots, x_n) , il est alors possible de définir des fonctions de *fitness partielles* notées $(G_l)_{1 \leq l \leq n}$ qui s’appliquent aux n variables et qui représentent la contribution au critère de la variable x_l concernée :

$$\forall l \in \{1, \dots, n\}, G_l(x_1, x_2, \dots, x_n) = \sum_{i \in E_l} F_i(x_{j_{i1}}, x_{j_{i2}}, \dots, x_{j_{ik}})$$

Cette fitness partielle permet alors de définir des opérateurs de variations adaptés au problème, en augmentant la probabilité que ces opérateurs modifient une variable x_l lorsque la valeur de G_l est forte.

7.4 Algorithme génétique pour la résolution des conflits au roulage

7.4.1 Codage des données

Dans le cadre de la résolution des conflits entre avions au roulage et en accord avec l’approche par classement décrite au chapitre précédent, chaque individu doit représenter l’ensemble des chemins et des rangs choisis pour chaque avion :

$$\forall i \in \{1, \dots, N\}, \begin{cases} \eta_i \in [1; n_{ch}^i] & \text{le chemin choisi pour l’avion } i ; \\ \theta_i \in [1, N] & \text{le rang de l’avion } i \text{ dans le classement.} \end{cases}$$

$$\text{avec } \begin{cases} n_{ch}^i & \text{le nombre de chemins possibles pour l’avion } i. \\ N & \text{le nombre total d’avions.} \end{cases}$$

Le *génotype* de chaque individu est alors codé par l'ensemble des couples (η_i, θ_i) :

$$\text{individu} = \{(\eta_1, \theta_1); (\eta_2, \theta_2); \dots; (\eta_n, \theta_n)\}$$

Chaque individu représente donc une permutation du classement des avions (via les rangs) et un ensemble de chemins qui leurs sont affectés. Cette approche par permutation permet alors d'assurer la contrainte sur les rangs :

$$\forall (i, j) \in [1; N]^2, i \neq j \Rightarrow \theta_i \neq \theta_j$$

7.4.2 Population et génération initiale

Dans l'implémentation choisie, la population est composée d'un nombre fixe T_p d'individus.

La génération de la *population initiale* consiste à choisir aléatoirement le chemin et le rang de chaque avion pour les T_p individus :

1. Le chemin de chaque avion peut être choisi aléatoirement dans le domaine de la variable $([1, n_{ch}^i],$ pour chaque avion i). Les valeurs les plus faibles (correspondant aux chemins les plus courts) sont cependant privilégiées.
2. De la même manière que dans la résolution par une méthode déterministe, le rang des avions établit le lien entre les séquences à réaliser sur les pistes et la phase de roulage. Le rang θ_i de l'avion i est donc déduit de l'heure d'accès à la piste prévue dans la séquence le concernant. Afin de conserver une part d'aléatoire et d'explorer ainsi un maximum de solutions sans converger immédiatement vers un optimum local, un bruit gaussien d'amplitude maximale H_p est appliqué aux heures prévues d'accès à la piste de chaque avion. Les rangs des avions sont alors déduits de ces heures bruitées.

7.4.3 Fonction d'évaluation

L'évaluation de chaque individu consiste à quantifier le critère défini en 6.5.4 en fonction des chemins et du rang de chaque avion de l'individu. Pour cela, il faut calculer les attentes à imposer aux avions pour obtenir leurs trajectoires résultantes.

Lors de l'évaluation d'un individu, la méthode déterministe décrite au 7.2 est alors appliquée à chaque avion, afin de déterminer sa trajectoire. Seul l'algorithme III.1 (choix des attentes) est donc utilisé, sur le chemin imposé par le génotype de l'individu : cet algorithme est appliqué successivement aux différents avions, dans l'ordre décrit par leurs rangs. Lorsque la méthode déterministe ne trouve pas de solution avec le chemin et le classement donnés, les avions en conflit sont identifiés.

Le caractère *partiellement séparable* du problème permet de mettre en place une *fitness partielle* qui correspond à une note pour chaque avion a de chaque individu : cette note est définie par la pénalité décrite en 6.5.4, si une trajectoire sans conflit a été trouvée pour l'avion. Dans le cas contraire (a est en conflit avec un avion b), la fitness partielle des deux avions est augmentée d'une constante K , caractérisant le fait qu'ils sont en conflit (cette constante est donc choisie de telle sorte qu'elle soit toujours supérieure aux valeurs possibles de la pénalité d'un avion) :

$$f_p^a = p_a + n_c^a \times K$$

avec $\left| \begin{array}{l} p_a \text{ la pénalité de l'avion } a \\ n_c^a \text{ le nombre de conflits non résolus dans lesquels } a \text{ est impliqué} \end{array} \right.$

La *fitness globale* f_g^i d'un individu s'exprime alors à partir des fitness partielles f_p^a de chaque avion a :

7.4. ALGORITHME GÉNÉTIQUE POUR LA RÉOLUTION DES CONFLITS AU ROULAGE 87

- Dans le cas où une solution a été trouvée pour tous les avions :

$$f_g^i = \frac{1}{2} + \frac{1}{2 + \sum_{a=1}^N f_p^a}$$

- Sinon, le nombre $n_c > 0$ de conflits non résolus est comptabilisé :

$$f_g^i = \frac{1}{2} \frac{1}{1 + n_c}$$

Les solutions non-admissibles (avec au moins un conflit non résolu) sont donc associées à une fitness dans l'intervalle $]0; \frac{1}{2}[$; tandis que les solutions admissibles sont associées à des fitness dans l'intervalle $]\frac{1}{2}; 1]$.

7.4.4 Reproduction des individus

Entre chaque génération, la reproduction des individus s'effectue en suivant les étapes de sélection puis de mutations et de croisements :

- **La méthode de sélection** retenue est celle de la roulette, car la population est composée d'individus représentant des solutions qui ne sont pas nécessairement admissibles : ce modèle de sélection permet en effet de favoriser les individus qui représentent des solutions avec peu de conflits (voire aucun) tout en conservant une probabilité de sélectionner des individus représentant une solution non admissible, afin d'assurer la diversification de la population.
- **La mutation** d'un individu modifie aléatoirement la valeur du chemin ou du rang d'un avion dans le génotype. Le choix de l'avion dont les variables seront modifiées est guidé par la fitness partielle associée à chaque avion, selon le principe de la roulette (chaque avion est représenté par un segment dont la longueur correspond au rapport entre sa fitness partielle et la somme des fitness partielles). Lorsque la mutation modifie le rang d'un avion, les rangs des autres avions sont également adaptés, pour que l'ensemble des rangs décrits dans l'individu résultant représente toujours une permutation conforme.
- **Le croisement** de deux individus consiste à en générer deux nouveaux, dont les caractéristiques sont un mélange entre les deux parents. Cette opération s'effectue sur un avion choisi de telle sorte que sa fitness partielle dans l'un des parents est bien meilleure que dans l'autre. Le parent dont l'avion sélectionné possède la plus haute fitness partielle est utilisé comme base pour le nouvel individu généré : le chemin et/ou le rang de cet avion (dans le nouvel individu) sont remplacés par les valeurs du même avion dans l'autre parent.

Soient les individus $I_1 = \{a_1^1, a_2^1, \dots, a_N^1\}$ et $I_2 = \{a_1^2, a_2^2, \dots, a_N^2\}$. Si l'avion a_i est choisi pour le croisement, et que a_i possède une meilleure fitness partielle dans I_2 , on pourra obtenir un nouvel avion b parmi les possibilités suivantes :

$$\begin{cases} b = (\eta_{a_i^2}, \theta_{a_i^2}) \\ b = (\eta_{a_i^1}, \theta_{a_i^2}) \\ b = (\eta_{a_i^2}, \theta_{a_i^1}) \end{cases}$$

L'individu généré par le croisement de I_1 et I_2 possède alors les mêmes chemins et rangs que I_1 à l'exception de l'avion a_i qui est remplacé par l'avion b :

$$I_{1 \times 2} = \{a_1^1, a_2^1, \dots, b, \dots, a_N^1\}$$

Lorsque le croisement entre deux individus est effectué, il est opéré dans les deux sens : on obtiendra alors deux individus $I_{1 \times 2}$ et $I_{2 \times 1}$. Un avion jugé moins bon dans chaque individu sera donc remplacé par celui de l'autre parent dans l'individu généré.

La condition d'arrêt de l'algorithme est définie en fonction de deux facteurs :

- lorsqu'un nombre maximal n_{gen} de générations est atteint ;
- lorsque la meilleure fitness de la population n'évolue plus sur un nombre fixé de générations.

7.4.5 Améliorations classiques

Les trois types d'améliorations classiques sont étudiées pour l'algorithme génétique :

- **Le scaling linéaire** s'avère très efficace lorsqu'il est couplé avec une sélection de type *roulette*. Le scaling utilisé est donc une adaptation de ce dernier et représente un *scaling linéaire adaptatif* : l'impact du scaling dépend de la moyenne et de l'écart type des fitness des individus de la génération courante :

$$f_{gs}^i = \max(0, f_g^i - (\bar{f}_g - c \times \sigma))$$

avec

	c une constante permettant de mettre l'écart type à l'échelle voulu.
	\bar{f}_g la moyenne des fitness globales
	σ l'écart type des fitness globales
	f_g^i la fitness globale de l'individu i
	f_{gs}^i la fitness globale de l'individu i après scaling

Dans cette définition, chaque fitness est diminuée de la moyenne et de l'écart type des fitness. L'ensemble des f_{gs}^i pour chaque avion i est ensuite à nouveau normalisé entre 0 et 1.

- **Le sharing clusterisé adaptatif** est utilisé, comme décrit dans la partie 7.3.5 :
 - La distance entre deux individus est définie par le taux d'avions dont les variables diffèrent : Soient deux individus $A = \{(\eta_1^A, \theta_1^A), \dots, (\eta_N^A, \theta_N^A)\}$ et $B = \{(\eta_1^B, \theta_1^B), \dots, (\eta_N^B, \theta_N^B)\}$. Un avion i est différent dans les deux individus lorsque :

$$\eta_i^A \neq \eta_i^B \text{ ou } \theta_i^A \neq \theta_i^B$$

La distance entre A et B s'exprime alors par :

$$d(A, B) = \frac{|\{i \in [1; N] \mid \eta_i^A \neq \eta_i^B \text{ ou } \theta_i^A \neq \theta_i^B\}|}{N}$$

- Le barycentre de deux individus A et B pondérés par les coefficients α_A et α_B correspond à un individu C dont les chemins correspondent à ceux de l'individu associé au plus haut coefficient et dont les rangs correspondent aux rangs des barycentres des rangs de A et B :

En supposant $\alpha_A \leq \alpha_B$,

$$\forall i \in [1; N], \begin{cases} \eta_i^C = \eta_i^B \\ \theta_i^C = |\{j \in [1; N] \mid \pi_j < \pi_i \text{ ou } (\pi_j = \pi_i \text{ et } j \leq i)\}| \end{cases}$$

$$\text{avec : } \forall i \in [1; N], \pi_i = \frac{\alpha_A \theta_i^A + \alpha_B \theta_i^B}{\alpha_A + \alpha_B}$$

- Différents valeurs de taux de sharing seront envisagées (de 0 à 0.9) au prochain chapitre, pour des seuils $S1 = 0.75$ et $S2 = 0.85$ fixés.
- **Aucune méthode d'optimisation locale** n'est nécessaire en fin de convergence, car dans le codage proposé, chaque individu représente déjà à un optimum local : la méthode déterministe est en effet utilisée pour rechercher systématiquement les meilleures trajectoires, en fonction de l'affectation de chemins et du classement des avions décrits par l'individu.

7.5 Conclusion

La mise au point de deux méthodes de résolutions très différentes a pour but de comparer leur efficacité sur le problème de résolution de conflits au roulage en intégrant des séquences d'avions à réaliser sur les pistes.

