



**HAL**  
open science

# Régulation court terme du trafic aérien et optimisation combinatoire Application de la méthode de génération de colonnes

O. Richard

► **To cite this version:**

O. Richard. Régulation court terme du trafic aérien et optimisation combinatoire Application de la méthode de génération de colonnes. Mathématiques [math]. Institut National Polytechnique de Grenoble - INPG, 2007. Français. NNT: . tel-00545025

**HAL Id: tel-00545025**

**<https://theses.hal.science/tel-00545025>**

Submitted on 9 Dec 2010

**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.

**INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE**

*N° attribué par la bibliothèque*

|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|\_|

**THESE**

pour obtenir le grade de

**DOCTEUR DE L'INP Grenoble**

***Spécialité : Recherche Opérationnelle, Combinatoire et Optimisation***

préparée au LICIT

Laboratoire d'Ingénierie Circulation Transports

ENTPE

INRETS

présentée et soutenue publiquement

par

Olivier RICHARD

Le 29/01/2007

**Régulation court terme du trafic aérien et optimisation  
combinatoire  
Application de la méthode de génération de colonnes**

Directeur de thèse : Maurice QUEYRANNE

Co-directeur de thèse : Rémy FONDACCI

Co-directeur de thèse : Wojciech BIENIA

**JURY**

M. FREIN Yannick  
M. MAHEY Philippe  
M. SEMET Frédéric  
M. QUEYRANNE Maurice  
M. FONDACCI Rémy  
M. BIENIA Wojciech

Président  
Rapporteur  
Rapporteur  
Directeur de thèse  
Co-directeur de thèse  
Co-directeur de thèse



## Résumé

Ce travail a pour objet la résolution d'un problème combinatoire posé dans le cadre de la régulation court terme (ou dynamique) du trafic aérien. On cherche à déterminer pour chaque vol régulable une trajectoire en 4 dimensions réalisable de manière à respecter les contraintes de capacité des secteurs tout en minimisant la somme des coûts des trajectoires choisies. Le problème est modélisé par un programme linéaire mixte. Une représentation ad hoc du système aérien sert de support à la modélisation fine des trajectoires. Un processus global de résolution basé sur la génération de colonnes couplée à la technique de branch-and-bound est détaillé. Les colonnes du problème représentant des trajectoires, la génération de colonnes par le sous problème de tarification se traduit par la recherche de chemins quadridimensionnels sur un réseau continu et dynamique. Un algorithme spécifique basé sur les algorithmes de plus court chemin par marquage et sur la programmation dynamique est développé et testé. Toute la méthode est évaluée sur des instances réelles représentant l'espace aérien géré par la CFMU, l'organisme européen de gestion des flux de trafic aérien. Les résultats obtenus en un temps de calcul compatible avec le contexte opérationnel valident finalement la méthode développée.

### Mots clés :

Régulation dynamique, régulation court terme, trafic aérien, congestion, ATFM, génération de colonnes, branch-and-price, algorithmes de plus court chemin par marquage, CFMU

## Abstract

In this work, we address a combinatorial problem arising in the context of dynamic air traffic regulation. In this problem, four-dimensional trajectories have to be built and assigned to every aircraft in order to suppress any sector overload, while minimizing the cost of the chosen policy. This problem is modelled with a mixed linear program, which is solved through column generation and branch-and-price. Its pricing sub-problem looks for feasible trajectories in a dynamic three dimensional network; to solve it, we developed a specific algorithm based on labelling shortest path algorithms and dynamic programming. Each algorithm is tested on real-world data relative to the European air system under the responsibility of the CFMU (Central Flow Management Unit). Results and computing times validate the whole solving process.

### Keywords

Dynamic regulation, short term regulation, Air traffic, congestion, ATFM, column generation, branch-and-price, labelling shortest path algorithms, CFMU



## Remerciements

*Un seul mot, usé, mais qui brille comme une vieille pièce de monnaie : Merci !*

*(Pablo NERUDA)*

Finalement, un peu plus qu'un seul mot pour remercier tout ceux qui m'ont permis de mener à bien ce travail, directement, indirectement ou parfois même les deux.

Tout d'abord un grand merci à Rémy pour son accompagnement attentif et productif tout au long des années que j'aurai passées au LICIT. Ce mémoire de thèse doit énormément à ses lectures et relectures et Rémy doit avoir passé presque autant de temps que moi sur ce document ! Merci à Wojciech Bienia de m'avoir fait découvrir et aimer la Recherche Opérationnelle, découverte sans laquelle mon parcours professionnel et personnel eut été tout autre. Le professeur Ziliaskopoulos m'a quant à lui fait découvrir l'univers de la recherche et il a donné un sérieux coup de pouce au dossier de démarrage de cette thèse : Thank You Dr. Zi ! Merci également au professeur Queyranne d'avoir accepté d'être mon directeur de thèse et à Jean Louis Pirat d'avoir contribué très positivement à mon dossier de thèse.

Je remercie M. Yannick Frein d'avoir accepté de présider le jury de thèse et j'adresse un très grand remerciement aux deux rapporteurs qui ont accepté de consacrer une part importante de leur temps à la lecture de mon mémoire de thèse : M. Philippe Mahey et M. Frédéric Semet.

Merci à tous les autres qui ont pris la peine de lire ce mémoire pour m'aider à le corriger et à l'améliorer : Olivier Goldschmidt dont les remarques m'ont été très utiles, Charline qui m'a fourni un regard complètement différent mais nécessaire sur ce document et Sophie dont les relectures ont été pour moi une source constante d'enrichissement et de réflexion.

Je ne peux évidemment pas oublier ici Fred Foschiatti sorcier du C++, grand manitou du SQL et ordonnateur des relations antinomiques Borland-Visual sans qui je serais encore en train de patauger dans une librairie absconse. Merci à Bastian Fontaine pour m'avoir aidé à extirper les données cachées dans l'atelier aérien et à Samuel pour l'assistance face aux caprices des unités centrales, serveurs et autres écrans lunatiques. Merci à Marie-Thérèse de sa disponibilité et de son efficacité.

Merci à tous les amateurs de balles et ballons du LICIT d'avoir contribué à l'entretien de ma forme physique et merci enfin à tous ceux que je ne peux pas citer ici et qui contribuent par leur gentillesse et leur présence à faire du LICIT un endroit où j'ai toujours eu du plaisir à aller sur mes vélos successifs, excepté peut être les matins trop pluvieux !



<u>Résumé</u> .....	<u>3</u>
<u>Abstract</u> .....	<u>3</u>
<u>Remerciements</u> .....	<u>5</u>
<u>Introduction Générale</u> .....	<u>9</u>
<u>chapitre I La régulation court terme du trafic aérien et son contexte</u> .....	<u>11</u>
I. Organisation actuelle de la régulation du trafic aérien.....	11
I.1. Objectifs et organisation de l’ATFM en Europe.....	11
I.2. Mise en œuvre opérationnelle .....	12
II. Limites de l’organisation actuelle et réponses de la régulation court terme.....	17
II.1. Les faiblesses indépensables de la régulation actuelle .....	17
II.2. Les principes de la régulation court terme .....	19
III. Principes généraux du régulateur court terme.....	22
III.1. Préliminaires.....	22
III.2. Cahier des charges de la régulation court terme.....	23
III.3. Architecture globale du régulateur court terme.....	26
<u>chapitre II Le module d’optimisation du régulateur court terme</u> .....	<u>29</u>
I. Description du problème d’optimisation principal.....	29
I.1. Description du problème à résoudre.....	29
I.2. Traduction des contraintes et des objectifs dans la modélisation.....	31
I.3. Autres choix de modélisation .....	35
II. Bibliographie sur les méthodes de résolutions candidates .....	37
II.1. Introduction .....	37
II.2. Taxinomie des problèmes de régulation ATFM .....	37
II.3. Etat de l’art sur le GHP et ses extensions.....	40
III. Choix de la méthode de génération de colonnes.....	47
III.1. Pourquoi la programmation linéaire en nombres entiers ?.....	47
III.2. Pourquoi la génération de colonnes ?.....	48
<u>chapitre III Résolution du problème d’optimisation principal par la méthode de génération de colonnes</u> .....	<u>51</u>
I. Notions générales sur la génération de colonnes et notations utilisées.....	51
I.1. La méthode de génération de colonnes .....	51
II. Modélisation du problème maître par un PLNE .....	52
II.1. Formulation mathématique du problème d’optimisation .....	52
II.2. Détails de la formulation .....	56
II.3. Complexité du problème maître .....	57
III. Architecture du système de résolution .....	59
III.1. Architecture globale du processus de résolution.....	59
IV. Formulation des différents modules de résolution .....	62
IV.1. Le problème maître restreint (RMP) .....	62
IV.2. Le processus d’initialisation du RMP .....	63
IV.3. Le sous problème de tarification .....	65
V. Le processus d’obtention d’une solution entière.....	66
V.1. La méthode branch-and-price dans le cas général .....	66
V.2. Application de la méthode de branch-and-price .....	70
<u>chapitre IV Modélisation et résolution du sous problème de tarification</u> .....	<u>77</u>
I. Description détaillée du sous problème de tarification .....	77
I.1. Entrées/sorties du sous problème de tarification.....	77



I.2.	Lien avec les problèmes de plus courts chemins.....	78
I.3.	Bibliographie et taxinomie des problèmes de plus courts chemins pertinents.....	79
II.	Modélisation du sous problème de tarification .....	83
II.1.	Modélisation en un graphe .....	83
II.2.	La fonction coût réduit .....	85
III.	Résolution du sous problème de tarification .....	86
III.1.	Principe général de la résolution .....	86
III.2.	Phase I : plus court chemin : recherche spatiale.....	87
III.3.	Extension des trajectoires partielles .....	91
III.4.	Règles de sélection des feuilles à examiner .....	94
III.5.	Phase II : Programmation dynamique : recherche temporelle .....	97
<u>chapitre V</u>	<u>Mise en œuvre expérimentale et résultats</u> .....	<u>111</u>
I.	Objectifs de l'expérimentation et vue d'ensemble.....	111
I.1.	Objectifs de la mise en œuvre expérimentale.....	111
I.2.	Contexte de la mise en œuvre expérimentale.....	112
I.3.	Vue d'ensemble sur la mise en œuvre expérimentale .....	113
II.	Résolution du sous problème de tarification .....	115
II.1.	Les différents scénarios et les paramètres de l'algorithme .....	115
II.2.	Résultats et temps de calcul .....	119
II.3.	Conclusions et stratégie adoptée pour la résolution du sous problème de tarification .....	128
III.	Résolution globale du problème maître .....	130
III.1.	Boucle de génération de colonnes.....	130
III.2.	Gestion des trajectoires artificielles .....	137
<u>chapitre VI</u>	<u>Conclusion générale et perspectives</u> .....	<u>141</u>
I.	Perspectives d'approfondissement .....	141
I.1.	Approfondissement sur la résolution d'une instance .....	141
I.2.	Intégration de la résolution d'une instance dans le cadre du régulateur court terme .....	144
II.	Bilan global de l'application de la méthode de génération de colonnes .....	145
II.1.	Objectifs initiaux et résultats obtenus .....	145
II.2.	Bilan global .....	146
<u>Appendices</u>	.....	<u>149</u>
I.	Bibliographie.....	149
I.1.	Fonctionnement de l'ATFM et de la CFMU, recherches globales .....	149
I.2.	Le Ground Holding Problem et ses extensions .....	150
I.3.	Le TFMP et ses extensions.....	150
I.4.	Problèmes de routage et de reroutement dynamique .....	151
I.5.	Programmation linéaire et applications, génération de colonnes.....	151
I.6.	Problèmes de plus courts chemins .....	152
I.7.	Optimisation combinatoire: autres. ....	153
I.8.	Sites Internet.....	153
II.	Annexes.....	153
II.1.	Descriptions et démonstrations mathématiques .....	153
II.2.	Lexique et acronymes.....	164

## Introduction Générale

Le trafic aérien connaît un contexte économique et environnemental particulièrement difficile et appelé à le rester : l'augmentation des prix du pétrole et la nécessité de réduire les émissions de gaz à effet de serre plaident pour une organisation du trafic la plus efficace possible. En ce qui concerne les vols, cela devrait notamment se traduire par des trajectoires optimisées en termes de consommation et de temps de parcours. Or, l'impératif premier que constitue la sécurité aérienne a conduit à la mise en place progressive du très complexe système de contrôle et d'organisation existant aujourd'hui dans toutes les zones denses de trafic aérien et qui restreint grandement les trajectoires possibles. Ce système est basé sur la division de l'espace aérien en secteurs gérés par des équipes de contrôleurs et sur l'organisation du trafic selon des routes aériennes et des niveaux de vol qui, en limitant la liberté de chaque avion, facilite le respect de normes strictes de séparation horizontale et verticale des aéronefs, permettant à l'aviation commerciale de rester le moyen de transport le plus sûr. Cette perte de liberté des avions a évidemment un coût en termes d'efficacité des trajectoires et l'organisation actuelle n'est probablement pas l'organisation optimale d'un point de vue théorique : un concept décentralisé tel que le vol sans contraintes (*free flight*) permettant à chaque vol de choisir sa trajectoire en transférant la responsabilité de la séparation aux différents avions (équipés de systèmes d'anticollision embarqués et communicants) est peut-être l'avenir du trafic aérien à long terme. Les énormes difficultés économiques, théoriques et pratiques de ce changement complet de paradigme de régulation (nécessité d'équiper tous les avions, définition des procédures en cas de pannes, incertitude sur le jour J du transfert de la responsabilité des contrôleurs vers les avions...) rendent cependant peu probable une telle évolution à court et moyen termes. Il est donc primordial de perfectionner le système actuel en optimisant notamment les trajectoires, tout en restant dans le contexte des routes aériennes et en utilisant au mieux une des principales ressources limitatrices du trafic : le contrôle aérien en vol et aux aéroports.