La méthode déterministe résout en réalité une simplification du problème en réduisant l'espace de recherche et en se basant sur un classement précalculé des avions en conflit. Un algorithme de type branch & bound parcourt alors l'arbre des possibilités pour chaque chemin de chaque avion. Le problème élémentaire à résoudre consiste donc à rechercher un plus court chemin (en évitant certains obstacles) dans un graphe. Le classement nécessaire à cette méthode peut être directement déduit des séquences optimales d'avions sur les pistes. En cas d'échec, le classement est adapté en fonction des conflits non résolus.

L'algorithme génétique possède l'avantage d'être flexible et évolutif car il ne fait aucune hypothèse sur le critère ni sur les contraintes. Ce type d'algorithme s'est déjà montré efficace sur le problème de résolution de conflits en l'air [Granger 01] et au roulage [Gotteland 04] mais nécessite un étalonnage des paramètres, parfois complexe à réaliser (comme cela est présenté au prochain chapitre), l'objectif étant de trouver de meilleures solutions en explorant un espace de recherche plus vaste. La version de l'algorithme génétique proposée dans cette thèse est en réalité une hybridation entre un algorithme génétique et la méthode déterministe qui est intégrée dans la phase d'évaluation des individus.

Chapitre 8

Résultats

8.1 Introduction

Les deux méthodes (déterministe et stochastique) présentées dans le chapitre précédent ont été définies de façon à pouvoir ou non intégrer les séquences optimales d'avions à réaliser sur les pistes. L'objectif de ce chapitre est donc de comparer ces méthodes.

Pour ce faire, il est indispensable d'étalonner l'algorithme génétique en trouvant les valeurs les plus efficaces pour les nombreux paramètres influençant la vitesse et la qualité de la convergence : cet étalonnage concerne entre autres les taux de croisement et de mutation, la taille de la population, le nombre maximal de générations et le taux de sharing. Il nécessite l'application de l'algorithme génétique sur une journée entière de trafic avec chaque combinaison des paramètres. L'évaluation de chaque configuration des paramètres est relative au délai moyen résultant et au nombre de créneaux imposés par le CFMU non respectés. Cette approche (plutôt qu'un étalonnage sur des problèmes fixes en comparant uniquement les fitness résultantes) permet de sélectionner la configuration de l'algorithme la plus efficace par rapport aux objectifs des simulations et non en terme de vitesse de convergence ou de meilleures fitness sur chaque situation.

L'efficacité de la résolution des conflits au roulage en intégrant les séquences d'avions à réaliser sur les pistes est ensuite mesurée. Chaque méthode de résolution est alors comparée dans ses deux configurations possibles (avec et sans prise en compte des séquences). Le temps de calcul de chaque méthode est tout d'abord mesuré à titre indicatif, sans que cela ne soit déterminant pour le choix de la meilleure méthode. Par la suite, la qualité globale des solutions est mesurée : cela inclut le délai moyen généré sur la journée complète ainsi que le respect des créneaux imposés par le CFMU. Enfin, le respect des séquences d'avions sur les pistes pour chaque méthode (uniquement avec prise en compte des séquences à réaliser) est évalué, afin de tester si la réalisation des séquences optimales d'avions sur les pistes est satisfaisante.

8.2 Étalonnage de l'algorithme génétique

8.2.1 Présentation

Afin de déterminer la valeur la plus appropriée de chaque paramètre de l'algorithme génétique, un classement des paramètres est effectué, en fonction de leur influence habituelle sur la qualité de la convergence : dans un premier temps, les paramètres calibrant le renouvellement de la population (taux de croisement et de mutation) sont étudiés, en prenant une taille de population et un nombre de

générations maximal fixés et volontairement élevés, afin de ne pas limiter la convergence de l'algorithme génétique. La somme du taux de croisement et de mutation donne le taux de renouvellement : il représente le pourcentage d'individus utilisés comme parents pour la nouvelle génération. Les effets de ces taux ne sont pas indépendants et doivent donc être étudiés en parallèle.

Une fois ces taux fixés, la rapidité de convergence de l'algorithme peut être étudiée en fonction des trois paramètres suivants :

- **La taille de la population** correspond au nombre d'individus composant chaque génération. Le caractère hybride de l'algorithme génétique défini (qui utilise une méthode d'optimisation déterministe pour calculer précisément les trajectoires décrites par chaque individu), associé au caractère partiellement séparable du problème de résolution de conflits (qui donne lieu à des opérateurs de croisement et de mutation accélérant la convergence), doit normalement permettre d'utiliser une taille de population relativement faible (inférieure à 100, si on se réfère au même type d'algorithme génétique, défini dans [Gotteland 04]).
- **Le nombre maximal de générations** correspond au nombre d'itérations du cycle de l'évolution de l'algorithme. Pour les mêmes raisons que précédemment, un faible nombre de générations (inférieur à 200) devrait être suffisant.
- **Le taux de sharing** a une influence sur la modification de la fitness de chaque individu et sur la répartition des individus conservés au cours de la reproduction.

L'étalonnage s'effectue alors en exécutant l'algorithme plusieurs fois sur une même journée de trafic, en faisant varier un à un les paramètres : avec une période de résolution de $\Delta_r = 2$ minutes, chaque simulation nécessite la résolution des conflits de 720 situations de trafic, comprenant en général plusieurs clusters différents.

La journée de trafic utilisée est celle du 22 mars 2002, présentée au 3.4.4 :

- 1454 vols, dont 735 départs et 719 arrivées ;
- 278 créneaux de décollage imposés par le CFMU, dont 14 non réalisables (à cause de l'heure tardive de libération du parking et du temps de roulage minimal).

8.2.2 Taux de croisement et de mutation

L'algorithme génétique est utilisé avec différents taux de croisement et de mutation sur une journée complète, en fixant les autres paramètres :

- La taille de la population est fixée à 100 individus, ce qui s'avère en pratique un bon compromis entre la diversité de la population et le temps de calcul.
- Le nombre de générations est fixé à 200, ce qui s'avère en pratique suffisant pour la convergence.
- Aucune méthode de sharing n'est initialement utilisée, afin de ne pas perturber la convergence.

Une limite sur le taux de renouvellement envisagé (somme du taux de croisement et du taux de mutation) est considérée a priori, car les valeurs extrêmes de ce taux (inférieures à 40% ou supérieures à 95%) ne correspondent pas à un bon fonctionnement de l'algorithme génétique.

Le tableau de la figure 8.1 présente les délais des départs non régulés dans les différentes configurations. Le respect des créneaux imposés par le CFMU étant une priorité, les résultats des simulations dans lesquelles des créneaux réalisables ont été ratés sont volontairement occultés. Il ne reste alors que les combinaisons de taux de croisement et mutation satisfaisantes : les délais moyens obtenus sont globalement similaires dans les différentes configurations sélectionnées, avec une légère augmentation lorsque le taux de mutation est faible : la valeur minimale du délai moyen (36 secondes par avion) est obtenue avec un taux de mutation de 50% et un taux de croisement de 40%.

Crois. \ Mut.	20	25	30	35	40	45	50	55	60	65	70
20										38.0	37.2
25									38.7		
30								37.4	37.3		
35						37.3			38.2		
40						36.6	37.1				
45				37.2	36.1	37.1					
50			36.2	36.6	36.0	36.7					
55		36.3	36.4	36.2	37.2						
60	36.2	36.9	36.5								
65		36.3	36.2								
70											

FIG. 8.1 – Étalonnage des taux de croisement et de mutation

Ces valeurs définissent donc un taux optimal de renouvellement de la population de 90%, ce qui peut apparaître relativement élevé (comparé aux taux habituellement retenus dans ce type d'algorithmes), mais elles s'expliquent par le caractère hybride de l'algorithme génétique, qui n'explore que des optima du problème posé et qui nécessite donc une population la plus diversifiée possible.

8.2.3 Taille de population et nombre de générations

L'étalonnage de la taille de la population est effectué en même temps que celui du nombre maximal de générations, car ces deux paramètres définissent ensemble l'association entre la qualité et la vitesse de convergence : la qualité de convergence nécessite une importante diversité des individus dans la population, donc une taille de population suffisante, tandis que la rapidité de convergence (en temps de calcul) s'exprime en fonction du nombre total d'individus visités, ce qui correspond au produit de la taille de la population et du nombre de générations.

La figure 8.2 présente le délai moyen par avion (sur la journée) ainsi que le nombre de créneaux réalisables ratés (lorsqu'il est non nul) en fonction du nombre maximal de générations, pour différentes tailles de la population.

Pour un nombre maximal de générations supérieur ou égal à 100, tous les créneaux réalisables sont respectés. À partir d'un nombre maximal de 200 générations, les délais ne varient plus, ce qui signifie que 200 générations sont suffisantes pour la convergence de l'algorithme, et ce pour toutes les tailles de population testées.

Pour un nombre maximal de générations supérieur ou égal à 150, des tailles de population de 100 ou 200 individus aboutissent à des résultats très similaires : le délai moyen est inférieur à 36.1 secondes par avion sur la journée. La convergence de l'algorithme génétique est ainsi satisfaisante à partir d'une population de 100 individus et un nombre maximal de 150 générations : ces valeurs sont donc retenues pour la suite.

8.2.4 Sharing

La figure 8.3 montre le nombre de créneaux ratés et le délai moyen par avion (sur la journée) en fonction du taux de sharing utilisé (un taux nul correspondant à l'absence de sharing).

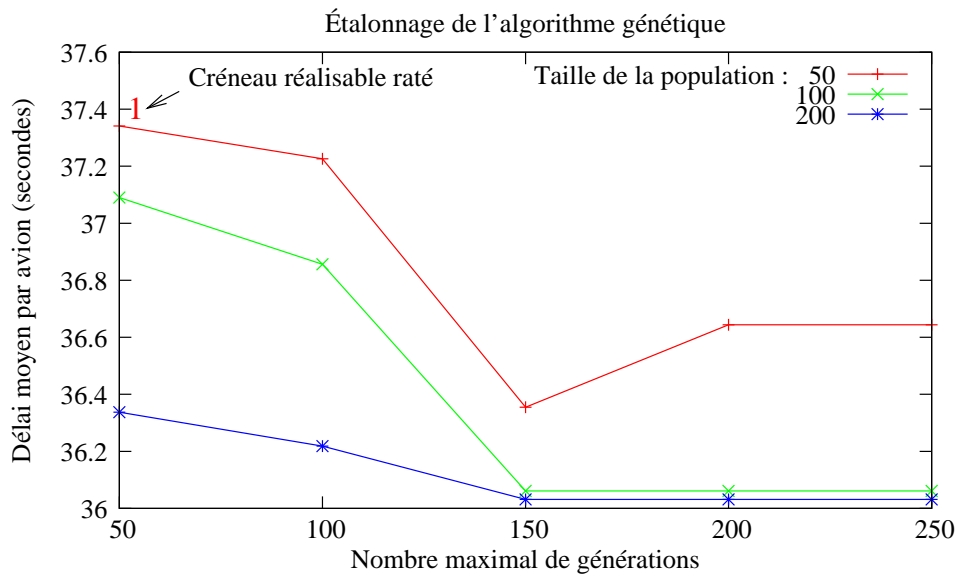


FIG. 8.2 – Étalonnage de la taille de la population et du nombre maximal de générations

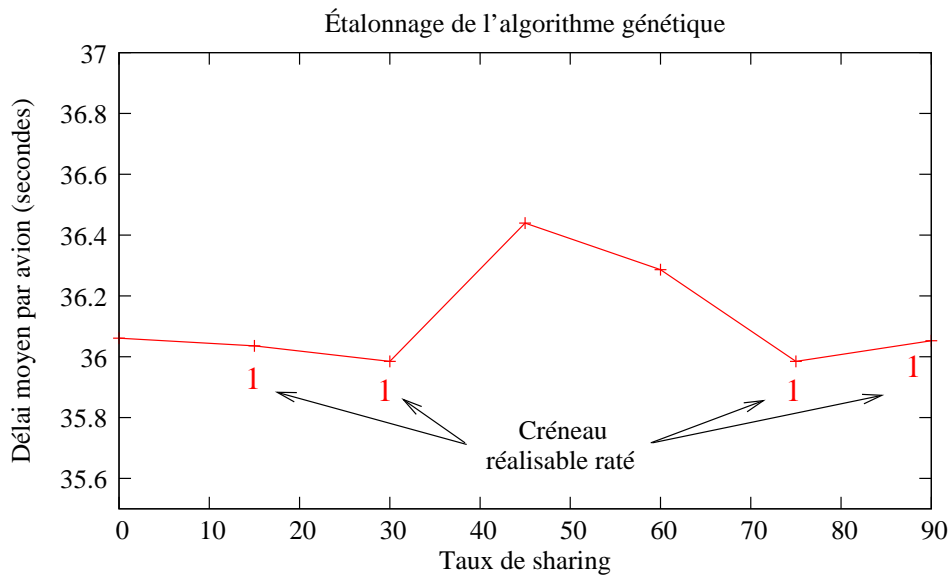


FIG. 8.3 – Étalonnage du taux de sharing (%)

Il apparaît que le sharing utilisé n'améliore pas la qualité des solutions trouvées par l'algorithme génétique : le délai moyen n'est inférieur à la valeur obtenue sans sharing que lorsqu'un créneau réalisable est raté. Ce résultat paradoxal peut provenir de l'horizon de prédiction limité associé au critère mettant en concurrence la minimisation des délais et le respect des créneaux. Malgré le fort coefficient affecté au non respect d'un créneau, il se peut que certaines situations incitent à repousser un décollage soumis à créneau pour réduire le délai des autres avions (sans que cela ne remette en cause à cet instant le respect du créneau) mais engendrent ensuite des situations où le créneau retardé ne peut plus être respecté. Cela signifie que des solutions peuvent sembler meilleures à court terme (moins de délai et aucun créneau raté) alors qu'elles génèrent des situations futures globalement plus pénalisantes pour les avions, comme cela a souvent été constaté dans le modèle de simulation avec horizon de prédiction limité. Ces résultats montrent qu'il serait bénéfique d'affiner le critère en fonction de la situation des avions en fin d'horizon de prédiction, pour détecter en avance les risques de non respect des créneaux imposés par le CFMU.

L'utilisation du sharing tel qu'il est défini ne semble en tous cas pas bénéfique, et n'est donc pas retenue.

8.3 Comparaison des deux méthodes

8.3.1 Temps de calcul

Les deux méthodes de résolution sont comparées sur la même journée de trafic à l'aéroport de Roissy Charles de Gaulle, avec les paramètres de simulation suivants :

- Le taux d'incertitude sur les vitesses est fixée à 20%.
- L'horizon de prédiction pour les séquences d'avions sur les pistes est de 30 minutes.
- L'horizon de prédiction pour la résolution des conflits au roulage est de 7 minutes
- Le pas de temps de la résolution est de 2 minutes.
- La perte de niveau de priorité des arrivées posées est fixée à $\cdot \delta_{prio} = 30$ minutes.

L'algorithme génétique est utilisé avec les paramètres définis précédemment :

- Taille de la population : 100
- Nombre de générations maximal : 150
- Taux de croisement : 40%
- Taux de mutation : 50%
- Sharing : sans

La figure 8.4 montre que le temps de calcul de la méthode déterministe est bien inférieur à celui de l'algorithme génétique. Cependant le temps d'exécution de l'algorithme génétique reste compatible avec le pas de rafraîchissement souhaité du système (une ou deux minutes) dans la plupart des cas rencontrés (dans le cadre d'une utilisation en temps réel, le temps de calcul d'une solution doit simplement être inférieur au pas de rafraîchissement). De plus, ce temps de calcul pourrait être grandement amélioré en parallélisant l'algorithme génétique (évaluation en parallèle des individus), ce qui n'était pas du tout l'objet de ces simulations.

Les temps de calcul extrêmement faibles obtenus avec la méthode déterministe (bien inférieurs à la seconde) peuvent s'avérer en revanche un atout considérable : les solutions aux conflits sont calculées presque immédiatement, ce qui laisse envisager des interactions avec les opérationnels (essais de différentes possibilités, révision manuelle du classement, etc.) ou même avec d'autres systèmes d'aide utilisés sur l'aéroport.

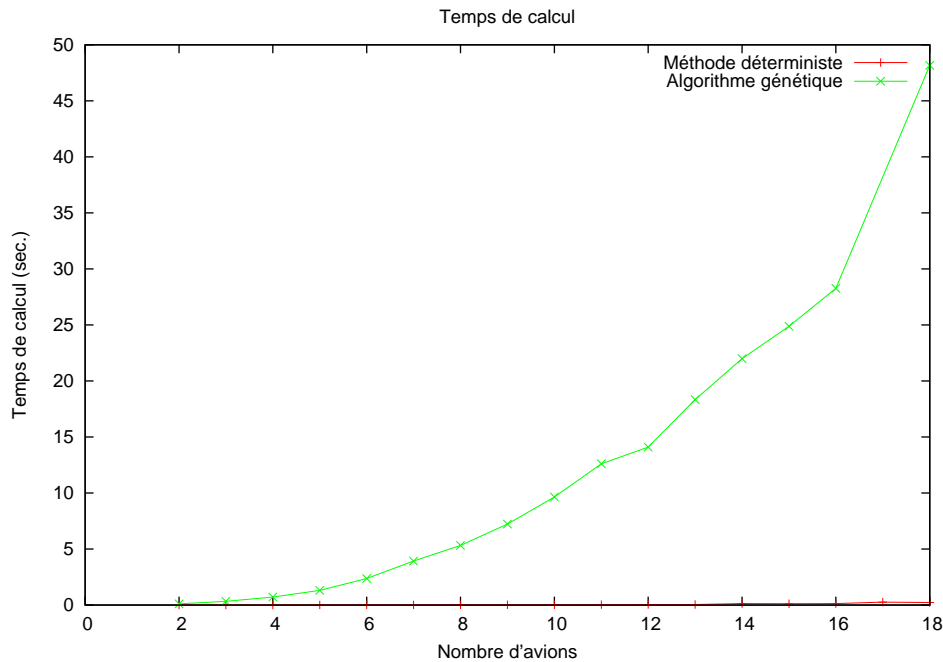


FIG. 8.4 – Comparaison des temps de calcul des deux méthodes

8.3.2 Comparaison des délais et du respect des créneaux

Les figures 8.5 et 8.6 présentent les résultats obtenus par simulation de la journée complète, pour chaque méthode de résolution de conflits, avec et sans prise en compte des séquences optimales.

Pour les deux méthodes, le délai moyen des départs (non régulés) est réduit de moitié lorsque les séquences optimales sont prises en compte. Les arrivées sont en revanche moins affectées par ce facteur : leur phase de roulage ne dépend pas des séquences réalisées sur les pistes. Ces résultats montrent néanmoins que l'intégration des séquences dans la résolution des conflits au sol permet de réduire fortement les délais générés.

Avec l'algorithme génétique, l'objectif de viser les séquences a un impact sur le respect des créneaux imposés par le CFMU (trois créneaux réalisables ratés sans viser les séquences contre aucun en les visant) : lorsqu'il ne dispose pas de séquences optimales d'avions sur les pistes, l'algorithme génétique effectue une minimisation des délais de l'ensemble des avions, sans réellement distinguer les départs régulés, et cela peut aller à l'encontre du respect des créneaux dans certaines situations.

La différence principale entre les résultats des deux méthodes de résolution concerne le délai moyen des arrivées : avec l'algorithme génétique, ces délais sont presque deux fois moindres que ceux obtenus avec la méthode déterministe, même lors d'une exploitation sans viser les séquences d'avions optimales sur les pistes (cf. figure 8.6). Le délai maximal (ainsi que l'écart type) est également fortement réduit avec l'algorithme génétique. Cela montre que les arrivées sont moins bien gérées avec la méthode déterministe et ceci provient certainement de la simplification opérée sur le problème : le classement des avions qui rend les arrivées systématiquement moins prioritaires au roulage leur inflige finalement beaucoup plus de retards.

La figure 8.7 montre le délai moyen en fonction du nombre d'avions en mouvement sur une période de 10 minutes. Là encore, les solutions apportées par l'algorithme génétique sont bien meilleures pour les arrivées : en période de pointe, les délais moyens par arrivée représentent le tiers de ceux

			Délai moyen (secondes/avion)		Délai maximal (secondes)	Écart type (secondes)	Créneaux ratés
Avec séquences	Algorithme génétique	Départs	57	36.0	695	91.9	14
		Arrivées	22		320	34.3	
	Méthode déterministe	Départs	59	54.0	920	104.5	15
		Arrivées	52		770	95.4	
Sans séquence	Algorithme génétique	Départs	97	58.3	1065	153.5	17
		Arrivées	28		325	44.9	
	Méthode déterministe	Départs	112	73.7	1380	180.5	15
		Arrivées	48		600	78.7	

FIG. 8.5 – Temps de calcul et pénalité sur la journée complète pour chaque méthode

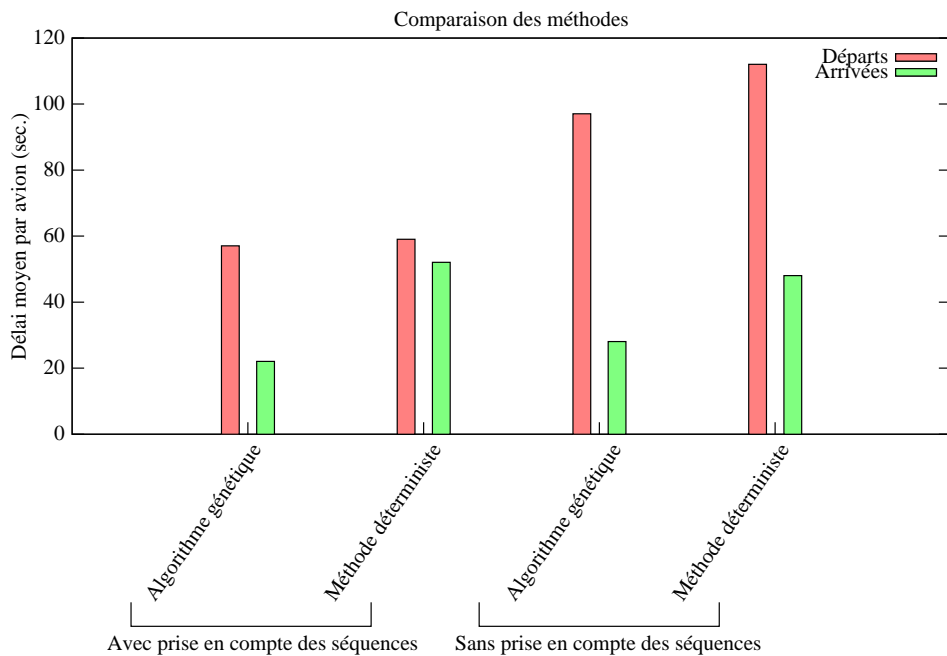


FIG. 8.6 – Comparaison des délais moyens des deux méthodes

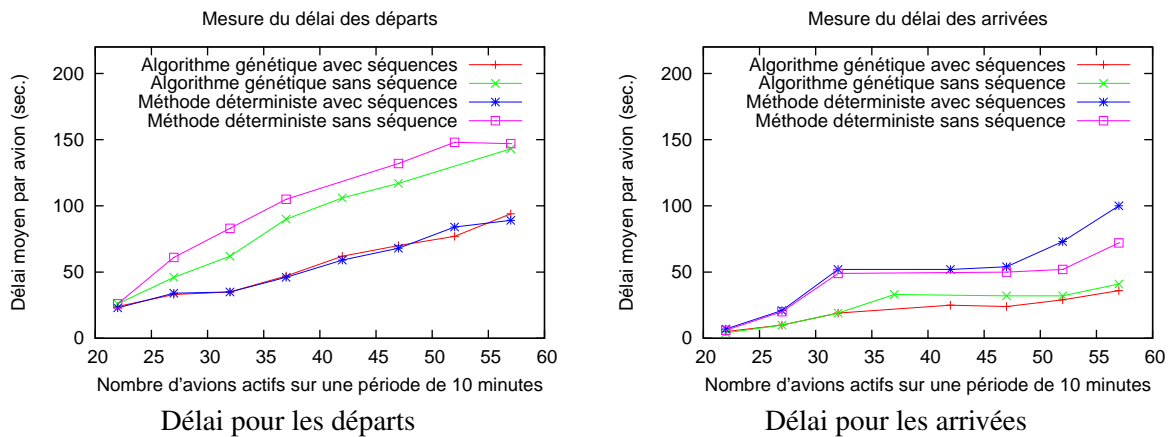


FIG. 8.7 – Délais en fonction du nombre d'avions considérés

obtenus avec la méthode déterministe. Pour les deux méthodes, la prise en compte des séquences optimales d'avions sur les pistes permet de réduire de plus de la moitié le délai moyen des départs dans les périodes chargées : une fois encore, les résultats montrent que l'intégration des séquences dans la gestion de la phase du roulage est bénéfique.