C'est dans cet esprit que le LICIT (Laboratoire d'Ingénierie Circulation Transports) travaille depuis quelques années sur un nouveau concept destiné à compléter et améliorer le système actuel de régulation du trafic : la régulation court terme du trafic aérien. Le travail de thèse présenté ici s'attache à l'étude, d'un point de vue « recherche opérationnelle », des problèmes combinatoires apparaissant dans ce contexte. Le problème principal d'optimisation réside dans la détermination et l'allocation d'une trajectoire réalisable à un certain nombre de vols en l'air, de manière à respecter les contraintes de capacité des secteurs tout en minimisant la somme des coûts des trajectoires choisies. Il est important de remarquer à ce stade qu'une attention toute particulière a été apportée au sens premier du terme « opérationnelle » : la philosophie adoptée au cours de ce travail a toujours été de se placer aussi près que possible des conditions « réelles ». On a ainsi cherché à faire le moins possible de concessions sur la modélisation pour rester au plus proche des réalités physiques. Pour des raisons expliquées dans la suite de ce document, nous avons décidé d'attaquer le problème d'optimisation découlant du régulateur court terme par le biais de la programmation linéaire en nombres entiers et plus particulièrement par la méthode de génération de colonnes. La résolution du problème comporte alors deux parties principales interdépendantes : la résolution du programme linéaire en nombres entiers par *branch-and-price* et la résolution du sous problème de tarification qui en découle. La mise en œuvre informatique de ce processus nous a également conduit à effectuer un travail significatif sur les données représentant le système aérien. Le mémoire est organisé de la manière suivante :

Le chapitre I est consacré à la présentation du contexte de la thèse : tout d'abord le contexte global de la régulation du trafic aérien, son organisation et ses limites ; le contexte

plus particulier ensuite de la régulation court terme telle qu'envisagée par le laboratoire LICIT.

Le chapitre II décrit le problème d'optimisation découlant de cette nouvelle organisation de régulation proposée et les contraintes imposées par le contexte. La réflexion ayant conduit au choix de la méthode de résolution de la génération de colonnes, à partir d'une bibliographie poussée, est exposée et justifiée.

Le chapitre III expose tout d'abord la modélisation retenue du problème par un programme linéaire mixte puis détaille l'architecture globale du système de résolution et les différents modules permettant d'aboutir à une solution réalisable du problème : initialisation, résolution des programmes linéaires intermédiaires, rôle du sous problème de tarification et obtention d'une solution réalisable entière par la méthode de *branch-and-bound* combinée à des heuristiques d'arrondi.

Le chapitre IV est consacré à un module particulièrement important de la méthode de génération de colonnes : la résolution du sous problème de tarification (ou *pricing sub-problem*) qui constitue dans ce cas un problème de recherche de chemins sur un réseau tridimensionnel dynamique avec contraintes additionnelles. L'algorithme pour résoudre ce problème est au cœur de la méthode de résolution et est développé dans ce chapitre.

Le chapitre V regroupe la description de la mise en œuvre expérimentale dans une première partie et l'analyse et l'exploitation des résultats de cette mise en œuvre dans une seconde partie. Certains algorithmes annexes de traitement des données seront évoqués dans cette partie.

Enfin le chapitre VI tire le bilan global de l'application de la méthode de génération de colonnes au problème d'optimisation posé par la régulation court terme du trafic aérien.

On trouvera en Appendices les références bibliographiques de ce travail et les annexes : démonstrations mathématiques non intégrées dans le texte et un lexique des acronymes utilisés.

Le travail de thèse présenté ici s'inscrit dans un projet de recherche plus global sur la régulation court terme du trafic aérien, projet qui a fait l'objet d'une convention entre l'INRETS et Eurocontrol et dont les résultats sont compilés dans le rapport final ([3]). C'est dans le cadre de cette convention que les données CFMU utilisées dans ce travail ont été fournies par Eurocontrol.

# chapitre I La régulation court terme du trafic aérien et son contexte

*Dans ce chapitre nous décrivons l'organisation actuelle de la régulation du trafic aérien en Europe, ses objectifs et ses moyens d'action. Les procédures d'allocation des créneaux de décollage par la CFMU (Central Flow Management Unit) sont plus particulièrement détaillées. Dans un second temps, les limites du système actuel face à l'incertitude du système aérien et à l'augmentation continue du trafic sont exposées et expliquées à partir notamment des rapports d'évaluation des performances d'Eurocontrol. Les principes de la régulation court terme et les réponses qu'elle peut apporter aux blocages actuels sont alors décrits. Enfin l'architecture d'un régulateur court terme du trafic aérien tel qu'envisagé dans les travaux du LICIT consacrés à ce sujet est présentée.*

## I. Organisation actuelle de la régulation du trafic aérien

### I.1. Objectifs et organisation de l'ATFM en Europe

#### I.1.a) Qu'est ce que la régulation aérienne ?

L'espace aérien est une ressource limitée qui est soumise à une forte demande dans certaines parties du monde telles que l'Europe de l'Ouest, l'Amérique du Nord et l'Asie du Sud Est. Afin d'assurer le respect des normes de sécurité imposant une séparation physique minimale entre deux avions, cet espace aérien a été divisé dans ces zones en différents secteurs aériens, zones d'espace tridimensionnelles : à chaque secteur est affectée une équipe de contrôleurs aériens qui est responsable du respect de ces normes en temps réel : c'est le contrôle aérien (ATC pour Air Traffic Control). Le nombre d'avions qu'une équipe de contrôleurs peut gérer en même temps est évidemment limité et face à la forte demande de trafic aérien, un mécanisme est nécessaire pour éviter les surcharges de travail pour les contrôleurs et pour assurer une bonne utilisation des ressources de contrôle disponibles : ce mécanisme de régulation du trafic aérien est appelé ATFM (Air Traffic Flow Management). Les trois objectifs principaux de l'ATFM sont alors : prévenir les surcharges, lisser l'écoulement du trafic et minimiser les conséquences des actions de régulation notamment pour les compagnies aériennes.

#### I.1.b) Historique de la régulation aérienne en Europe

En Europe les premières tentatives de gestion des flux de trafic pour répondre à une saturation du système de contrôle du trafic aérien (ATC) ont été envisagées à la fin des années 60 et mises en place au début des années 70 (1972 en France, 1975 en Allemagne). Ces tentatives n'ont pas été coordonnées à un niveau supranational du fait de la séparation de la responsabilité de la gestion du trafic entre les différents pays. Bien que pouvant être efficaces au niveau local, ces premières mises en œuvre aggravèrent les conséquences des problèmes croissants de congestion au niveau européen, du fait d'un manque de communication et de coordination entre les différentes unités d'ATFM ([15]). Les bases d'un système européen global d'ATFM (en remplacement des différents systèmes nationaux) ont été posées lors du meeting ICAO (International Civil Aviation Organisation) consacré à l'espace aérien européen en 1980. La CFMU (Central Flow Management Unit) devait voir réellement le jour

en juillet 1989, sous la responsabilité naturelle d'Eurocontrol, l'organisation créée en 1963 afin de centraliser progressivement la gestion du trafic aérien en Europe assurée jusqu'ici par les différentes administrations nationales. Pour mémoire l'équivalent américain de la CFMU s'appelle Air Traffic Control System Command Center (ATCCC).

#### I.1.c) L'évolution de la régulation aérienne : de l'ATFM à l'ATFCM

Depuis avril 2004 et la publication du document « Air Traffic Flow and Capacity Management Strategy » ([5]), on assiste à un glissement terminologique de ATFM vers ATFCM pour désigner les activités de régulation assurées par la CFMU. Le C pour Capacity insiste ainsi sur le fait que les actions de régulation ne concerneront plus seulement la demande (le trafic) mais également l'offre (la capacité de contrôle), traduisant ainsi un rééquilibrage des objectifs de la CFMU en faveur des usagers des services de contrôle aérien.

#### I.1.d) La CFMU et les autres acteurs du trafic aérien

L'ATFM est donc un service de gestion globale du trafic aérien contribuant à assurer la circulation sûre, ordonnée et efficace des flux de trafic aérien. La CFMU fournit ce service de gestion des flux de trafic aux gestionnaires de l'espace aérien (Air Traffic Service ATS) comprenant notamment les centres de contrôles aérien et aux utilisateurs de cet espace aérien (Aircraft Operators AOs) dont font partie les compagnies aériennes. La CFMU cherche à optimiser l'utilisation du système aérien : à la fois celle de l'espace-temps aérien (Airspace Time) et celle des capacités du système (capacités des centres de contrôles et des aéroports). Envers l'ATS l'objectif est de lisser l'écoulement du trafic et de protéger les contrôleurs aériens des surcharges. Envers les AOs l'objectif est d'aider à la planification des vols et de minimiser les surcoûts liés à la congestion.

### I.2. Mise en œuvre opérationnelle

#### I.2.a) Les différentes phases de l'ATFM

La régulation est effectuée en plusieurs phases successives, aux moyens d'action et horizons d'action bien distincts (voir la Figure 1).

##### *i) L'ATFM stratégique*

De 18 mois à quelques jours avant le décollage. On analyse les prévisions de demande et de capacité, on cherche à anticiper les problèmes éventuels et à évaluer les solutions possibles pour y répondre. Pendant cette phase sont élaborés le plan de capacité (capacity plan) pour l'année à venir qui assigne à chaque centre de contrôle des objectifs de capacité, les schémas de routage (routing scheme), décrivant les routes à suivre en fonction de l'origine et de la destination et d'autres plans qui peuvent être activés dans les phases suivantes.

##### *ii) L'ATFM pré-tactique*

Quelques jours avant l'opération. On cherche à gérer au mieux l'offre disponible en fonction de la demande. Les mesures globales de gestion des flux (reroutement ou régulation des décollages) sont décidées pendant cette phase et rendues publiques sous forme d'un

ATFM Daily Plan (ADP) communiqué par ATFM Notification Message (ANM) aux compagnies aériennes et aux contrôleurs aériens.

iii) *L'ATFM tactique*

Le jour de l'opération. On actualise aux conditions réelles l'ADP et on assure son application à travers la gestion des reroutements et l'allocation des créneaux de décollage.