La figure 8.8 illustre la façon dont sont programmés les décollages soumis à un créneau imposé par le CFMU pour chaque méthode. Les deux méthodes engendrent une programmation équivalente, avec une majorité de décollages programmés dans l'intervalle $[-5min.; +5min.]$ par rapport à l'heure imposée. Lorsque le critère prend en compte les séquences optimales calculées, seuls 14 créneaux sont ratés, ce qui correspond au minimum atteignable (ces 14 créneaux ne peuvent pas être réalisés à cause de l'heure tardive de libération du parking et du temps de roulage minimum des vols concernés).

Néanmoins, la prise en compte des séquences d'avions sur les pistes n'influence pas fortement la distribution des heures de décollage soumis à un créneau imposé par le CFMU : la fenêtre de tolérance définie par le CFMU (entre -5 et $+10$ minutes autour de l'heure de décollage imposée) est suffisamment large pour que les algorithmes de résolution de conflits parviennent à respecter correctement ces créneaux, même sans anticipation des séquences d'avions sur les pistes (mais ce résultat pourrait ne plus être valable si le nombre de créneaux imposés venait à augmenter).

8.3.3 Pertinence des séquences d'avions prévues sur les pistes

L'objectif de calculer en avance des séquences optimales d'avions sur les pistes (à réaliser lors de la gestion des conflits au roulage) n'est pas seulement de réduire des délais des avions : il s'agit également de rendre le système de gestion du trafic aérien plus prévisible, notamment en ce qui concerne les heures de décollage (systèmes DMAN). Les deux méthodes de résolution de conflits intégrant les séquences optimales sur les pistes doivent donc également être comparées en terme de réalisation des séquences calculées : la simulation de la phase de roulage est effectuée avec incertitudes sur les vitesses de roulage et résolution des conflits entre avions, ce qui vient donc nécessairement perturber la réalisation des séquences optimales.

Les heures d'atterrissage étant très peu modifiables, et les heures de décollage des départs régulés étant évaluées par rapport aux créneaux imposés par le CFMU, la comparaison ne concerne que les décollages non régulés et s'effectue en calculant la différence entre l'heure de décollage prévue par la séquence (un certain temps en avance) et l'heure effective de décollage dans la simulation. Le caractère

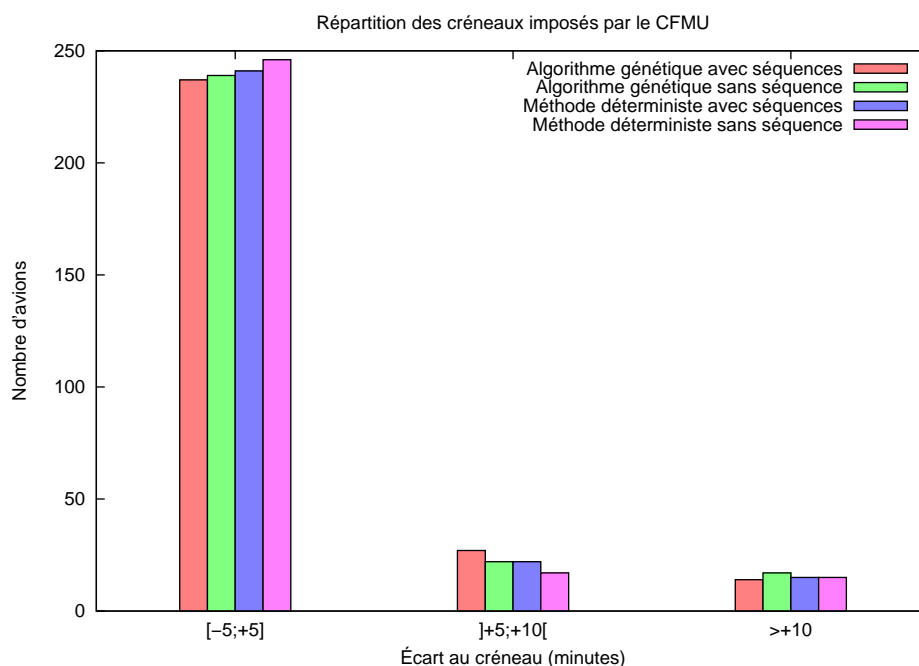


FIG. 8.8 – Répartition des décollages soumis à un créneau imposé par le CFMU

dynamique des séquences d'avions (recalculées à chaque situation) rend la mesure du respect de ces séquences délicate : à quel instant les heures de décollage prévues dans les séquences doivent-elles être comparées à l'heure effective de décollage ?

Les exigences opérationnelles sont actuellement d'obtenir une heure fiable de décollage (donc une séquence stable) 30 minutes avant le décollage effectif des avions.

L'heure effective de décollage de chaque départ non régulé est donc comparée avec l'heure prévue par les séquences optimales d'avions sur les pistes 30 minutes avant. La distribution de la différence entre ces heures est présentée sur la figure 8.9 : les résultats sont similaires avec les deux méthodes. Une grande majorité des départs (plus de 360 sur un total de 457 départs non-régulés) respectent à une minute près l'heure de décollage prévue par la séquence 30 minutes en avance. De plus, les heures de décollage prévues sont fiables à 5 minutes près dans plus de 98% des cas (toujours avec les deux méthodes de résolution) : la pertinence des séquences calculées 30 minutes en avance apparaît donc tout à fait satisfaisante.

8.4 Conclusion

L'intégration des séquences optimales d'avions sur les pistes dans la gestion de la phase de roulage des avions améliore nettement les résultats obtenus par les deux méthodes de résolution de conflits étudiées, que ce soit en terme de délai moyen par avion, de respect des créneaux de décollage imposés par le CFMU ou encore de pertinence des heures de décollage prévues 30 minutes en avance.

Cette intégration se révèle un véritable atout pour la méthode déterministe, dont les résultats deviennent compétitifs avec ceux de la méthode stochastique : l'approche par classement et le temps de calcul presque instantané de cette méthode laissent entrevoir de nombreuses possibilités d'interactions avec les contrôleurs ou avec d'autres systèmes d'aide à la décision utilisés sur l'aéroport. La principale

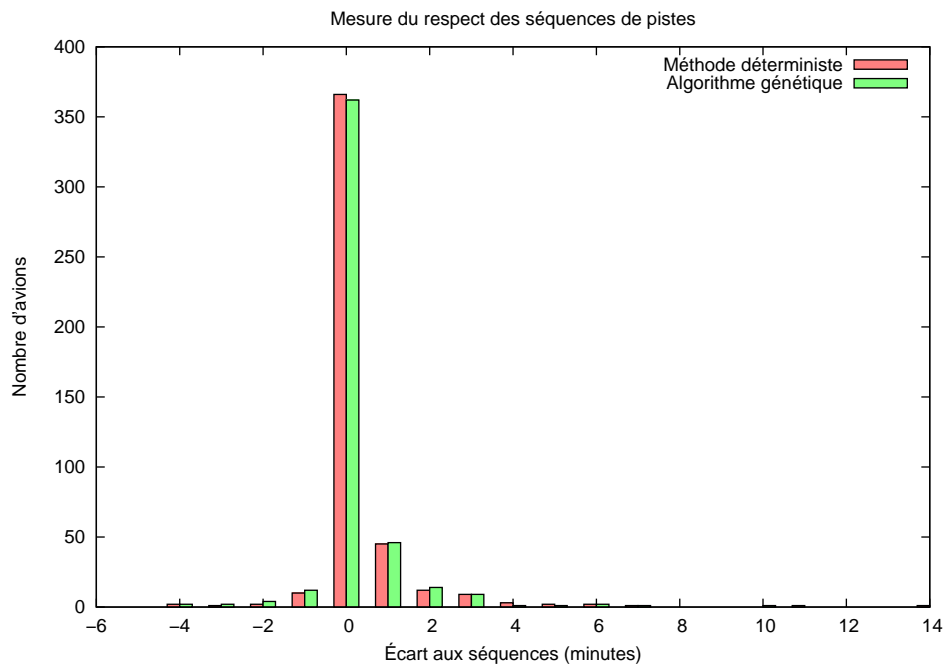


FIG. 8.9 – Mesure du respect des séquences d’avions sur les pistes

différence réside dans le délai moyen des arrivées qui reste élevé avec la méthode déterministe, car ces dernières sont définitivement considérées comme moins prioritaires une fois posées.

La méthode de résolution par algorithme génétique obtient les meilleurs résultats, et reste plus évolutive (par rapport aux critères et aux contraintes envisageables dans la gestion du trafic au roulage). Les temps de calcul mesurés, qui pourraient certainement être nettement améliorés en parallélisant l’algorithme (ce qui n’est pas l’objet de cette thèse), sont élevés mais restent néanmoins compatibles avec un système dont le temps de réponse souhaité serait de l’ordre de la minute.

Quatrième partie

Applications

Chapitre 9

Influence des principaux paramètres

9.1 Introduction

Ce chapitre a pour objectif de mesurer l'impact de différents paramètres opérationnels sur la qualité des solutions trouvées lors de la résolution des conflits au roulage. Parmi les nombreux paramètres utilisés dans la modélisation du trafic au sol, certains correspondent en effet directement à des concepts ou à des pratiques opérationnelles souvent à l'étude :

- **La taille de l'horizon de prédiction** utilisé pour l'optimisation des séquences d'avions sur les pistes peut avoir un impact plus ou moins fort sur les délais résultants pour le trafic. En effet, une anticipation à plus ou moins long terme des séquences d'avions sur les pistes peut jouer un rôle direct sur leur stabilité et donc sur la cohérence des résolutions de conflits successives. L'étude de la variation des délais en fonction de cet horizon de prédiction doit permettre de mesurer l'importance de l'anticipation des séquences d'avions sur les pistes : à partir d'une certaine taille de l'horizon de prédiction, les résultats en terme de délai moyen devraient atteindre une limite inférieure, et signifier qu'il n'est pas utile d'anticiper davantage les séquences d'avions, du point de vue du contrôle d'aérodrome.
- **Le taux d'incertitude sur les vitesses de roulage** influence de façon évidente la complexité des situations de trafic à résoudre et peut donc fortement dégrader la qualité des solutions trouvées : lorsque le taux d'incertitude est grand, les zones d'incertitudes des avions sont plus grandes et le nombre de conflits à résoudre peut devenir conséquent, ce qui se traduit logiquement par une augmentation du délai moyen des avions. L'anticipation des séquences optimales d'avions sur les pistes doit cependant permettre une meilleure organisation du trafic, pour éventuellement réduire cette augmentation du délai avec les incertitudes.
- **Le pas de rafraîchissement** du système, qui correspond à la période de résolution des conflits, peut également être déterminant sur la qualité des solutions obtenues : lorsque ce pas est élevé, le système perd en flexibilité et ne peut plus affiner certaines décisions prises au cours de la résolution précédente. Ce paramètre doit naturellement être étudié conjointement au taux d'incertitude sur les vitesses : plus l'incertitude est élevée, plus le système se doit d'être flexible.
- **La priorité des départs sur les arrivées** au roulage, souvent décrite comme nécessaire par les opérationnels, doit également être étudiée : la réduction du niveau de priorité des arrivées au roulage, mise en place dans la méthode déterministe augmente de façon significative leur délai moyen, ce qui en fait le point faible de la méthode. Un paramétrage moins pénalisant peut éventuellement permettre d'améliorer les résultats obtenus.

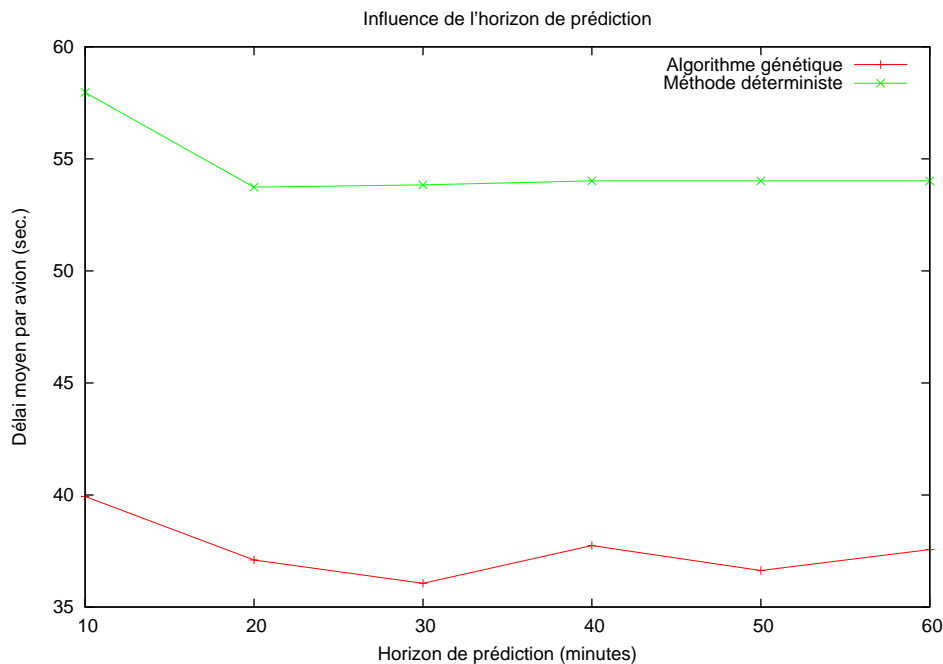


FIG. 9.1 – Influence de l'horizon de prédiction sur les délais

9.2 Horizon de prédiction des séquences d'avions sur les pistes

9.2.1 Présentation

L'horizon de prédiction des séquences correspond à une anticipation des opérations sur les pistes à plus ou moins long terme. Ce niveau d'anticipation se répercute tout au long de la résolution des conflits au roulage, lorsque la réalisation des séquences fait partie des objectifs de cette dernière. L'objectif est alors de déterminer la valeur minimale de l'horizon de prédiction à partir de laquelle la résolution des conflits au roulage engendre des délais minimaux, tout en respectant les créneaux de décollage imposés par le CFMU.

Dans ce cadre, les simulations sont effectuées sur une même journée de trafic, en faisant varier l'horizon de prédiction des séquences de 10 à 60 minutes. Afin de pouvoir comparer les résultats, les autres paramètres du simulateur sont fixés :

- Le taux d'incertitude sur les vitesses de roulage est de $\delta_v = 20\%$.
- Le pas de rafraîchissement du système est fixé à $\Delta_r = 2$ minutes.
- La réduction du niveau de priorité des arrivées après atterrissage est maintenue à $\delta_{prio} = 30$ minutes.

Dans cette configuration, les deux méthodes de résolution de conflits intégrant les séquences à réaliser sont comparées.

9.2.2 Résultats

La figure 9.1 montre le délai moyen par avion (sur la journée) en fonction de l'horizon de prédiction des séquences : pour les deux méthodes de résolution, le délai moyen est légèrement supérieur avec un horizon réduit à 10 minutes. Les délais obtenus avec la méthode déterministe se stabilisent à partir

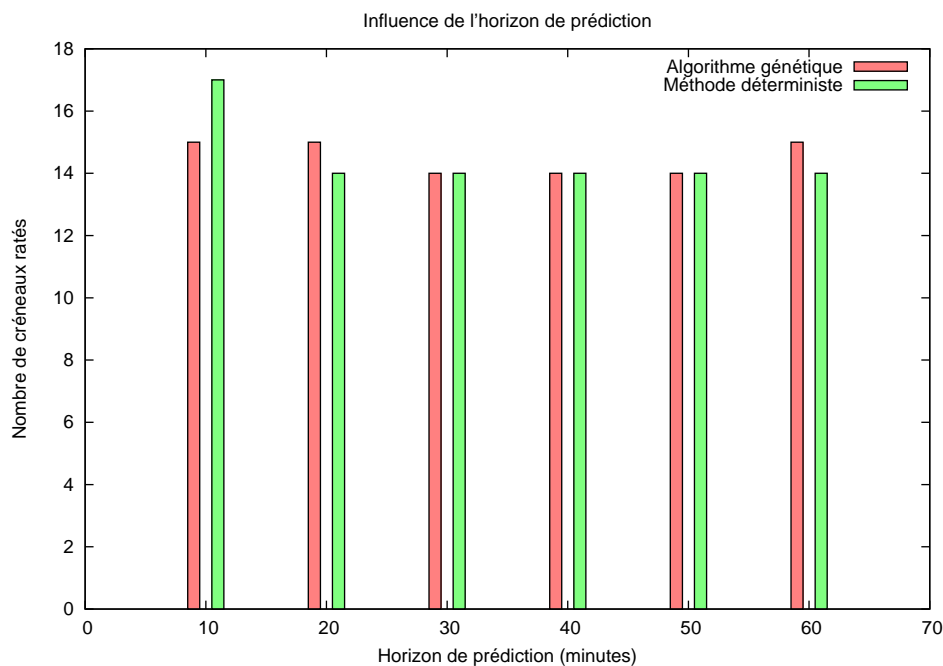


FIG. 9.2 – Influence de l'horizon de prédiction sur les créneaux

d'un horizon de 20 minutes mais restent significativement supérieurs à ceux obtenus avec l'algorithme génétique : les résultats moins réguliers obtenus avec ce dernier correspondent toujours à de bien meilleures solutions.

La figure 9.2 présente le nombre de créneaux ratés (sur la journée) en fonction de l'horizon de prédiction des séquences. Parmi ces créneaux ratés, 14 ne sont pas réalisables (à cause de l'heure de libération tardive du parking et du temps de roulage minimal), mais avec les deux méthodes, d'autres créneaux (réalisables) sont ratés lorsque l'horizon est réduit à 10 minutes. Pour des horizons supérieurs, il n'y a plus de créneau réalisable raté avec la méthode déterministe alors qu'il en reste un, pour des horizons de 20 et 60 minutes avec l'algorithme génétique. L'analyse de ces cas particuliers de créneaux ratés alors qu'ils sont réalisables (et qu'ils ne sont pas ratés avec d'autres horizons de prédiction) fait apparaître qu'il s'agit toujours des mêmes départs, dont le temps minimal de roulage implique qu'ils ne peuvent atteindre la piste que très peu de temps avant la fin de leurs créneaux (moins de trente secondes). De ce fait, le respect de ces créneaux est extrêmement sensible à toute forme de perturbation antérieure (retard additionnel dû à une attente imposée, un détour ou encore des vitesses de roulage plus lentes que prévu) : en cherchant à minimiser les délais des vols non régulés, l'algorithme génétique peut parfois aboutir à des situations dans lesquelles un créneau ne peut plus être respecté. Dans la réalité, ce type de créneau est souvent annulé ou retardé. Dans les simulations, de tels cas peuvent donc être considérés comme non significatifs.

Plus globalement, ces simulations montrent que l'algorithme génétique parvient toujours à trouver de meilleurs résultats que la méthode déterministe, quelle que soit la taille de l'horizon, et que l'horizon de prédiction pour les séquences semble suffisant à partir de 20 minutes : la valeur de 30 minutes est toutefois conservée pour la suite, car elle ne pose pas de problème de calcul de séquence et correspond à la valeur souhaitée par les opérationnels.

9.3 Taux d'incertitude et pas de rafraîchissement

9.3.1 Présentation

Dans la réalité, le taux d'incertitude sur la vitesse des avions peut s'avérer supérieur à 20% : l'étude des traces radar des avions à Roissy montre de grandes variations des vitesses de roulage adoptées (même lorsque ces mesures sont faites au même endroit, pour le même type d'appareil) et ces variations correspondent plutôt à un taux d'incertitude de l'ordre de 50%, par rapport à la vitesse maximale.

Un tel niveau d'incertitude sur les vitesses pose naturellement de nombreux problèmes pour la résolution des conflits entre avions au roulage : la prévision des trajectoires ne peut plus être envisagée qu'avec des horizons réduits (inférieurs à 5 minutes) et la réalisation de séquences de décollages prévues 30 minutes en avance s'avère a priori beaucoup plus délicate. Dans ce mode de fonctionnement, le système doit nécessairement rester beaucoup plus flexible : pour éviter une augmentation significative des délais, les décisions prévues dans les situations précédentes doivent pouvoir être modifiées ou annulées, en fonction de l'évolution réelle du trafic. Dans les simulations, la flexibilité du système est représentée par le pas de résolution Δ_r : avant chaque résolution, le système est censé être mis à jour et les zones d'incertitudes des avions sont donc réduites. Les effets de l'augmentation du taux d'incertitude sur les vitesses doivent donc pouvoir être modérés par une diminution du pas de rafraîchissement Δ_r .

L'objectif est donc d'étudier par simulation l'influence du taux d'incertitude et du pas de rafraîchissement sur le délai généré et le nombre de créneaux ratés : pour chaque valeur envisagée du taux d'incertitude (de 0 et 50%) et du pas (1 ou 2 minutes), une simulation est effectuée sur la journée complète. Comme justifié dans la partie précédente, l'horizon de prédiction pour les séquences d'avions sur les pistes reste fixé à 30 minutes dans tous les cas.

9.3.2 Résultats

La figure 9.3 présente les délais et les créneaux ratés résultant des différentes simulations, pour les deux valeurs du pas de rafraîchissement Δ_r envisagées (1 et 2 minutes).

Dans tous les cas de figure, l'influence du taux d'incertitude est conséquente pour les valeurs élevées de ce dernier : lorsque le taux d'incertitude passe de 20% à 50%, le délai moyen fait plus que doubler avec un pas de rafraîchissement de deux minutes, et augmente de plus de 50% avec un pas de rafraîchissement d'une minute. De même, le nombre de créneaux ratés augmente, surtout avec l'algorithme génétique. Par ailleurs, des cas d'échec de la résolution de conflits apparaissent avec la méthode déterministe (cf. tableau 9.4) : ces situations de trafic non résolues correspondent à des modifications soudaines des séquences prévues sur les pistes (dues aux délais importants des départs en cours de roulage), qui impliquent un nouveau classement des avions, incompatible avec leur situation courante. Les quelques révisions du classement explorées par la méthode déterministe ne permettent pas, dans ces situations particulières, de trouver un classement admissible. Ceci montre la nécessité d'envisager de nombreux classements (comme le fait l'algorithme génétique) pour résoudre correctement certains conflits plus compliqués.