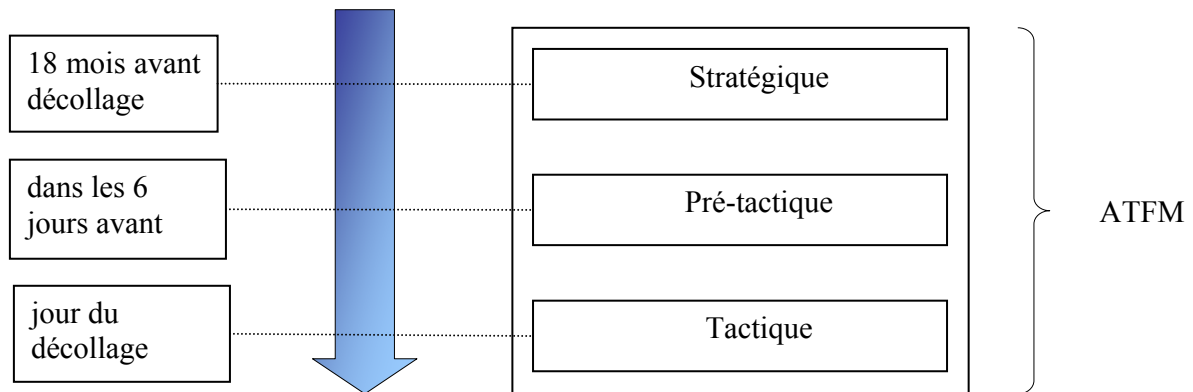


Figure 1) Organisation temporelle de l'ATFM

I.2.b) L'allocation des créneaux de décollage : principe général

Le principal outil de régulation utilisé par la CFMU est de retarder les avions au sol par le biais d'allocation de créneaux de décollage. Cette allocation est un processus complexe qui a été partiellement automatisé en Europe à l'aide du système CASA (Computer Assisted Slot Allocation). Tout d'abord la CFMU alloue un créneau de passage dans le secteur saturé et en déduit ensuite un créneau de décollage à partir des prévisions de temps de parcours. De nombreux travaux de recherche opérationnelle ont été consacrés à ce problème d'allocation de créneau qui présente une structure semblable aux problèmes combinatoires qui se posent dans le cadre de la régulation court terme : il est donc utile de décrire en détail le fonctionnement actuel de CASA.

La CFMU et les FMP (Flow Management Positions, annexes décentralisées de la CFMU qui servent notamment de relais auprès des centres de contrôle) s'accordent sur les zones où des mesures de régulation ATFM sont nécessaires et la CFMU active la régulation en spécifiant la zone, le flux d'entrée, les dates de début et de fin de la régulation etc. Le système CASA commence par sélectionner les vols concernés et calcule l'ordre virtuel d'arrivée dans la zone régulée en l'absence de régulation et calcule en fonction du principe « premier planifié, premier servi » (« first planned first served ») l'heure de décollage (CTOT : Calculated Take-Off Time) qui est transmise aux compagnies aériennes et à la tour de contrôle de l'aéroport de départ.

## I.2.c) Description détaillée de l'allocation de créneaux de décollage par CASA

### i) *Phase d'allocation (Allocation Stage)*

A partir de la durée de la régulation et du taux du flux d'entrée possible pour respecter la capacité, CASA fabrique une liste de créneaux (Slot Allocation List) qu'il va progressivement remplir au fur et à mesure qu'il reçoit les plans de vols. A chaque fois qu'un nouveau plan de vol arrive, CASA dispose de l'heure de départ supposée du parking : EOBT (Estimated Off-Block Time) : à partir du temps nécessaire pour aller du parking jusqu'à la piste d'envol (taxi time, dépend de l'aéroport) on calcule la date supposée de décollage ETOT (Estimated Take-off Time) puis la date de passage au dessus de la zone régulée ETO (Estimated Time Over). On détermine ainsi le créneau le plus proche du temps de passage sans régulation. Si ce créneau est libre il est provisoirement attribué à ce vol, s'il est déjà occupé CASA attribue le créneau au vol qui serait passé le premier sans régulation. Il y a ensuite une réaction en chaîne jusqu'à ce que tous les vols connus aient un créneau alloué. Cette phase est appelée phase de pré-allocation (pre-allocation stage) et est interne à CASA. Pour chaque vol pré-alloué on a alors un créneau de passage au dessus de la zone régulée (CTO Calculated Time Over) ce qui entraîne un créneau de décollage (Calculated Take-off Time). A chaque fois qu'un vol est annulé, la procédure de correction des créneaux s'applique aux vols qui avaient un créneau pré-alloué afin d'utiliser les nouveaux créneaux disponibles : c'est la procédure de correction des créneaux (Slot Amendment Procedure). A une période de temps fixée (c'est un paramètre appelé Slot Issue Time SIT) avant l'EOBT, le CTOT est officiellement alloué au vol et un message d'allocation (Slot Allocation Message, SAM) est envoyé aux compagnies et aux contrôleurs aériens. Cette phase est appelée phase d'allocation (Allocation Stage). Si un vol est soumis à plusieurs régulations, le CTOT de la régulation la plus pénalisante s'applique au vol.

### ii) *Améliorations ETFMS*

Le programme Enhanced Tactical Flow Management System (ETFMS, gestion tactique améliorée des flots de trafic) améliore l'allocation des créneaux en prenant en compte des données réelles et actualisées sur les positions des vols en l'air. A partir de données radar (Correlated Position Reports) CPR's et des données météo (vents...) la trajectoire des vols est recalculée afin de prévoir plus précisément les dates de passage des vols dans la zone régulée. Pour les vols déjà en l'air les dates de passage réelles (Actual Time Over, ATO) sont alors actualisées et elles remplacent dans la SAL les dates de passages calculées (CTO). L'allocation des créneaux de décollage se fait alors en prenant en compte ces nouvelles données. Si pour un vol donné de nouvelles informations météo arrivent avant le SIT, CASA recalcule les temps de passage et alloue le créneau de décollage en fonction mais si elles arrivent après le SIT les changements sont alors limités aux améliorations.

### iii) *Mécanismes d'ajustements*

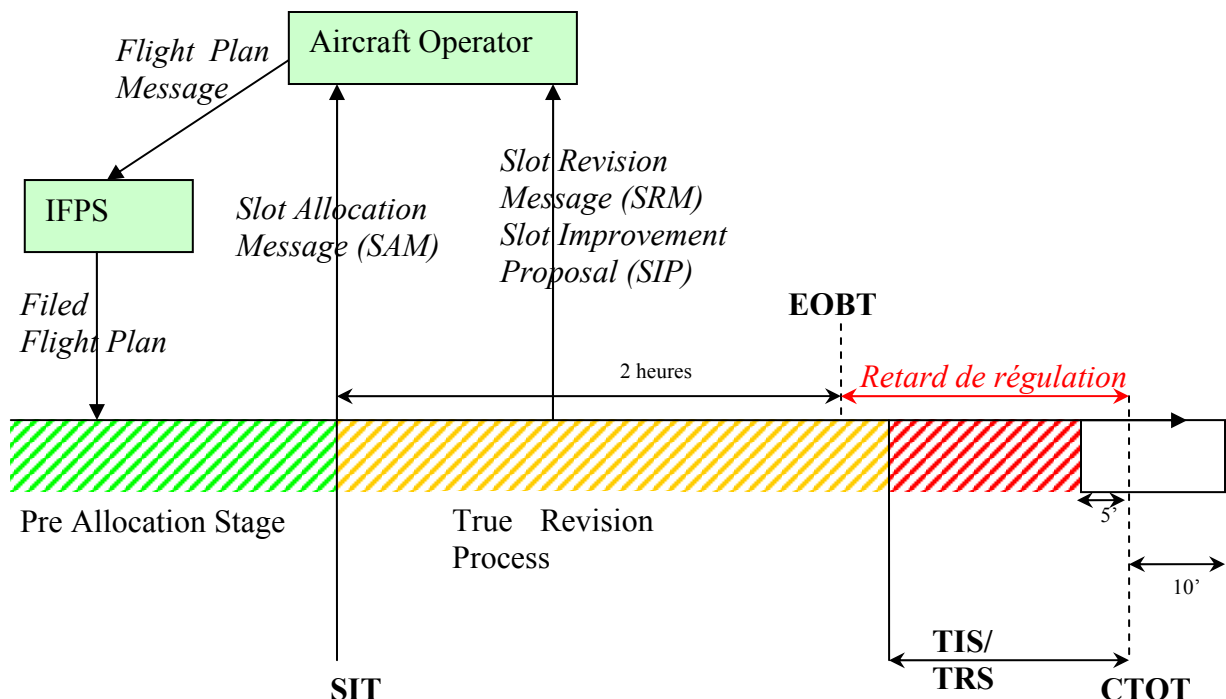
La dernière phase est appelée True Revision Process : jusqu'à un temps donné (qui dépend de l'aéroport) avant le CTOT le créneau peut être amélioré pour prendre en compte les nouvelles données.

En plus de ce principe fondamental de fonctionnement de CASA d'autres mécanismes d'ajustements (réservation d'intervalles...) existent afin de prendre en compte certaines particularités, notamment pour rééquilibrer le traitement des plans de vol reçus tard. En effet étant donné que les créneaux sont fixés deux heures avant le décollage prévu, les vols courte distance peuvent soumettre leur plans de vols alors qu'une majorité de la capacité a déjà été

définitivement attribuée à des vols longue distance. Pour préserver l'équité entre les différents types de vols une nouvelle notion a donc été introduite dans CASA : la bande de temps (time band) : la capacité est divisée en différentes bandes de temps spécifiquement réservées à des types de vols donnés. Par exemple on peut diviser la capacité en trois bandes TB1 pour les courtes distances, TB2 pour les moyennes et TB3 pour les longues distances. Les vols de la TB1 ont accès à toutes les bandes, ceux de la TB2 aux bandes TB2 et TB3 et ceux de la TB3 uniquement à la TB3. Ce mécanisme permet d'équilibrer le traitement des différents types de vols (pas seulement caractérisés par la distance comme dans cet exemple).

Dans la réalité il y aura toujours des vols connus tardivement par le système. Afin d'éviter un retard prohibitif on réserve un petit pourcentage de capacité aux vols de dernière minute : ce taux appelé « pending rate » (taux en attente) est une réserve de capacité non utilisée en temps normal.

Enfin il peut exister des cas où on autorise un taux supplémentaire négocié avec les centres d'ATC. Les différentes actions en fonction du temps sont décrites dans la Figure 2 et un exemple d'allocation est détaillé dans la Figure 3.



- EOBT**: Estimated Off Block Time, temps de départ du parking estimé dans le plan de vol
- CTOT** : Calculated Take Off Time, date de départ calculée par CASA
- SIT** : Slot Issue Time, date à laquelle on informe la compagnie du créneau alloué
- TIS** : Time to Insert into the Sequence, temps minimal pour prévenir d'une amélioration sur le créneau de départ (dépend de l'aéroport)
- TRS** : Time to Remove from the Sequence, temps minimal pour prévenir d'un retard sur le créneau de départ (dépend de l'aéroport) TIS et TRS peuvent être distincts

Figure 2) Actions en fonction de la date lors de l'allocation de créneaux par le mécanisme CASA



Fonctionnement de CASA

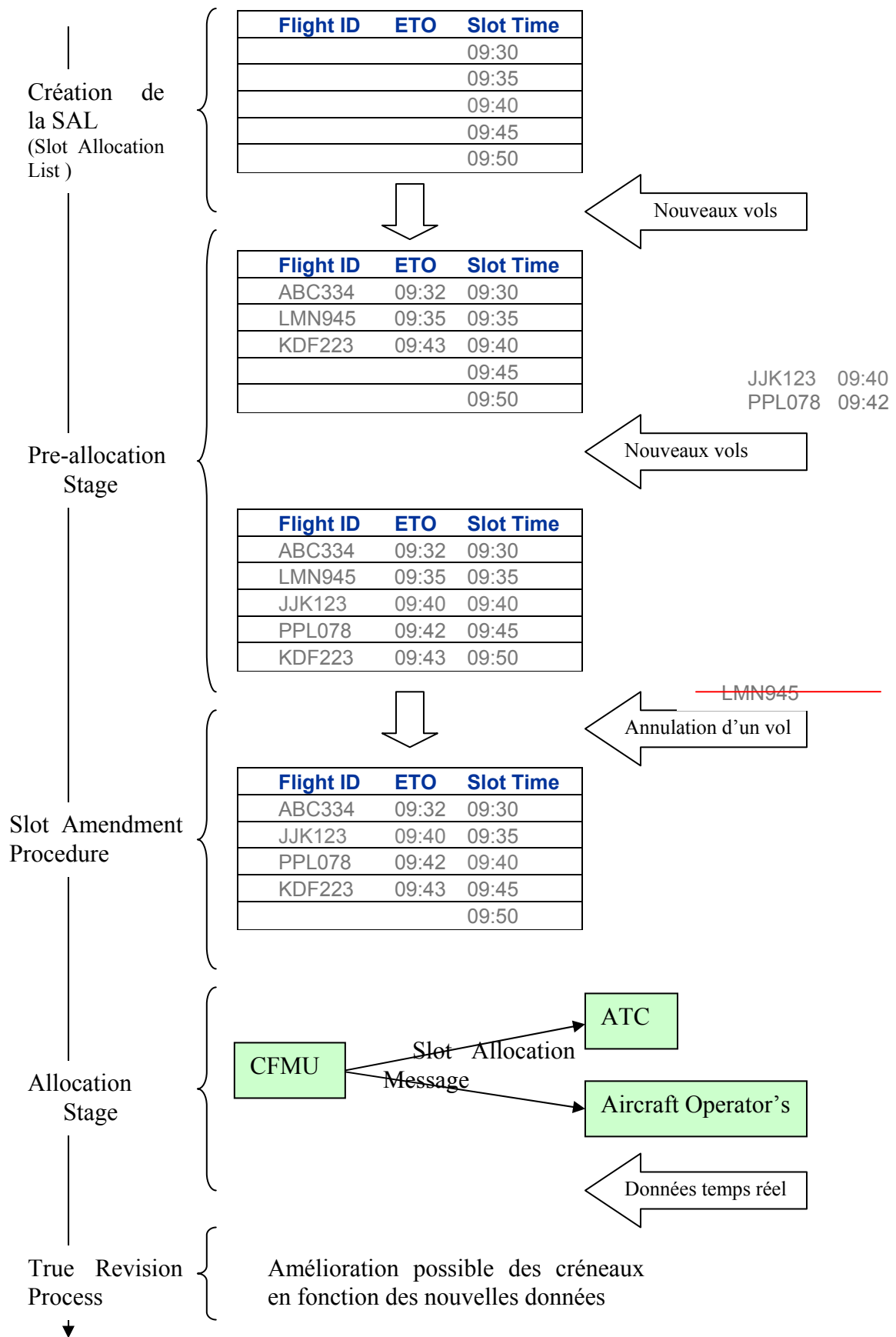


Figure 3) Exemple d'allocation de créneaux par CASA

## **II. Limites de l'organisation actuelle et réponses de la régulation court terme.**

*« Les régulations ATFM au sol devraient être appliquées principalement lorsque la demande est bien au dessus de la capacité (10% ou plus). L'application de régulations ATFM pour toutes autres raisons ne permet pas de juguler les surcharges et génère au contraire des retards superflus »*

(Rapport d'examen des performances Eurocontrol pour 2004 [4])

### **II.1. Les faiblesses indépassables de la régulation actuelle**

#### **II.1.a) Introduction**

Les différentes données chiffrées sur les retards et leurs conséquences sont tirées des rapports d'examen des performances Eurocontrol<sup>1</sup> 2005 et 2006 pour les années 2004 et 2005 ([2] et [4]) ce qui montre l'actualité des problèmes soulignés.

Comme énoncé précédemment l'ATFM cherche à assurer un écoulement sûr du trafic aérien sur le réseau avec trois objectifs principaux : protection des contrôleurs contre la surcharge, lissage du trafic et minimisation des conséquences de la congestion pour les compagnies aériennes. Les retards au sol constituent l'essentiel des actions de régulation utilisées par la CFMU. Le système actuel a su faire preuve d'efficacité en diminuant le retard moyen par vol dû aux régulations destinées à respecter les capacités des secteurs en route de 5,5 minutes en 1999 à 1,3 minutes en 2005 malgré un trafic reparti à la hausse. Ainsi lorsque la demande dépasse fortement la capacité, la politique de retard au sol est efficace. On peut cependant noter que la capacité effective des secteurs de contrôle a connu une augmentation supérieure à celle du trafic aérien, ce qui explique en partie cette diminution des retards. Cette augmentation de la capacité est notamment due à des améliorations de productivité des centres de contrôle, facilitées par des changements techniques (ETFMS<sup>2</sup>, niveaux RVSM<sup>3</sup>, fréquences 8.33kHz<sup>4</sup>). Il est difficile de séparer, d'une manière générale, les différentes causes d'amélioration. Il reste que la politique de retard au sol se heurte à des difficultés inhérentes qui ne pourront pas être résolues par des gains de productivité ou par augmentation de la capacité mais qui nécessitent une évolution du mode de régulation :

- l'incertitude inhérente au système aérien est une limitation indépassable de la politique de retards au sol
- le lissage du trafic aux aéroports d'arrivée pour lequel les retards au sol sont inopérants
- la gestion des situations dynamiques d'exception (météo, défaillances techniques...) est peu compatible avec une régulation trop en amont

---

<sup>1</sup> La Commission d'Examen des Performances (PRC), organisme indépendant chargé d'analyser les performances des services de navigation aérienne en Europe pour le compte d'Eurocontrol publie chaque année un rapport couvrant tous les aspects ATM (sécurité, retards, efficacité économique et efficacité des vols) de l'année écoulée en Europe.

<sup>2</sup> ETFMS Enhanced Tactical Flow Management System, voir paragraphe I.2.c)

<sup>3</sup> RVSM : Reduced Vertical Separation Minimum : diminution de la séparation verticale minimale, se traduit par l'ajout de nouveaux niveaux de vol, mis en place en Europe depuis 2002

<sup>4</sup> Nouvelles bandes de fréquences destinées à lutter contre l'encombrement des fréquences aériennes

Ainsi, ces limitations expliquent que le retard global dû aux régulations ATFM subisse une légère augmentation depuis 2003 (de 1,8 minute par vol en 2003 à 2.2 minutes en 2005) et on peut penser que le système actuel de régulation a atteint ses limites et ne pourra pas faire face en l'état à l'augmentation prévue du trafic pour les années à venir (4% par an en moyenne). Nous allons décrire ces limitations et expliquer en quoi la régulation court terme leur apporte une réponse.

#### II.1.b) L'incertitude sur le système aérien

Le système aérien est un système multi-acteurs extrêmement complexe comprenant des facteurs techniques, humains et naturels tous porteurs d'incertitude. Ainsi la vitesse et la direction du vent, la charge de l'avion, son état technique, les éventuelles mesures d'évitement en vol commandées par les contrôleurs aériens vont fortement peser sur la prévision de marche des avions. La précision des prévisions d'heure de passage dans des secteurs ou d'arrivée à l'aéroport final va donc évidemment diminuer avec l'augmentation de l'horizon de prévision. Il existe cependant un effet de seuil significatif pour les prévisions avant le décollage. En effet la taille des créneaux de décollage alloués (créneau ATFM de 15 minutes), la tolérance existante sur le respect de ces heures de décollage, l'incertitude sur les temps de roulage vont entraîner que la marge d'incertitude sur une prévision effectuée avant le décollage d'une heure de passage d'un avion dans un secteur va, en de nombreux cas, dépasser la durée de survol de ce secteur. Etant donné que la régulation actuelle est basée sur des prévisions effectuées quelques heures avant la saturation prévue, on comprend la limite indépassable de la régulation basée sur les retards au sol. La commission d'examen des performances d'Eurocontrol l'écrit explicitement dans son rapport 2005 ([4]) « *Les régulations ATFM ne peuvent pas précisément contrôler les flux à travers les secteurs où le temps de transit est généralement plus court que le créneau ATFM lui-même* ». L'ajout d'un nouveau filtre de régulation agissant notamment sur les avions en vol, avec un horizon de prévision très inférieur permettrait de prendre des décisions avec une incertitude significativement réduite.

#### II.1.c) Lissage du trafic en entrée d'aéroports

Les retards ATFM aéroportuaires (dus à une régulation liée à un aéroport) représentent 49 % de tous les retards ATFM après une augmentation de 8 % en 2004. De plus les attentes en l'air au niveau des aéroports sont lourdes de conséquences en termes économiques, environnementaux et sociaux (les aéroports étant situés à proximité des zones d'habitation). Enfin, la qualité du compromis entre l'utilisation optimale de la capacité des aéroports et l'attente en l'air à l'arrivée dépend de la fluidité du trafic arrivant : plus le trafic est fluide et meilleur est ce compromis. Or les régulations ATFM effectuées plusieurs heures avant l'arrivée avec une incertitude élevée (créneau de 15mn, non respecté dans 20 % des cas), si elles peuvent protéger contre une grosse surcharge, ne peuvent en aucun cas assurer le lissage nécessaire des flux d'arrivée aux aéroports. Ceci ne peut être fait que par un contrôle des vitesses fin sur les avions en vol, ce qui rentre dans le cadre de la régulation dynamique du trafic.

#### II.1.d) Gestion des situations d'exception

En cas de mauvaises conditions météorologiques ou de défaillance technique ou humaine la capacité des secteurs et/ou des aéroports peut varier drastiquement (et même être tout simplement réduite à zéro) et devient fortement dynamique. Les perturbations engendrées par

ce type de situations sont très pénalisantes à la fois pour les usagers et pour les services aériens en situation de stress et sont mal amorties par le système actuel d'ATFM, une nouvelle fois à cause de sa nature même qui consiste à prendre des actions de régulations plusieurs heures avant les événements que l'on souhaite modifier ou éviter. Une telle approche ne permet pas de gérer efficacement des situations où la capacité peut beaucoup varier sur une petite échelle de temps et où on n'a pas de prévision suffisamment fiable et précise pour estimer la capacité dans le futur (mauvaises conditions météorologiques par exemple). Les situations d'exception ne peuvent être gérées de manière optimale (ou s'en approchant) que par une régulation à un horizon de temps réduit par rapport aux horizons de temps actuels.

Toutes ces raisons montrent les limites intrinsèques de la régulation actuelle et plaident pour une approche complémentaire de la régulation existante.

## II.2. Les principes de la régulation court terme

### II.2.a) Pourquoi la régulation court terme ?

Comme on vient de l'expliquer, les obstacles à la régulation classique du trafic aérien basée sur les retards au sol restent nombreux. Ils compromettent la possibilité de continuer la baisse des retards ATFM observés ces dernières années (qui peut également s'expliquer par une augmentation de la capacité). On ne peut même exclure un retour à la hausse de ces retards face à la croissance prévue du trafic dans les années à venir (trafic en augmentation de 3.9% en Europe en 2005<sup>5</sup> et augmentation similaire prévue pour les années à venir : voir Figure 4). Tous ces obstacles ont le point commun d'être par nature irréductibles face à une approche basée exclusivement sur une régulation largement en amont agissant sur des vols avant leur décollage. Les deux grands objectifs de l'ATFM que sont la protection contre les surcharges et la minimisation des conséquences de la congestion pour les compagnies aériennes ne sont pas complètement satisfaits comme le souligne la commission d'examen des performances : « *les proportions de vols en surnombre et de retards ATFM inappropriés restent élevées* » ([4]). Ainsi en 2005 le trafic réel dépassait la capacité déclarée du secteur concerné de plus de 10% pour 10.7% du total des heures régulées (9.4 % en 2004) et 7.1% des retards ATFM affectés l'ont été à tort (6.9 % en 2004).

Cette situation de violation courante de la capacité déclarée des centres de contrôle aérien peut conduire ceux-ci à minorer en amont la capacité qu'ils déclarent pour se protéger vraiment des surcharges avec pour conséquence une baisse globale de la capacité du système. De même les compagnies vont introduire des marges programmées dans leur planning pour prendre en compte les retards prévisionnels ce qui se traduit par des coûts importants. L'ajout d'un filtre de régulation pouvant agir sur des avions déjà en vol avec un horizon de prévision beaucoup plus court (de l'ordre de 30 mn par exemple) permettrait de prendre des décisions basées sur des prévisions beaucoup plus fiables, avec une marge d'incertitude grandement réduite, de gérer beaucoup plus finement le lissage du trafic et le respect des capacités à la fois des secteurs en-route et des aéroports, rendant la confiance aux contrôleurs aériens et évitant la prise de marge en amont dans la définition de la capacité. Un tel filtre serait également un outil précieux dans les situations d'exceptions hautement dynamiques telles que mauvaises conditions météorologiques ou défaillances techniques ou humaines.