La réduction du pas de rafraîchissement du système de 2 à 1 minute permet en revanche d'améliorer le fonctionnement des méthodes de résolution lorsque l'incertitude est élevée : pour un taux d'incertitude de 50%, le délai moyen passe de 90 secondes par avion (pour un pas de rafraîchissement de deux minutes) à 47 secondes par avion (pour un pas d'une minute) avec la résolution par algorithme génétique. Le nombre de créneaux ratés, ainsi que les cas d'échecs de la méthode déterministe se

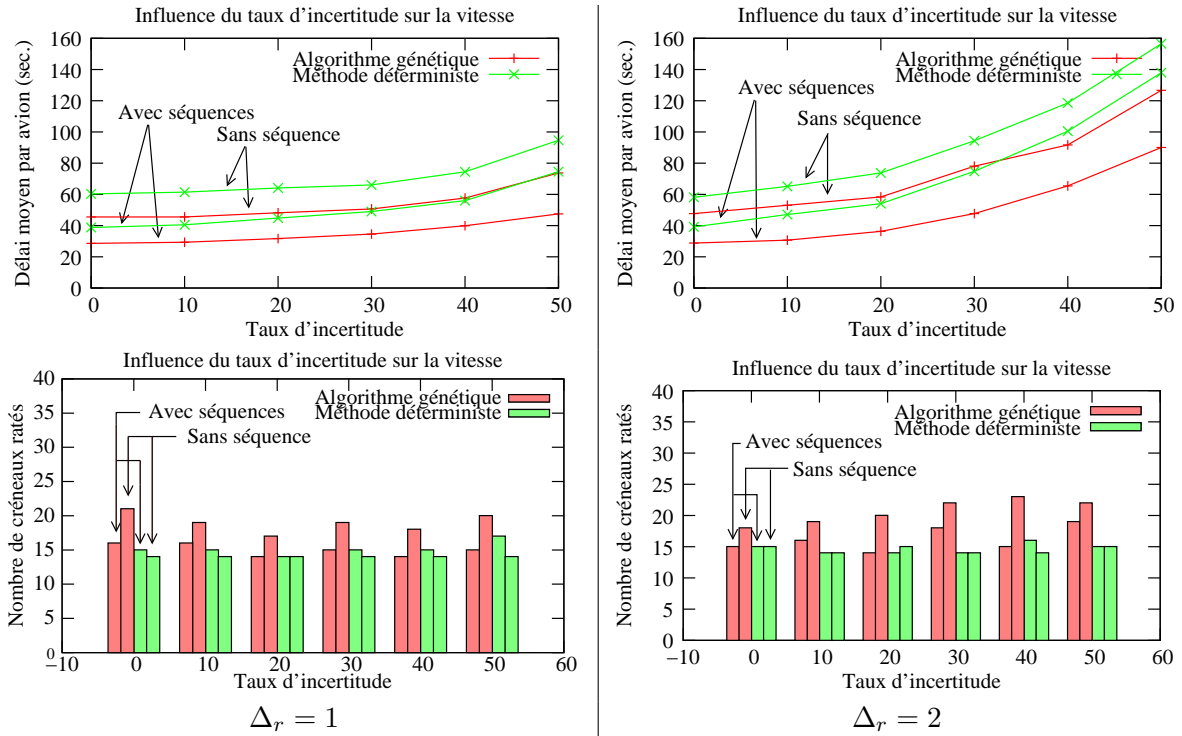


FIG. 9.3 – Influence du taux d'incertitude et du pas de rafraîchissement

		Taux d'incertitude (%)					
		0	10	20	30	40	50
$\Delta_r = 1$	Avec séquences						
	Sans séquence						3
$\Delta_r = 2$	Avec séquences					1	4
	Sans séquence	2			2	2	1

FIG. 9.4 – Cas d'échecs de la résolution de conflits avec la méthode déterministe

font également plus rares. Cela montre qu'un rafraîchissement plus fréquent du système (qui correspond à une augmentation de sa flexibilité) permet d'absorber en partie les effets négatifs d'un niveau d'incertitudes élevé.

Ces résultats confirment également le meilleur fonctionnement de l'algorithme génétique par rapport à la méthode déterministe : le délai moyen est globalement diminué d'au moins un tiers dans toutes les configurations et la résolution des conflits (par algorithme génétique) se montre plus robuste à l'augmentation des incertitudes (aucun cas d'échec et une augmentation relative du délai moyen plus faible entre 20% et 50% d'incertitude). Le seul point négatif concerne le nombre de créneaux réalisables ratés (plus nombreux avec l'algorithme génétique). Ce point est à rapprocher du dilemme entre minimisation des délais et respect des créneaux imposés, déjà discuté au 8.2.4 : une évaluation plus précise des risques de non respect de créneaux en fonction de la situation des avions à la fin de l'horizon de prédiction s'avère d'autant plus nécessaire avec des incertitudes élevées.

Enfin, l'intégration des séquences d'avions prévues sur les pistes dans le critère à optimiser pour résoudre les conflits au roulage se montre bénéfique, même avec un taux d'incertitude élevé et une plus grande flexibilité : pour les deux méthodes de résolution, la prise en compte des séquences permet une nette diminution du délai moyen et favorise globalement le respect des créneaux imposés. Pour la méthode déterministe, le nombre de cas d'échecs est également diminué. Globalement, la réalisation des séquences optimales fournit un objectif à long terme, qui permet de mieux résoudre les conflits au roulage, en rendant les résolutions successives plus cohérentes.

9.4 Niveau de priorité entre départs et arrivées

9.4.1 Présentation

Le paramètre δ_{prio} défini au 7.2.1 a pour fonction de moduler le rapport de priorité entre départs et arrivées au roulage, avec la méthode déterministe : sans ce paramètre, le classement utilisé (dédit des heures de décollage et d'atterrissage réalisées ou prévues dans les séquences) rendrait les arrivées systématiquement prioritaires sur les départs au roulage, ce qui est contraire aux objectifs opérationnels. La valeur $\delta_{prio} = 30$ minutes assure l'inverse, mais rend le délai moyen des arrivées bien supérieur à celui obtenu avec l'algorithme génétique (pour un délai moyen des départs équivalent, lorsque le critère prend en compte les séquences). Si cette valeur est revue à la baisse, les départs sont moins prioritaires : le délai des arrivées doit diminuer, mais celui des départs risque d'augmenter : l'objectif de cette partie est de quantifier précisément ces effets, afin de pouvoir envisager les valeurs les plus appropriées pour ce paramètre de la méthode déterministe.

Pour cela, différentes simulations sont effectuées sur la journée complète, avec la méthode déterministe intégrant les séquences à réaliser et en faisant varier ce paramètre de 0 à 30 minutes.

9.4.2 Résultats

La figure 9.5 montre le délai moyen par avion en fonction du paramètre δ_{prio} . Au delà de 20 minutes, le délai moyen (aussi bien pour les arrivées que pour les départs) est sensiblement constant. Ceci correspond logiquement au temps maximal de roulage des départs : pour ces valeurs de δ_{prio} , les arrivées sont systématiquement moins prioritaires que les départs, donc la méthode de résolution déterministe trouve le plus souvent les mêmes solutions aux différents conflits. En dessous de cette valeur, le délai moyen des départs diminue et celui des arrivées augmente avec δ_{prio} . Globalement, le délai par avion (arrivées et départs confondus) augmente avec ce paramètre, bien que cette augmentation soit relativement faible. Cela peut s'expliquer par le nombre d'arrivées (719) supérieur au nombre

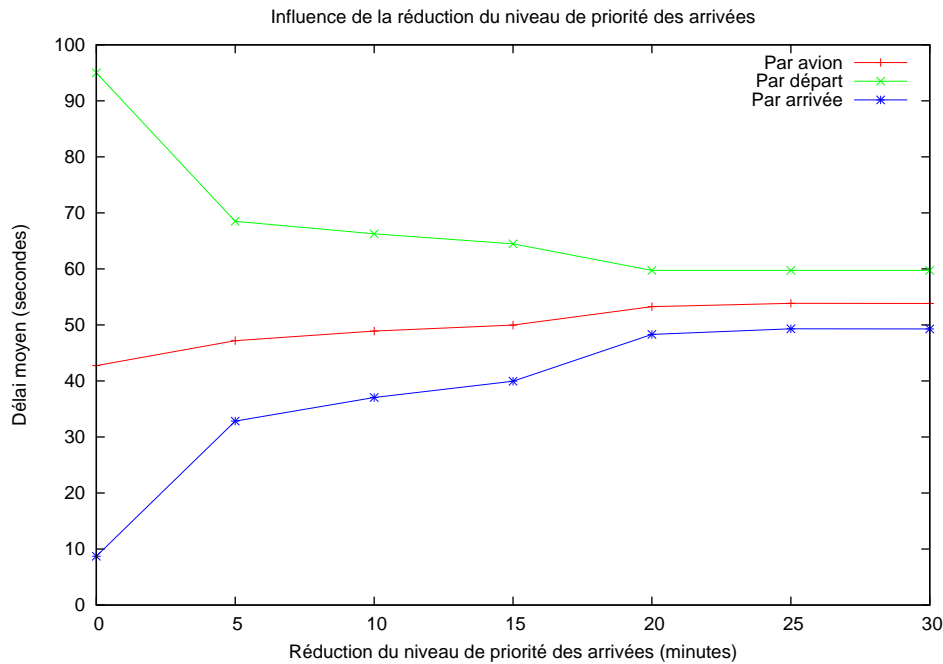


FIG. 9.5 – Délai moyen par avion en fonction de la réduction du niveau de priorité des arrivées

de départs non régulés (457), qui implique que le délai des arrivées influence davantage le délai moyen global. Cela montre cependant que la stratégie visant à favoriser les départs sur les arrivées au roulage n'est pas forcément la plus efficace, en terme de délai global au roulage.

La figure 9.6 présente le nombre de créneaux ratés pour les différentes valeurs de δ_{prio} : ce nombre correspond au minimum atteignable (14 créneaux ratés) pour toutes les valeurs non nulles testées du paramètre. Par contre, lorsque $\delta_{prio} = 0$, les arrivées sont systématiquement prioritaires sur les départs au roulage (puisque leur heure d'atterrissage est forcément inférieure à l'heure prévue de décollage des départs), et ceci cause finalement des créneaux ratés supplémentaires (4 cas sur cette journée).

Les valeurs intéressantes pour ce paramètre correspondent donc à l'intervalle [5; 20] minutes, et le choix de cette valeur dépend des objectifs opérationnels : la volonté de donner priorité aux départs correspond à une stratégie globale de gestion du trafic, visant à minimiser les retards en début de vol, ce qui dépasse le cadre de la simple minimisation des temps de roulage. Quelque soit la valeur choisie pour ce paramètre, le délai moyen (départs et arrivées confondus) obtenu avec la méthode déterministe ne varie pas suffisamment pour atteindre celui obtenu avec l'algorithme génétique (qui se situe entre 10 et 20 secondes de moins par avion).

9.5 Conclusion

Dans ce chapitre, les simulations arithmétiques d'une journée de trafic sont utilisées pour quantifier l'influence de différents paramètres.

L'horizon de prédiction des séquences d'avions sur les pistes n'a finalement plus grande influence sur la résolution des conflits au roulage dès lors qu'il est supérieur à 20 minutes, ce qui signifie que les séquences calculées sont robustes aux situations successives rencontrées et globalement respectées lors de la résolution, et qu'un horizon plus important n'améliore pas la qualité des séquences.