---

<sup>5</sup> En 2006, le nombre de vols régulés a augmenté de 3,9% sur les 9 premiers mois de l'année par rapport à la même période de 2005 (statistiques Eurocontrol [70])

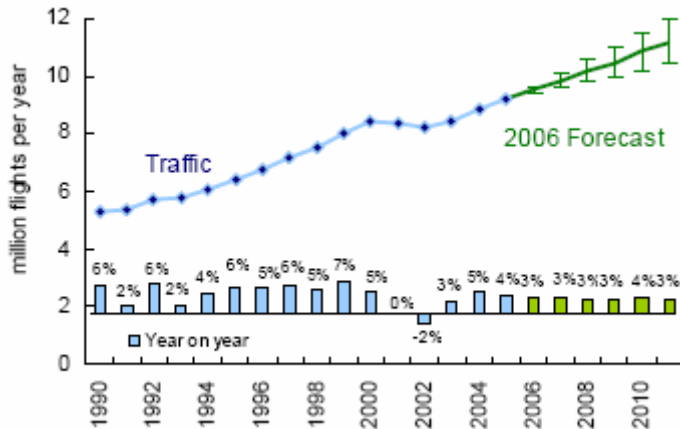


Figure 4) Evolution du trafic aérien en Europe et prévisions pour les années à venir ([2])

## II.2.b) Principe général de la régulation court terme

Malgré les limites que nous venons de décrire, le système actuel de régulation a fait preuve ces dernières années de son efficacité et de sa capacité à gérer rationnellement cette ressource rare qu'est devenu le contrôle aérien. L'approche proposée dans ce travail ne vise pas à remplacer le système actuel mais au contraire à en développer une évolution naturelle, complémentaire de l'existant et permettant de dépasser les problèmes apparus et d'appréhender l'augmentation du trafic. L'avantage principal de cette approche est qu'elle permet d'envisager une application assez rapide puisqu'elle n'implique pas une remise en cause totale de l'organisation actuelle de l'ATFM et de l'ATC. La méthode de régulation proposée consisterait donc en un nouveau filtre agissant notamment sur les avions en vol (voir Figure 5). Les outils de régulation existants seraient alors principalement utilisés lorsque la demande excède sensiblement la capacité, comme préconisé par la commission d'examen des performances dans ses deux derniers rapports.

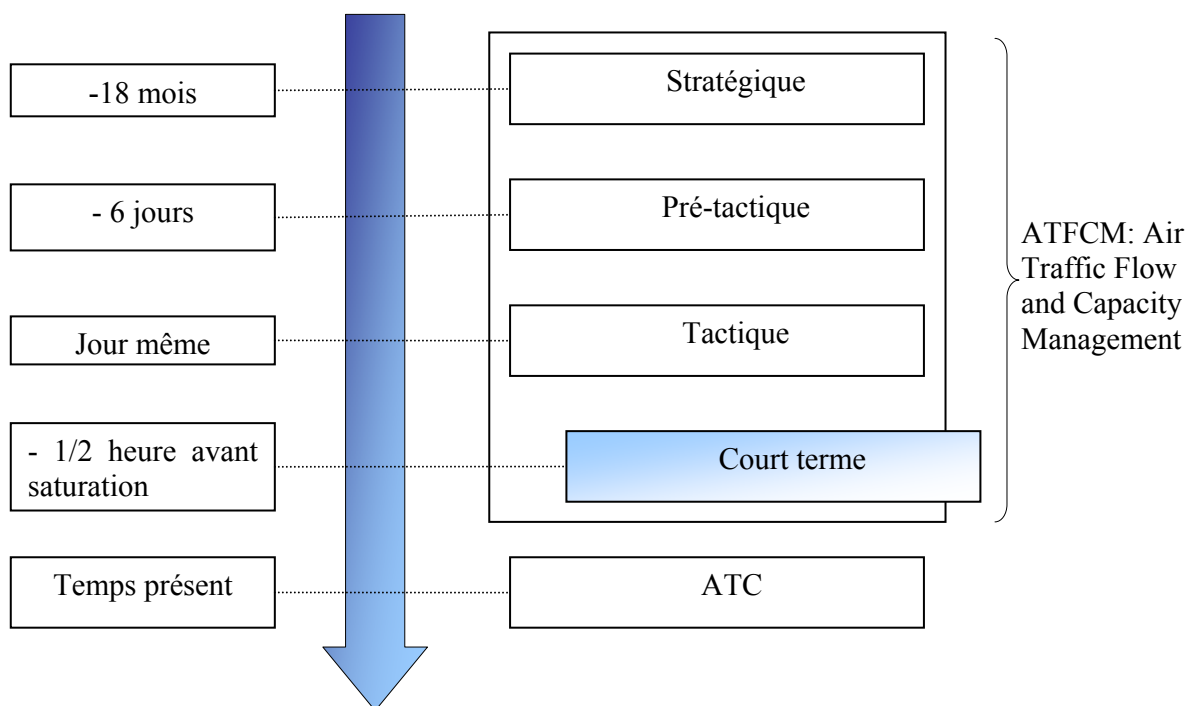


Figure 5) Intégration de la régulation court terme dans le processus d'ATFM

## II.2.c) Les différentes approches actuelles de la régulation court terme

### ▪ L'approche de la CFMU

L'évolution souhaitée à court et moyen terme de la pratique de la régulation aérienne est notamment exposée dans le document « ATFCM Strategy » ([5]). Les principales tendances sont les suivantes : gestion combinée de la demande (vols) et de l'offre (capacité de contrôle) afin que la régulation aérienne ne se résume pas à adapter la demande à l'offre, approche plus concertée avec les compagnies aériennes tant dans le partage d'informations que dans la prise de décision<sup>6</sup>, une plus grande adaptabilité de la régulation stratégique, une amélioration de l'optimisation de l'utilisation de la capacité de contrôle et une modification des horizons d'action pour ne pas limiter la régulation aérienne à la phase avant le décollage. Ce dernier point explicitement mentionné dans le rapport (voir Figure 6) montre que le diagnostic de la nécessité de la régulation court terme est partagé par tous les acteurs du système aérien. L'approche FAM-CAMES décrite ci-dessous en est une première approche.

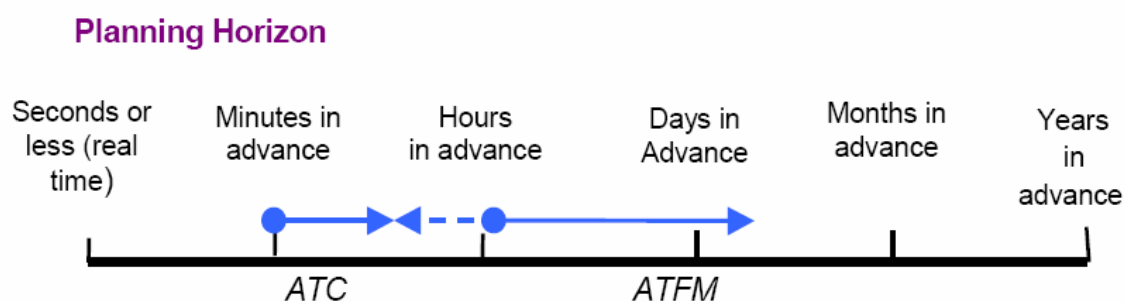


Figure 6) Horizon de planification ATFM tiré du rapport ATFCM Strategy [5] p. 7

### ▪ L'approche FAM-CAMES

Eurocontrol travaille actuellement sur un projet nommé CAMES (Co-operative ATM Measures for a European Single Sky), application opérationnelle du concept FAM (Future ATFM Measure), une approche basée sur la « synchronisation du trafic aérien en temps réel », c'est-à-dire une régulation dynamique du trafic. Le but de ce projet est de spécifier, développer et tester de nouvelles méthodes de travail, mesures de régulation et outils d'aide à la décision. Cette approche se situe cependant au niveau pré opérationnel et est plus axée vers la faisabilité organisationnelle de la régulation court terme alors que l'approche du LICIT est d'étudier les principaux aspects d'une régulation dynamique, sans s'intéresser aux détails de la mise en œuvre opérationnelle des systèmes ou actions proposées. La description de la régulation court terme dans la partie suivante correspond aux recherches du LICIT sur ce sujet.

<sup>6</sup> Ce concept n'est pas spécifique à l'ATFM et représente une tendance lourde de l'organisation du système aérien : il est généralement nommé CDM (Collaborative Decision Making) et certaines applications sont déjà en place aux Etats-Unis notamment. On pourra consulter le site Internet dédié aux applications européennes [70].

### **III. Principes généraux du régulateur court terme du trafic aérien**

#### III.1. Préliminaires

##### III.1.a) Introduction

La connaissance du contexte du régulateur court terme du trafic aérien est indispensable à la compréhension des problèmes combinatoires étudiés dans cette thèse et à ce titre, la présente partie s'attache à décrire précisément les objectifs et le fonctionnement de ce filtre. Ce mémoire étant cependant focalisé sur le module d'optimisation du régulateur court terme, le lecteur intéressé trouvera plus de détails sur le régulateur court terme dans le rapport de recherche établi par le LICIT ([3]) dans le cadre de la convention entre Eurocontrol et l'INRETS.

##### III.1.b) Définitions

*Vol* : C'est la donnée de l'ensemble des informations suivantes :

- identifiant,
- aéroport de départ,
- heure de départ prévue, heure de départ réelle,
- heure d'arrivée prévue
- aéroport de destination,
- type d'appareil, qui permet de déterminer les caractéristiques au sens large des trajectoires 4D réalisables par le vol (niveaux de vol atteignables, gamme des vitesses et taux de montée/descente possibles) et la consommation de carburant dans les différentes trajectoires envisageables,
- trajectoire 4D prévue à partir de l'instant courant (voir plus loin)
- état (réserve de carburant, pénalisations subies depuis le décollage)

*Trajectoire* d'un vol : ensemble de points 4D (définis en latitude, longitude, niveau de vol, heure de passage) survolés par le vol. Dans certains cas on ne considèrera que la *trajectoire 3D* d'un vol, sans la donnée temporelle.

*Trajectoire réalisable* : trajectoire respectant une liste de contraintes techniques et réglementaires.

*Route* d'un vol : ensemble de balises aériennes survolées par le vol

*Secteur de contrôle* ou *secteur* : zone de l'espace aérien définie en trois dimensions à l'intérieur de laquelle les contrôleurs aériens rendent un service d'anticollision.

*Charge secteur* : nombre d'avions entrant dans un secteur pendant une période de temps fixe (habituellement une heure) appelée *période de définition de la capacité*.

*Capacité secteur* : valeur que la charge secteur ne doit pas dépasser, fixée par les centres de contrôle.

*Période de définition de la capacité* : voir *charge secteur*

*Secteur-période* : secteur considéré pendant une période de définition de la capacité.

*Secteur saturé* : secteur dont la charge dépasse la capacité pendant une période de définition de la capacité.

*Saturation* : couple (secteur saturé, pas de définition de la capacité).

*Actions de régulation* : ce sont les actions susceptibles de modifier la trajectoire 4D d'un vol afin d'atteindre les objectifs de la régulation.

*Vol régularisable* : vol dont il est possible dans un contexte opérationnel de modifier la trajectoire dans un but de régulation. Les vols non régularisables sont les vols longue distance, les vols exemptés (sanitaires, recherche et sauvetage, gouvernements), les vols à cours de carburant, les vols ayant subi une pénalisation trop importante depuis leur mise en route, les vols appartenant à des compagnies ayant subi globalement trop de pénalisations dans une période donnée. Les vols non régularisables sont donc dispensés d'action de régulation.

*Vol commandable* : vol régularisable sur lequel on s'autorise pour des raisons méthodologiques à appliquer des actions de régulation. C'est une notion technique : on peut choisir par exemple que les vols commandables sont exclusivement les vols régularisables qui traversent une saturation ou inversement que les vols commandables sont tous les vols régularisables. Le caractère commandable ou non d'un vol peut évoluer au cours du temps et dépendre du module technique qui l'envisage.