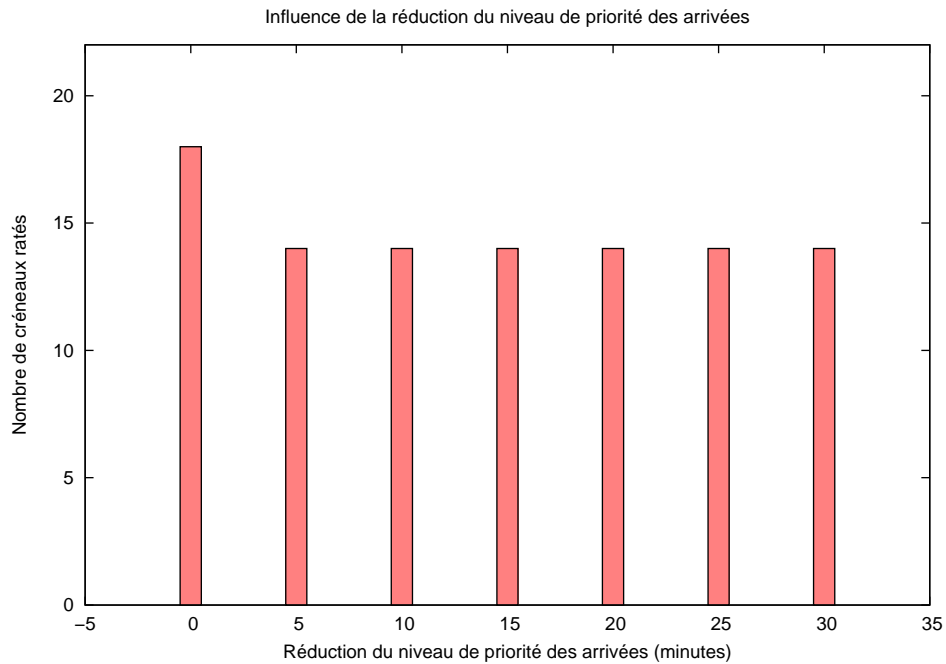


FIG. 9.6 – Nombre de créneaux ratés en fonction de la réduction du niveau de priorité des arrivées

En revanche, l'influence des incertitudes sur les vitesses de roulage des avions est considérable : le délai généré croît rapidement avec le taux d'incertitude. Ainsi, tout moyen permettant de réduire ces incertitudes ou du moins à atténuer leurs effets peut contribuer fortement à diminuer les délais : la diminution du pas de rafraîchissement du système permet notamment d'augmenter sa flexibilité et a pour effet d'amortir une grande partie des délais. De la même façon, l'anticipation des séquences d'avions à réaliser sur les pistes permet de mieux ordonner le trafic en visant des objectifs à plus long terme, ce qui aboutit à des délais moyens moins sensibles aux incertitudes. Dans ce contexte, les projets de guidage automatique des avions au sol sont également extrêmement prometteurs, puisque cela apporterait une nette amélioration de la prévision des trajectoires des avions, donc permettrait une résolution plus optimisée des conflits au roulage.

L'importance du niveau de priorité des départs a pu également être étudiée avec la méthode de résolution déterministe : rendre les départs plus prioritaires permet de renforcer le respect des créneaux imposés par le CFMU et de réduire leur délai, bien que ce soit au détriment de celui des arrivées. Cependant, le délai global (départs et arrivées confondus) évolue peu et reste toujours supérieur au délai moyen obtenu avec l'algorithme génétique.

Conclusions et perspectives

L'étude des séquences d'avions sur les pistes présentée dans le cadre de cette thèse permet d'en dégager les principales caractéristiques : d'un point de vue opérationnel, la gestion des pistes est un problème à résoudre en temps réel et à plus long terme (30 minutes) que la résolution des conflits au roulage, bien qu'elle doive s'adapter à chaque situation de trafic. Les mouvements étant soumis à des règles de séparation sur les pistes (qui dépendent notamment des catégories de turbulence de sillage de chacun d'entre eux), l'ordre des avions y accédant peut fortement influencer les retards générés.

L'optimisation de ces séquences peut s'effectuer par une approche par classement : l'ordre des avions est déterminé de façon à optimiser l'utilisation des pistes en fonction des règles prédéfinies. Une résolution par itérations sur les situations successives d'une journée permet alors d'obtenir des séquences d'avions sur les pistes pour une journée complète et ainsi de quantifier la part du retard dû aux pistes. Bien que ces dernières semblent souvent être le point le plus problématique de l'aéroport en terme de capacité, l'optimisation des séquences d'avions montre qu'une meilleure gestion de la phase de roulage permettrait de réduire considérablement les retards qu'elles génèrent : le retard moyen par départ en ne considérant que les pistes peut être réduit à 35 secondes avec des séquences optimales, contre 97 secondes en incluant les conflits au roulage (mais sans considérer de séquences optimales à réaliser sur les pistes). Cela montre que la phase de roulage doit pouvoir être adaptée plus efficacement aux contraintes imposées par les pistes et qu'un lien entre les deux pourrait fortement réduire les retards.

L'analyse du trafic actuel sur les aéroports permet de définir un modèle de simulation intégrant les incertitudes lors de la phase de roulage. La résolution des conflits au roulage se base alors sur ce modèle et une approche par niveaux de priorité permet de formaliser le problème et ainsi de résoudre l'ensemble des situations de trafic. Cette approche détermine les chemins et les rangs de chaque avion à partir desquels les attentes peuvent être définies grâce à un algorithme déterministe. La définition d'une attente stratégique permet de faire attendre les avions en amont afin que chaque avion respecte son heure de décollage prévue. Cette attente permet par ailleurs de fluidifier le trafic et d'éviter la congestion aux abords des pistes.

Les deux méthodes de résolution étudiées se sont révélées efficaces et ont chacune leurs points d'intérêt :

- **La méthode déterministe** considère les avions les uns après les autres, selon un classement initial préétabli et détermine la trajectoire de chaque avion en fonction des avions précédemment traités. Elle a l'avantage d'avoir un temps d'exécution très rapide et devient très intéressante au niveau des retards des départs lorsque les séquences optimales d'avions sur les pistes sont prises en compte. Par contre, dans certaines situations particulières, les classements explorés peuvent s'avérer incompatibles avec la situation courante des avions et créer des cas d'échec de résolution de la méthode.
- **L'algorithme génétique**, qui est en fait une hybridation entre un algorithme génétique et la méthode déterministe sus-citée, considère le problème de façon plus globale. Il obtient alors

de meilleurs résultats, principalement pour les arrivées (moins bien gérées avec la méthode déterministe) mais son temps de calcul, bien supérieur, le destine à des systèmes dont le temps de réponse serait supérieur à la minute. Dans l'ensemble des simulations étudiées, aucun cas d'échec de résolution n'a été rencontré.

Le lien entre la gestion des pistes et la phase de roulage s'avère prometteur, en terme de réduction des retards et de pertinence des séquences de décollage calculées 30 minutes en avance. L'approche par classement contribue grandement à la qualité des résultats obtenus : les pistes imposent d'ordonner les avions. Dès lors, classer les avions au roulage s'avère logique (l'ordre est fourni par les séquences) et efficace (les avions arrivent dans l'ordre au niveau des pistes, ce qui limite les attentes et les points de congestion).

Le simulateur ATOS permet ainsi d'étudier l'efficacité des méthodes de résolution des conflits au roulage en fonction de différents facteurs caractérisant la gestion du trafic au sol : la variation du retard et le respect des créneaux de décollage ont pu ainsi être étudiés en fonction de l'horizon de prédiction utilisé, du taux d'incertitude sur les vitesses de roulage, du pas de rafraîchissement du système ou encore du niveau de priorité des départs sur les arrivées. Dans ce domaine, le simulateur offre de nombreuses perspectives d'applications pour l'étude des futurs concepts liés à la gestion du trafic au sol sur les grands aéroports :

- Une première application concerne naturellement la définition d'un outil de planification du cheminement des avions sur l'aéroport : l'objectif est de trouver les chemins les plus adéquats à imposer aux vols, en fonction de la situation de trafic à venir et des procédures opérationnelles autorisées. Les deux méthodes de résolution présentées peuvent être envisagées pour cela, en fonction du temps de réponse souhaité et des contraintes à prendre en compte. Ce type d'application, très proche du monde opérationnel, pose des problèmes de robustesse et nécessite l'intégration de données plus précises (certifiées cohérentes avec les règles d'exploitation de l'aéroport) et plus complètes (incluant notamment la description des aires d'attente utilisables, des aires de stationnement, éventuellement des postes de dégivrage disponibles, etc.)
- Un second sujet d'actualité concerne la maximisation des temps d'attente au parking [Pujet 99, Burgain 08, Simaiakis 09], pour permettre aux avions de minimiser leur consommation de carburant au roulage, et à l'aéroport de réduire sa pollution et ses nuisances sonores : dans ce concept (déjà mis à mal dans le passé), le risque est de créer des périodes de pénurie d'avions devant la piste (pendant lesquelles les contrôleurs ne peuvent plus optimiser la séquence) alors que la demande réelle de trafic est élevée. Pour évaluer ce réglage délicat, un seuil d'attente maximal devant la piste peut être défini pour calculer une attente stratégique au parking : l'étude des variations du retard et du nombre d'avions en attente devant la piste en fonction de ce seuil doit permettre de déduire une valeur opérationnellement acceptable pour ce dernier.
- Dans le même registre, l'optimisation du trafic au sol peut également être élargie : dans les simulations présentées, les parkings et les pistes affectés aux vols sont considérés comme des données d'entrée du problème. Il peut également être judicieux de chercher à optimiser ces affectations (choix des portes d'embarquement ou des pistes par exemple) pour chercher à réduire au maximum les temps de roulage des avions.
- Enfin, l'anticipation des séquences de décollage sur les pistes reste un point central pour les outils de type DMAN actuels : la pertinence des séquences prévues est souvent dégradée par les nombreuses incertitudes relatives aux heures de départ des vols et à leurs temps de roulage effectifs. Dans ce contexte, l'intégration des séquences de décollage dans la résolution des conflits au roulage présentée dans cette thèse est prometteuse. La possibilité de figer autant que possible ces séquences un certain temps en avance peut être étudiée par simulation, en évaluant là encore les variations du retard que chaque stratégie peut impliquer.

Glossaire

- AMAN** : *Arrival MANagement*. Gestion des vols à l'arrivée d'un aéroport pour préparer leur séquençement.
- ASMGCS** : *Advanced Surface Movement Guidance and Control System*. Concepts définis par l'OACI pour le nouveau système de contrôle des avions au sol.
- ATFM** : *Air Traffic Flow Management*. Régulation des flux de trafic aérien.
- ATM** : *Air Traffic Management*. Gestion du trafic aérien.
- ATOS** : *Airport Traffic Optimization Simulator*. Simulateur d'aéroport développé par la DTI/R&D
- AVISO** : *Aide à la Visualisation au SOL*. Outil d'aide à la décision de type SMAN.
- APO** : *Algorithmes Parallèles et Optimisation*. Équipe appartenant à l'IRIT.
- CATS** : *CAML All purpose Traffic Simulator*. Simulateur de trafic aéroportuaire.
- 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.
- Data-Link** : *Liaison de données*. Systèmes de transmission automatique des données entre le sol et les avions ou entre avions.
- D-GPS** : *Differential Global Positionning System*.
- DMAN** : *Departure MANagement*. Gestion des vols au départ d'un aéroport.
- DSNA** : *Direction des Services de la Navigation Aérienne*.
- DTI** : *Direction de la Technique et de l'Innovation*.
- ENAC** : *Ecole Nationale de l'Aviation Civile*.
- EUROCONTROL** : *European Organisation for the Safety of Air Navigation*. Organisme Européen chargé de la gestion du trafic aérien en Europe.
- FAA** : *Federal Aviation Administration*. Organisme de l'aviation civile américaine.
- FaCiLe** : *Functional Constraint Library*. Librairie de programmation par contraintes.
- FCFS** : *First Come First Served*. Premier Arrivé Premier Servi.
- GPS** : *Global Positionning System*. Système de positionnement par satellite provenant du département de la défense américaine.
- IRIT** : *Institut de Recherche en Informatique de Toulouse*.
- LOG** : *Laboratoire d'Optimisation Globale*. Structure de recherche commune à l'ENAC et au CENA.
- MAESTRO** : *Moyen d'Aide à l'Écoulement Séquencé du Trafic avec Recherche d'Optimisation*. Outil d'aide au contrôle de type AMAN.

MIP : *Mixed Integer Programming*

MILP : *Mixed Integer Linear Programming*

OACI : *Organisation de l'Aviation Civile Internationale.*

OPAS : *Outil de Planification ATM et de Simulation.* Simulateur de trafic aéroportuaire.

PL : *Programmation Linéaire.*

PLNE : *Programmation Linéaire en Nombre Entier.*

POM : *Planification, Optimisation et Modélisation du trafic aérien.* Pôle appartenant au domaine Recherche & Développement de la DTI

PPC : *Programmation Par Contraintes.*

Push-back : Sortie d'un parking en marche arrière, avec l'aide d'un véhicule tracteur.

SALADIN : *Système d'Aide à LA Détection d'INcursions.* Extension d'AVISO.