## III.2. Cahier des charges de la régulation court terme.

### III.2.a) Objectifs de la régulation court terme

D'une manière générale les objectifs de la régulation court terme sont inclus dans les objectifs actuels de l'ATFM puisqu'elle en est une extension : faire respecter les capacités déclarées (qui peuvent varier dynamiquement) des secteurs et des aéroports, lisser la charge pour assurer un écoulement harmonieux du trafic dans les secteurs et aux aéroports d'arrivée et minimiser les conséquences de la congestion ou des situations d'exception pour l'ensemble des usagers de l'espace aérien. Les travaux menés par le LICIT ne considèrent pas explicitement la possibilité de faire varier la capacité pour l'adapter à la demande, étant donné que les actions d'augmentation de la capacité seront plutôt effectuées plus en amont dans le processus global de régulation mais un objectif additionnel que l'on peut s'assigner dans cet esprit est d'identifier les secteurs-périodes critiques pour le système aérien.

Dans le cadre des objectifs futurs de l'ATFCM tels qu'exposés dans *ATFCM Strategy* ([5]), il faut inclure l'optimisation des trajectoires en proposant aux compagnies des routes plus directes ou des niveaux de vol plus adaptés lorsque c'est possible afin de minimiser la consommation de carburant et/ou le temps perdu et d'améliorer d'une manière générale l'efficacité des vols. Il est à noter que ce changement de finalité de la régulation doit s'effectuer en concertation avec les compagnies aériennes qui en sont les bénéficiaires et l'application opérationnelle devra prendre en compte les concepts CDM (voir note 6 p.19). D'un point de vue théorique cependant, l'ajout de ce nouvel objectif ne change pas fondamentalement le problème puisque le calcul des actions de régulation afin d'éviter les saturations comporte déjà une fonction objectif cherchant à minimiser les conséquences de la régulation.



### III.2.b) Actions de régulation envisagées

#### i) *Les actions spatiales*

Pour prévenir les saturations, une première possibilité consiste à modifier la trajectoire spatiale des vols, de manière à leur faire éviter les secteurs les plus chargés. Etant donné que les secteurs sont des polyèdres comportant un niveau de vol minimal et un niveau de vol maximal, les actions permettant d'éviter un secteur sont le changement de niveau de vol qui pourra permettre de passer au dessus ou en dessous d'un secteur saturé et les reroutements qui permettent d'éviter un secteur en suivant une autre route aérienne. Pour le reroutement on peut envisager de prévoir une réserve de carburant supplémentaire destinée à la régulation court terme pour les vols planifiés prévus pendant une période chargée et susceptibles d'être soumis à des régulations, mais étant donné le coût d'embarquement de carburant pour les compagnies, la taille de cette réserve sera à étudier soigneusement.

#### ii) *Les actions temporelles*

Le problème que l'on cherche à résoudre est hautement dynamique : les saturations concernent des secteurs-périodes, les temps de parcours dépendent du temps, certains itinéraires sont ouverts ou fermés en fonction de la date ou du moment de la journée. On souhaite donc apporter une réponse temporelle, c'est-à-dire proposer des solutions qui n'utilisent pas seulement les alternatives spatiales (reroutement, changement de niveau) mais aussi les alternatives temporelles. Les actions de régulation permettant de jouer sur la dimension temporelle d'une trajectoire sont l'attente en l'air ou au sol pour les avions n'ayant pas encore décollé et la modulation de la vitesse de croisière entre deux vitesses minimale et maximale admissibles. La réserve de carburant de régulation évoquée ci-dessus pourra donc également être utilisée pour l'attente en l'air. L'attente en l'air est normalisée aux aéroports par des boucles au dessus d'une balise prédéfinie et ayant généralement la forme d'un hippodrome. Il existe également des points d'attente définis en fonction de critères opérationnels et environnementaux. Généralement la procédure d'attente au dessus de ces points est également codifiée.

### III.2.c) Contraintes à respecter

Le système que l'on se propose de mettre en place doit respecter les contraintes suivantes.

#### i) *Contraintes sur les trajectoires*

Le système doit être capable de répondre à des situations dynamiques et doit donc être un processus dynamique. Les actions de régulation doivent respecter les règles usuelles de la navigation aérienne : ainsi les reroutements doivent suivre les routes aériennes existantes disponibles, les changements et les choix de niveau de vols doivent être conformes aux séries de niveaux de vol habituelles<sup>7</sup>. Toutes les actions doivent être compatibles avec une utilisation

---

<sup>7</sup> Afin d'assurer la séparation verticale des vols, des niveaux de vol obligatoires sont définis par l'altitude exprimée en centaine de pieds au dessus d'une surface isobare de référence (1013.25 hPa) et sont nommés en fonction de leur altitude : le niveau de vol 60 correspond ainsi à une altitude de 6 000 pieds. Les vols IFR (vol aux instruments) utilisent les niveaux de vol se terminant par un 0 tandis que les vols VFR (vol à vue) utilisent ceux se terminant par un 5. Enfin, pour les routes bidirectionnelles les niveaux impairs (110, 130 ...) sont utilisés par les aéronefs suivant une route magnétique entre 0° et 179° et les niveaux pairs (120, 140 ...) par ceux suivant une route magnétique entre 180° et 359°.

normale des performances des avions en prenant en compte toutes les informations disponibles (charge, type d'appareil, prévisions météorologiques...), notamment les changements de niveau de vol et la modulation de vitesse. Le confort des passagers doit être respecté en instaurant des bornes d'accélération, une limitation du nombre de changements de niveau et un angle maximal de changement de trajectoire horizontale.

ii) *Limitation du nombre d'actions de régulation*

Le nombre d'actions de régulation imposées à un vol sera limité. Ces contraintes ont pour but de limiter l'inconfort pour l'équipage et les passagers occasionné par des manœuvres trop fréquentes de changement d'altitude, d'inclinaison pour virage, d'accélération et de décélération, même si l'amplitude de celles-ci est déjà limitée. Pour prendre en compte ces contraintes de limitation du nombre d'actions de régulation, il n'est pas nécessaire de les inclure dans le problème d'optimisation. Il suffit de garder un historique par vol de toutes les actions de régulation effectuées depuis le décollage pour déterminer les vols qui seront ultérieurement dispensés de régulation, c'est-à-dire déclarés non régulables.

iii) *Contraintes d'équité*

Une autre contrainte importante pour le système est celle de l'équité de traitement entre les différentes compagnies aériennes. Les compagnies aériennes doivent subir, du fait de la régulation, des niveaux de pénalisation identiques en rapport avec le nombre d'heures de vol qu'elles opèrent. Le même souci d'équité peut exister pour les aéroports ou pour les couples (origine-destination). Cependant ces contraintes d'équité ne concernent pas exclusivement la régulation court terme et pourront être considérées à un niveau global sur une période longue de régulation (par mois voire par année) par l'existence de mécanismes de compensation : les compagnies ayant un taux de pénalisation plus faible que la moyenne abondent par exemple un fond permettant d'indemniser celles ayant un taux plus fort que la moyenne. Les contraintes d'équité ne seront donc pas introduites explicitement à l'échelle de régulation court terme.

III.2.d) Coûts à prendre en compte

La politique de régulation choisie doit atteindre les objectifs énoncés ci-dessus à moindre coût pour les utilisateurs du système aérien : les opérateurs de vols. Pour cela la comparaison de deux politiques doit s'effectuer par rapport à leur coût global pour l'ensemble des vols concernés. Cela inclut le coût global des retards infligés aux vols (heures supplémentaires du personnel, perte d'image, coût des retards des passagers) et des surconsommations de carburant dues aux attentes en l'air, aux reroutements et aux changements de niveau de vol. De plus il faut prendre en compte les coûts liés aux retards réactifs, définis ci-dessous.

i) *Les retards réactifs*

Le système aérien est un système hautement interconnecté et les retards assignés à un vol se propagent durant le reste de la journée aux autres vols. Ces retards sont appelés retards réactifs et viennent du fait que plusieurs vols sont prévus par jour pour chaque équipage et chaque avion. En 2004 les retards réactifs représentaient la plus grande part des retards totaux avec plus de 45% des retards de plus de 15 minutes ([4]). Cette situation est notamment causée et aggravée par le système de fonctionnement des compagnies en « hub » : les compagnies disposent généralement d'une plate forme centrale à laquelle arrivent des vols provenant des autres aéroports desservis par la compagnie et de laquelle ils repartent chacun

vers une destination différente. Les arrivées et les départs s'effectuent sur une plage de temps très courte afin de minimiser les temps d'attente. Ce processus peut être répété plusieurs fois par jour. Ce système permet d'assurer une liaison entre un grand nombre de paires origine-destination tout en limitant le nombre de lignes et d'avions nécessaires. Un des principaux inconvénients de ce mode d'organisation est sa sensibilité aux retards qui se propagent à de plus nombreux vols. Le système devra prendre en compte d'une manière fine les conséquences des retards alloués aux vols.

### III.3. Architecture globale du régulateur court terme

#### III.3.a) Le cadre général de la commande prédictive à horizon glissant

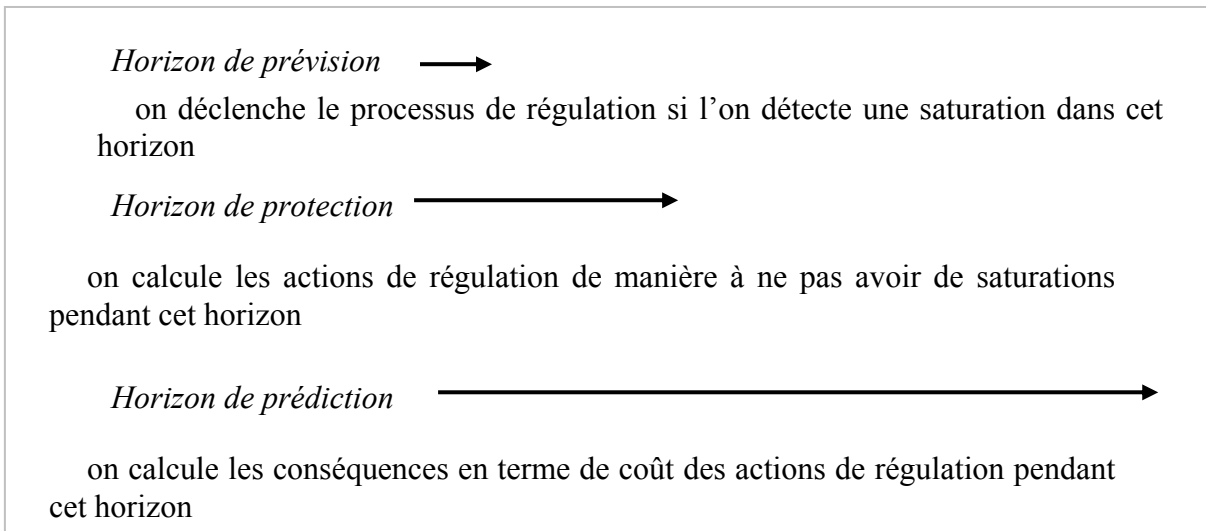
La principale raison d'être du régulateur court terme est le contexte hautement dynamique du système aérien. Les conséquences des décisions prises à un instant donné doivent donc être réévaluées continuellement et le cas échéant, des modifications ou de nouvelles actions doivent pouvoir être décidées. Pour ces raisons le processus de régulation envisagé est basé sur la commande prédictive à horizon glissant. Cette technique de commande consiste à calculer à un instant donné les prévisions de valeur des variables caractérisant notre système pour un horizon de temps appelé *horizon de prédiction*. On calcule alors une politique à appliquer aux variables sur lesquelles on peut agir, politique à appliquer sur un horizon de temps inférieur à l'horizon de prédiction et appelé parfois *horizon de commande*. Enfin on applique au système les éléments de la politique calculée correspondant à la première période suivant l'instant présent. A l'issue de cette période, la politique à suivre est réévaluée en fonction des nouvelles données disponibles, les deux horizons ayant « glissé » d'une période. La durée d'une période et la longueur des horizons de prédiction et de commande sont les principaux paramètres de la commande prédictive à horizon glissant.

#### III.3.b) Architecture du régulateur court terme

Dans le cadre de la technique de commande prédictive, le régulateur court terme comporte différents modules. Le premier module appelé module de prévision et de surveillance regroupe toutes les données disponibles concernant les vols et leur position actuelle ainsi que les données sur la capacité effective et prévue des secteurs de contrôle. Le module effectue alors une simulation accélérée des trajectoires des avions et en déduit une prévision des futures saturations. Si une saturation est détectée dans un horizon de temps que l'on appellera *horizon de prévision* le module de prévision appelle le module d'optimisation. L'horizon de prévision est plus court que l'horizon de prédiction : en effet, du fait des incertitudes sur le système aérien on ne souhaite pas déclencher le mécanisme de régulation pour éviter des éventuelles saturations situées trop loin dans le temps et dont l'occurrence est très incertaine.

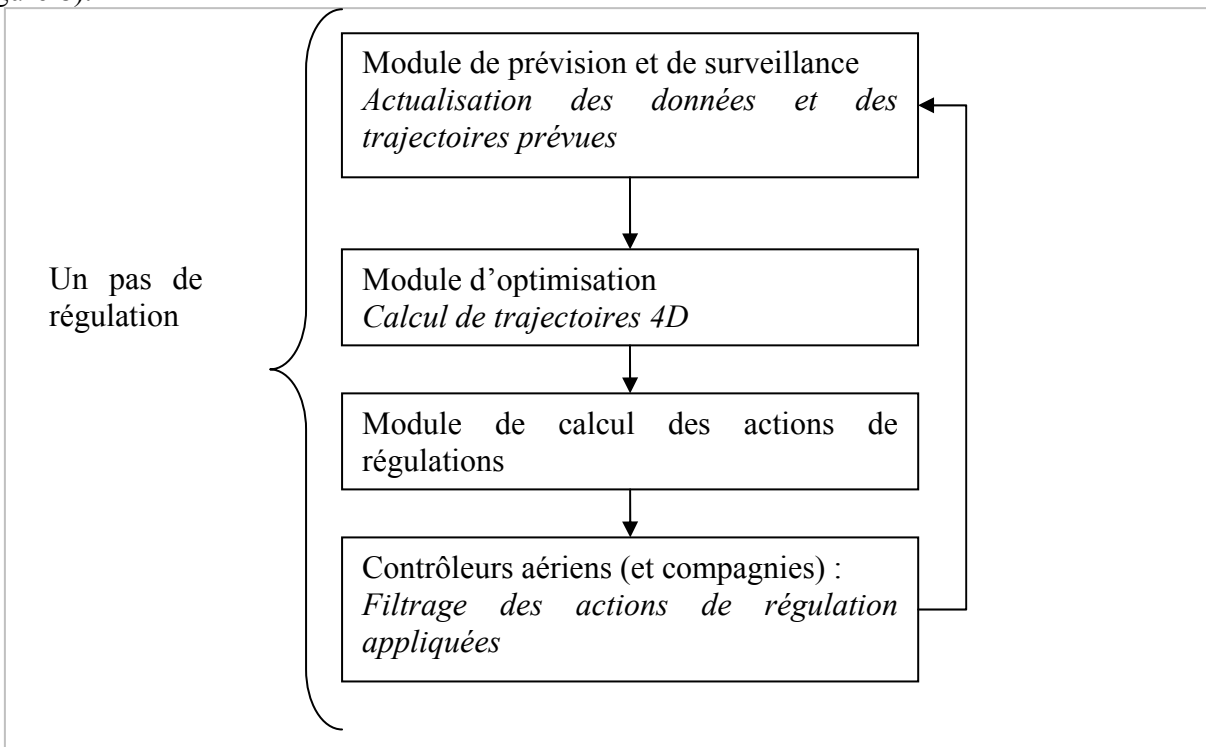
Le module d'optimisation cherche alors à déterminer un ensemble de trajectoires pour les vols commandables empêchant toute saturation durant un horizon appelé *horizon de protection*, plus long que l'horizon de prévision, et ce afin d'éviter de repousser des problèmes de saturation juste en dehors de la période contrôlée. Les trajectoires sont calculées de manière à minimiser le coût total pour tous les vols concernées sur tout l'horizon de prédiction. L'imbrication des différents horizons de temps est résumée dans la Figure 7. Les valeurs envisagées pour les différents paramètres du système sont les suivantes :

- durée entre deux pas de régulation : de l'ordre de 10 minutes
- horizon de prédiction : jusqu'à l'atterrissage du dernier avion commandable.
- horizon de protection : de l'ordre de deux ou trois heures.
- horizon de prévision : de l'ordre de 45 minutes.



**Figure 7) Définition des différents horizons de temps utilisés par le régulateur court terme**

Un module de contrôle des trajectoires traduit les trajectoires 4D en séquences d'actions de régulations applicables aux vols. Ce module qui doit notamment déterminer la politique de vitesse à suivre pour obtenir les dates de passage souhaitées a été étudié plus en détail dans le rapport ([3]). Enfin, les actions calculées et qui doivent s'appliquer durant la période de régulation sont alors proposées aux contrôleurs aériens qui peuvent les appliquer ou non. Dans le cas où aucune saturation n'a été repérée mais que des actions peuvent améliorer l'efficacité d'un vol, elles sont proposées aux compagnies aériennes concernées qui peuvent les accepter ou les refuser. Le processus recommence au pas de régulation suivant (voir Figure 8).



**Figure 8) Architecture globale du régulateur court terme du trafic aérien**

Le chapitre suivant est consacré plus particulièrement au module d'optimisation du régulateur et au choix de la méthode de résolution retenue pour le résoudre.



## chapitre II Le module d'optimisation du régulateur court terme

*Le régulateur court terme du trafic aérien requiert de calculer des actions de régulation à intervalles réguliers. La détermination de ces actions de régulations constitue un problème d'optimisation dont les objectifs, les décisions et les contraintes sont décrits dans ce chapitre. Les choix de modélisation qui vont conduire à la formulation mathématique du problème sont énumérés et justifiés. Dans un second temps, les résultats d'une étude bibliographique menée sur le problème proche nommé TFMP (Traffic Flow Management Problem) sont exposés et la troisième partie, qui en découle, est consacrée à l'explication du choix de la méthode de résolution retenue : la programmation linéaire en nombres entiers associée à la méthode de génération de colonnes.*

### I. Description du problème d'optimisation principal

#### I.1. Description du problème à résoudre

##### I.1.a) Introduction

Les objectifs, contraintes et actions possibles du régulateur court terme étant définis, il faut maintenant en faire la transcription pour le module d'optimisation. Celui-ci est chargé de trouver, en un temps raisonnable, un ensemble d'actions à appliquer aux vols commandables de manière à lisser la charge et à supprimer les saturations, tout en minimisant les conséquences économiques de ces actions. Deux remarques importantes à ce stade :

-étant donné la séquence de tâches à accomplir durant un pas de régulation et la durée réduite de celui-ci, le temps de calcul alloué au module d'optimisation est très limité. L'un des objectifs premiers de ce travail est donc d'évaluer les méthodes mises en œuvre à l'aune du temps de calcul nécessaire à l'obtention d'une solution réalisable.

-on passe maintenant dans le champ de la recherche opérationnelle et un glissement sémantique s'opère sur les termes d'objectifs, de contraintes et d'actions possibles. On parlera désormais de *décisions* à prendre, décisions qui détermineront le respect ou non de *contraintes* et la valeur d'une *fonction objectif* qui évaluera un ensemble de décisions, permettant ainsi de comparer mathématiquement deux ensembles de décisions entre eux. A noter que les objectifs (dans le sens courant) exposés ci-dessus peuvent se traduire par : choisir un ensemble de décisions respectant les contraintes et minimisant la fonction objectif.

##### I.1.b) Ensemble d'actions réalisables et trajectoires 4D

###### i) *Avantages de la représentation par trajectoires 4D*

On cherche à déterminer un ensemble d'actions réalisables pour chaque vol : un tel ensemble d'actions détermine alors la trajectoire prévue en 4 dimensions de l'avion. Pour des raisons de commodité de modélisation, nous avons choisi de travailler à partir de trajectoires 4D qui constitueront la sortie attendue du problème d'optimisation. La manipulation de telles trajectoires présente deux avantages principaux. Tout d'abord, elle permet de se ramener dans le contexte général de la recherche d'itinéraire, champ bien connu de la recherche opérationnelle avec notamment le cadre général de la théorie des graphes. Ensuite, elle permet de vérifier instantanément le respect des contraintes de capacité en donnant directement les

dates d'entrée dans les différents secteurs, dates qu'il aurait fallu de toute façon calculer si l'on avait travaillé à partir des actions de régulation elles mêmes.

*ii) Détermination des actions correspondant à une trajectoire 4D*

Cependant, si le passage d'un ensemble d'actions à une trajectoire 4D prévue est déterministe (mais pas celui à la trajectoire réelle), la détermination d'un ensemble d'actions à appliquer pour obtenir une trajectoire 4D est un problème à part entière. Typiquement, si on a déterminé par exemple le temps de passage entre deux balises, temps de passage que l'on sait réalisable du fait de la connaissance des vitesses limites possibles de l'avion, le problème réside dans la détermination du champ de vitesse à suivre pour respecter le temps de passage. La précision toujours améliorée de la nouvelle génération d'instruments de localisation et de mesures de paramètres de la trajectoire, basés notamment sur le GPS ou bientôt sur Galiléo, rend ce que l'on appelle la navigation 4D envisageable à court terme. D'un point de vue théorique, le rapport du LICIT sur la régulation court terme ([3]) explore l'utilisation de la méthode de Predictive Functional Control (PFC) pour résoudre ce problème de traduction de temps de passage en actions sur la vitesse.

Les trajectoires 4D calculées et choisies par le module d'optimisation devront alors vérifier les contraintes sur les trajectoires, exposées dans le chapitre I paragraphe III.2.c) : performances de l'avion, règles de la navigation aériennes, confort des passagers et de l'équipage.

*iii) Limites à l'utilisation des trajectoires 4D*

Il faut souligner une limite à cette approche : le calcul du coût d'une trajectoire s'effectuera à partir d'une vitesse constante entre deux points de repère et d'un taux de montée ou de descente constant, alors que l'application opérationnelle impliquera probablement des variations de vitesse ou de taux de montée ou de descente pour prendre en compte les conditions réelles (vents, pression...). Il y a donc une différence entre le coût calculé d'une trajectoire et son coût opérationnel : cela n'est pas surprenant et il n'y a pas de biais systématique en faveur ou en défaveur d'un type de trajectoires particulier : ce n'est donc pas un obstacle au travail avec les trajectoires 4D.

I.1.c) Décisions, contraintes et fonction objectif du problème d'optimisation

Le but du module d'optimisation devient de déterminer et de choisir pour chaque vol commandable une trajectoire 4D réalisable. Il faut noter que toutes les contraintes sur la faisabilité des trajectoires sont directement incluses dans cette formulation et n'auront donc pas besoin d'être mentionnées dans les contraintes.

Les contraintes du problème d'optimisation seront alors simplement de respecter les capacités déclarées des secteurs et de lisser la charge. La notion de lissage de la charge sera plus précisément étudiée ci-dessous (I.2).

Enfin la fonction objectif visera à minimiser la somme des coûts des trajectoires retenues, coûts généralisés (consommation, retards...). La définition de la fonction coût d'une trajectoire est également détaillée dans le paragraphe suivant.

## I.2. Traduction des contraintes et des objectifs dans la modélisation

### I.2.a) Capacité des secteurs et lissage de la charge

S'il n'y a qu'une définition officielle de la charge des secteurs, en nombre d'avions entrants par tranche horaire fixe, la prise en compte du lissage de la charge est un peu plus compliquée. Nous présentons ici plusieurs modélisations mathématiques possibles de la charge des secteurs, la dernière seulement ayant été retenue pour des raisons explicitées ci-dessous.