SID : *Standart Instrucment Departure.*

SIMMOD : *SIMulation MODel.* Simulateur de trafic aérien développé par la FAA.

SMAN : *Surface MANagement.* Gestion du trafic au roulage sur un aéroport.

TAAM : *Total Airspace and Airport Modeler.* Simulateur de trafic aérien.

Bibliographie

- [Alliot 96] J.-M. ALLIOT. *Technique d'optimisation stochastique appliquées aux problèmes du contrôle aérien*. Thèse d'habilitation : Institut National Polytechnique de Toulouse, 1996. *Cité p 84*
- [Atkin 04] J. ATKIN, E. BURKE, J. GREENWOOD et D. REESON. Departure runway scheduling at london heathrow airport. *4th EU/ME Workshop, The Design and Evaluation of Advanced Hybrid Meta-Heuristics*, octobre 2004. *Cité p 13*
- [Atkin 08] J. ATKIN. *On-line decision support for take-off runway scheduling at London Heathrow airport*. Thèse : University of Nottingham, 2008. *Cité p 13*
- [Balakrishna 09] P. BALAKRISHNA, R. GANESAN et L. SHERRY. Application of reinforcement learning algorithms for predicting taxi-out times. *8th Seminar of Air Traffic Management*, juin 2009. *Cité p 12*
- [Balakrishnan 07a] H. BALAKRISHNAN et B. CHANDRAN. A dynamic programming algorithm for robust runway scheduling. *Proceedings of the American Control Conference*, juillet 2007. *Cité p 59*
- [Balakrishnan 07b] H. BALAKRISHNAN et Y. JUNG. A framework for coordinated surface operations planning at dallas-fort worth international airport. *AIAA Guidance, Navigation and Control Conference*, août 2007. *Cité p 13*
- [Bourdon 98] B. BOURDON. Moyen d'aide à l'écoulement séquencé du trafic. *STNA, Revue technique n°55*, novembre 1998. *Cité p 9*
- [Burgain 08] P. BURGAIN, E. FERON et J.-P. CLARKE. Collaborative virtual queue : Fair management of congested departure operations and benefit analysis. *CoRR*, 2008, vol abs/0807.0661. *Cité p 12, 112*
- [Böhme 07] D. BÖHME. Integration of aman, dman and sman with a-smgcs. *3rd AP21 Workshop, NASA Ames*, octobre 2007. *Cité p 12*
- [CENA 05] Centre d'Étude de la Navigation Aérienne CENA. *FaCiLe*, 2005. <http://pom.tls.cena.fr/facile/index.html.fr>. *Cité p 35*
- [Curtis Holden 03] T. CURTIS HOLDEN et F. WIELAND. Runway schedule determination by simulation optimization. *Winter Simulation Conference*, décembre 2003. *Cité p 15*
- [De Pablo 05] J.M. DE PABLO et P. PINA. Benefits obtained from the estimation and distribution of realistic taxi times. *6th Seminar of Air Traffic Management*, juin 2005. *Cité p 12*
- [Deau 08] R. DEAU, J.-B. GOTTELAND et N. DURAND. Runways sequence and ground traffic optimisation. *International Conference on Research in Air Transportation*, juin 2008. *Cité p 2*

- [Deau 09] R. DEAU, J.-B. GOTTELAND et N. DURAND. Airport surface management and runways scheduling. *8th Seminar of Air Traffic Management*, juin 2009. Cité p 2
- [Deau 10] R. DEAU et J.-B. GOTTELAND. Algorithmes d'optimisation appliqués au trafic aéroportuaire. *Recherche Opérationnelle et Aide à la Décision*, février 2010. Cité p 2
- [Desaulniers 05] G. DESAULNIERS, J. DESROSIERS et M. M. SOLOMON. *Column Generation*. Springer, 2005. Cité p 33
- [DNSA/DTI 07] Direction des Services de la Navigation Aérienne / Direction de la Technique et de l'Innovation DNSA/DTI. *CAML All purpose Traffic Simulator*, 2007. <http://pom.tls.cena.fr/CATS/>. Cité p 15
- [DNSA/DTI/R&D 07] Domaine Recherche & Développement DNSA/DTI/R&D DIRECTION DES SERVICES DE LA NAVIGATION AÉRIENNE / DIRECTION DE LA TECHNIQUE ET DE L'INNOVATION. *Airport Traffic Optimization Simulator*, 2007. <http://pom.tls.cena.fr/ATOS/index.html>. Cité p 15
- [Doble 09] N. DOBLE, J. TIMMERMAN, T. CARNIOL, M. KLOPFENSTEIN, M. TANINO et V. SUD. Linking traffic management to the airport surface. *8th Seminar of Air Traffic Management*, juin 2009. Cité p 12
- [DSNA 08] Direction des Services de la Navigation Aérienne DSNA. *MANEX, MANuel d'Exploitation de l'aéroport de Roissy*, 2008. Cité p 21
- [Durand 94] N. DURAND, J.-M. ALLIOT et J. NOAILLES. Algorithmes génétiques : un croisement adapté aux fonctions partiellement séparables. *Journées 'Evolutions Artificielles 94'*, 1994. Cité p 85
- [Durand 98] N. DURAND et J.-M. ALLIOT. Genetic crossover operator for partially separable functions. *Proceedings of the third annual Genetic Programming Conference*, 1998. Cité p 85
- [Eiben 03] A.E. EIBEN et J.E. SMITH. *Introduction to Evolutionary Computing*. Springer, 2003. Cité p 79
- [EMN 10] École des Mines de Nantes EMN. *Choco Solver*, 2010. <http://www.emn.fr/z-info/choco-solver/index.html>. Cité p 35
- [FAA 02] FAA et ATAC CORPORATION. *How SIMMOD Works*. Mountain View, 2002. www.tc.faa.gov/acb300/how_AASW_works.pdf. Cité p 15
- [FAA 10] FAA. *Colaborative Decision Making (CDM)*, 2010. <http://cdm.fly.faa.gov/>. Cité p 12
- [Goldberg 87] D. GOLDBERG et J. RICHARDSON. Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*. Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc. Cité p 83
- [Goldberg 89] D. GOLDBERG. *Genetic algorithms*. Addison Wesley, 1989. Cité p 81
- [Gondzio 01] J. GONDZIO. *Numerical techniques in interior point methods for optimization*, 2001. Cité p 33
- [Gotteland 04] J.-B. GOTTELAND. *Optimisation du trafic au sol sur les grands aéroports*. Thèse : Institut National Polytechnique de Toulouse, 2004. Cité p 5, 14, 57, 58, 59, 62, 68, 69, 77, 89, 92

- [Granger 01] Geraud GRANGER, Nicolas DURAND et Jean-Marc ALLIOT. Optimal resolution of en route conflicts. *ATM 2001*. 2001. *Cité p 89*
- [Granger 02] G. GRANGER. *Détection et résolution de conflits aériens : modélisation et analyse*. Thèse : École Polytechnique, 2002. *Cité p 85*
- [Horst 95] Reiner HORST et Hoang TUY. *Global Optimization, Deterministic Approaches*. Springer, 1995. *Cité p 33*
- [Huart 09] A. HUART et F. SEMET. Optimisation de la planification de tournées de cars. *Journée Franciliennes de Recherche Opérationnelle JFRO*, 2009. *Cité p 33*
- [IBM 10] IBM. *ILOG CPLEX*, 2010. <http://www-142.ibm.com/software/products/fr/fr/ilogcple>. *Cité p 35*
- [Janikow 91] C. Z. JANIKOW et Z. MICHALEWICZ. An experimental comparison of binary and floating point representation in genetic algorithm. *International Conference on Genetic Algorithms*, 1991. *Cité p 79*
- [Kim 10] S.H. KIM, E. FERON et J.-P. CLARKE. Airport gate assignment that minimizes passenger flow in terminals and aircraft congestion on ramps. *AIAA Guidance, Navigation and Control conference*, 2010. *Cité p 12*
- [Laborie 07] P. LABORIE et D. GODARD. Self-adapting large neighborhood search : Application to single-mode scheduling problems. *Proceedings MISTA-07*, 2007. *Cité p 44*
- [Legge 08] J. LEGGE et B. LEVY. Departure taxi time predictions using asde-x surveillance data. *26th International Congress Of Aeronautical Sciences*, septembre 2008. *Cité p 12*
- [Malika 10] W. MALIKA, G. GUPTAB et Y. C. JUNG. Managing departure aircraft release for efficient airport surface operations. *AIAA Guidance, Navigation and Control conference*, 2010. *Cité p 12*
- [Michalewicz 92] Z. MICHALEWICZ. *Genetic algorithms + data structures = evolution programs*. Springer-Verlag, 1992. *Cité p 82*
- [Michalewicz 96] Z. MICHALEWICZ et M. SCHOENAUER. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 1996, vol 4, p 1–32. *Cité p 82*
- [Minton 93] S. MINTON, A. PHILIPS, M. D. JOHNSTON et P. LAIRD. Minimizing conflicts : A heuristic repair method for constraint-satisfaction and scheduling problems. *Journal of Artificial Intelligence Research 1*, 1993. *Cité p 44*
- [Miquel 01] I. MIQUEL et Q. SHEN. *Solution Techniques for constraint satisfaction problems : advanced approaches*. *Artificial Intelligence Review*, p 269–293. Springer Netherlands, 2001. *Cité p 34*
- [Mitchell 96] M. MITCHELL. *An introduction to Genetic Algorithms*. MA : MIT Press, 1996. *Cité p 79*
- [Norin 09] A. NORIN, T. ANDERSSON GRANBERG, P. VÄRBRAND et D. YUAN. Integrating optimization and simulation to gain more efficient airport logistics. *8th Seminar of Air Traffic Management*, juin 2009. *Cité p 12*
- [Papadimitriou 82] C.H PAPADIMITRIOU et K STEIGLITZ. *Combinatorial Optimization : Algorithms and Complexity*. Prentice Hall, Englewood Clifs. NJ (USA), 1982. *Cité p 33*

- [Pick 07] A. PICK. Validation results of airport total operations planner prototype clou. *7th Seminar of Air Traffic Management*, juillet 2007. *Cité p 13*
- [Pujet 99] N. PUJET, B. DELCAIRE et E. FERON. Input-output modeling and control of the departure process of congested airports. *AIAA Guidance, Navigation and Control Conference*, 1999. *Cité p 112*
- [Rathinam 08] S. RATHINAM, J. MONTOYA et Y. JUNG. An optimization model for reducing aircraft taxi times at dallas fort worth international airport. *26th International Congress Of Aeronautical Sciences*, septembre 2008. *Cité p 13*
- [Roling 08] P.C. ROLING et H. G. VISSER. Optimal airport surface traffic planning using mixed-integer linera programming. *International Journal of Aerospace Engineering*, avril 2008. *Cité p 14*
- [Rust 06] J. RUST, L. BLUME, M. BUCHINSKY, L. EPSTEIN et C. PHELAN. *Dynamic Programming entry for consideration by the New Palgrave Dictionary of Economics*, 2006. *Cité p 32*
- [Schoenauer 97] M. SCHOENAUER et Z. MICHALEWICZ. Boundary operators for constrained parameter optimization problems. *Proceedings of the 7th International Conference on Genetic Algorithms*, 1997, p 322–329. *Cité p 82*
- [Simaiakis 09] SIMAIAKIS et H. BALAKRISHNAN. Queuing models of airport departure processes for emissions reduction. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, août 2009. *Cité p 112*
- [Smeltink 05] J.W SMELTINK, M.J. SOOMER, P.R. DE WAAL et R.D. VAN DER MEI. An optimisation model for airport taxi scheduling. *30th Conference on the Mathematics of Operations Research*, janvier 2005. *Cité p 14*
- [Talbot 08] H. TALBOT. *Algorithme du simplexe, une solution à la programmation linéaire*, 2008. *Cité p 33*
- [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 9*
- [Woeginger 03] G. WOEGINGER. Exact algorithms for np-hard problems : A survey. *Lecture Notes in Computer Sciences*, 2003. *Cité p 26*
- [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 84*