#### i) *Nombre d'avions entrants par secteurs par période fixe*

Dans le système actuel de régulation du trafic aérien la capacité est définie uniquement comme un nombre limite d'avions pouvant entrer dans un secteur pendant une période de temps donnée d'une durée fixe (une ou trois heures). Bien qu'étant la capacité officiellement définie, la capacité en nombre d'avions entrants n'est pas la seule réellement prise en compte dans les processus de régulation du trafic en cas de saturation et notamment par le système ETFMS (Enhanced Tactical Flow Management System, système amélioré de gestion tactique des flux) de la CFMU pour allouer les créneaux de décollage. En effet, lors d'une opération de régulation au dessus d'un secteur, la CFMU divise la période de temps concernée par la régulation par la capacité du secteur, soit le nombre d'avions autorisés à entrer et elle en déduit des créneaux de survol qu'elle alloue aux différents vols (le mécanisme est précisément décrit dans le chapitre I paragraphe I.2.c). Par exemple, pour une période de deux heures avec une capacité de 28 vols par heure, le système va créer une liste de 56 créneaux d'une durée de 2 minutes environ. Ces créneaux seront ensuite attribués tour à tour aux vols concernés. On remarque qu'une telle allocation effectue implicitement un lissage de la charge et va plus loin que le simple respect de la capacité en nombre d'avions entrants. Dans un programme d'optimisation, il est important de préciser les contraintes de capacité explicitement si l'on souhaite lisser la charge : prendre en compte exclusivement la capacité en nombre d'avions entrants pourrait en effet conduire à un pic de charge en début ou en fin de chaque période de régulation, ce qui n'est pas admissible.

#### ii) *Nombre d'avions entrants par secteur sur période glissante*

Une situation aberrante possible avec une définition sur période fixe est de voir arriver par exemple 30 vols (si 30 est la capacité maximale du secteur) juste avant une heure fixe et 30 juste après quand le compte a été recommencé à zéro. Une première manière d'éviter ce genre de situations est de chercher à imposer que le nombre d'avions entrants soit inférieur à la capacité, non seulement pendant des périodes de durée fixe commençant à des dates fixes (généralement les heures piles : entre 8h00 et 9h00 etc.), mais aussi pendant des périodes de durée fixes commençant à d'autres dates plus proches les unes des autres. Ainsi pour  $m$  inférieur à 60 minutes on peut surveiller la capacité entre 8h00 et 9h00, entre 8h00+m et 9h+m, entre 8h00+2m et 9h+2m etc. Si l'on prend  $m$  égal à 15 minutes on multiplie par quatre le nombre de contraintes. Plus on prend  $m$  petit et plus on est assuré du respect réel de la contrainte de capacité en nombre de vols entrants mais plus on augmente le nombre total de contraintes. Si deux périodes de temps initiales consécutives ont des capacités différentes, la capacité d'une période à cheval sur les deux est donnée par la moyenne pondérée par le recouvrement avec chaque période initiale. On n'a par contre pas de garantie sur le lissage de la charge puisque tous les avions autorisés pour une période peuvent toujours arriver en même temps.



iii) *Nombre d'avions présents au dessus d'un secteur*

Une manière d'agir directement sur le lissage de la charge des secteurs est d'introduire une capacité instantanée, c'est-à-dire de limiter le nombre d'avions présents à une date donnée au dessus d'un secteur. Cette approche a l'avantage de prendre en compte explicitement la durée pendant laquelle les vols restent au dessus d'un secteur, car tant qu'ils sont présents, ils demandent de l'attention au contrôleur aérien. Dans le cas où l'on ne considère que le nombre d'avions entrants, on ne prend pas en compte la sortie effective de l'avion du secteur : qu'elle se situe 2 minutes après son entrée ou 15 minutes ne change rien au calcul de la charge. L'inconvénient de l'approche en nombre d'avions présents est d'ajouter un nombre important de contraintes au problème d'optimisation, nombre dépendant de la taille de l'horizon de simulation et du pas de discrétisation du temps. Une autre difficulté réside dans le choix des valeurs à utiliser comme limites étant donné qu'il n'y a rien de tel dans les définitions officielles de capacité de secteurs. Enfin, dans le cas d'un processus basé sur la programmation linéaire en nombres entiers tel celui que nous avons retenu, l'ajout de telles contraintes peut compliquer la résolution générale du problème en rendant plus difficile le respect des contraintes d'intégrité.

iv) *Capacité en minutes de vol avions par heure*

Pour prendre en compte la durée effective de survol d'un secteur par les avions sans introduire la capacité instantanée coûteuse en terme de taille du modèle et en temps de résolution, on peut introduire une mesure intermédiaire : définir pour chaque secteur-période une capacité en temps de survol\*avion. Ainsi, si un secteur dispose de 280 minutes de temps de vol\*avion par heure, il peut accepter 28 vols restant 10 minutes, 20 vols restant 14 minutes etc. La difficulté est ici également de définir une telle capacité puisqu'elle n'existe pas officiellement : il faudrait la déduire de la capacité en avions entrants et de la géométrie du secteur (temps de survol moyen...). Pour prendre en compte cette nouvelle contrainte de capacité, on peut, soit introduire des contraintes déconnectées (une capacité en nombre d'avions entrants et une capacité en temps de survol\* nombre d'avions), soit essayer de créer une fonction d'utilisation de la capacité commune prenant en compte à la fois l'entrée d'un vol dans le secteur et son temps de survol, en considérant qu'un contrôleur ayant moins d'avions entrants dispose de plus de temps pour gérer les avions sur son secteur.

Cette approche nécessite cependant la détermination d'une comparaison entre ces deux éléments distincts de capacité ou plus clairement de répondre à la question : à combien de vols entrants correspond, en terme de charge de travail, la surveillance d'un survol de x minutes, question à laquelle il est difficile de répondre.

v) *Approche retenue : contraintes de capacité sur des sous périodes*

Pour obtenir une répartition de la charge plus équilibrée, une méthode, facilement modélisable et moins coûteuse en terme de taille du modèle, consiste à diviser les périodes de mesure de la capacité pour obtenir des sous-périodes. On leur attribuera la capacité de la période initiale divisée par le nombre de sous-périodes. Ainsi, si l'on divise par 4 une période de une heure disposant d'une capacité de 40 avions entrants, on obtient 4 sous-périodes disposant d'une capacité de 10 avions entrants chacune. Pour laisser plus de liberté au modèle et augmenter la qualité de la solution réalisable, on peut prendre comme capacité des nouvelles périodes réduites un nombre un peu supérieur à la capacité divisée par le nombre de sous-périodes, tout en maintenant la contrainte de capacité sur la période initiale. Ce facteur d'augmentation de la capacité des sous-périodes est un paramètre du système de résolution tout comme le nombre de sous-périodes. Sur notre même exemple, on peut avoir 4 sous-

périodes permettant chacune le passage de 11 (ou 12...) avions entrants et une période globale permettant toujours 40 avions entrants (voir Figure 9). Il y a un compromis à trouver entre le lissage de la solution optimale, la difficulté de résolution du problème et l'obtention d'une solution réalisable satisfaisante en terme de coût (plus les contraintes de lissage sont contraignantes et plus le coût de la solution optimale obtenue augmente).

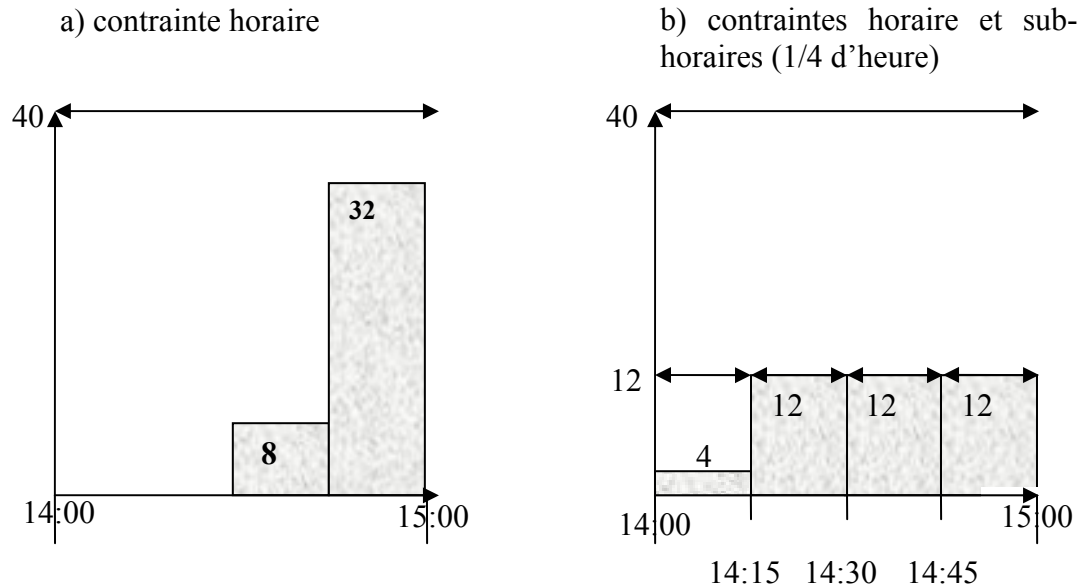


Figure 9) Contraintes de capacité et lissage du flux de trafic

En résumé, on imposera dans le problème d'optimisation la capacité officielle en nombre d'avions entrants, ainsi que sur des périodes plus courtes qu'une heure afin d'obtenir un lissage de la charge. Plusieurs arguments défendent ce choix : le nombre d'avions pouvant entrer dans un secteur pour une heure est la définition officielle de la capacité d'un secteur et on dispose donc de données officielles et réelles. On peut d'ailleurs remarquer que la détermination de ces valeurs de capacité par les centres de contrôle se fait à partir de la charge de travail complète induite par le survol d'un avion. De plus, le flux entrant correspond à une source de charge de travail identifiée : la prise en compte des avions entrants qui nécessite une coordination avec le contrôleur du secteur précédent. Enfin cette approche permet de contrôler le nombre de contraintes du modèle final.

### I.2.b) La fonction coût d'une trajectoire

La fonction coût est une fonction qui associe à chaque trajectoire réalisable un coût : elle permet d'évaluer les trajectoires et de les comparer entre elles. Les coûts communs à toutes les trajectoires d'un même vol tels que les frais initiaux de personnels ou les frais d'assurance ne sont pas pris en compte puisque les considérer consisterait simplement à décaler la fonction objectif vers le haut. On pourra consulter le rapport ([13]) concernant les questions des fonctions coûts dans les problèmes d'optimisation de la régulation du trafic aérien. Les coûts susceptibles de différer d'une trajectoire à l'autre peuvent être alors séparés en trois composantes distinctes :

- Le coût d'exploitation de la trajectoire, comprenant la consommation de carburant sur le parcours, mais aussi le coût d'usure des avions proportionnel à la durée du vol.
- Le coût des retards à l'arrivée : à la fois pour les passagers et pour la compagnie qui pourra avoir des frais supplémentaires de personnel liés aux heures supplémentaires.

-Les coûts de propagation des retards aux correspondances que devaient assurer l'avion, l'équipage ou les passagers.

On explicite dans ce paragraphe les choix effectués pour la fonction coût des trajectoires.

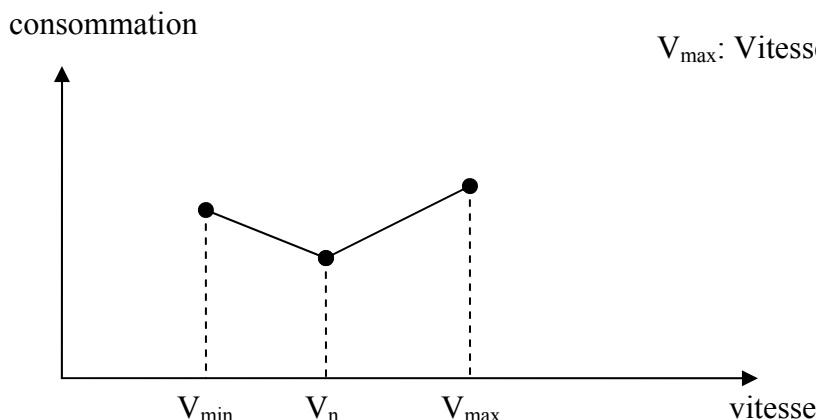
*i) Coût d'exploitation de la trajectoire.*

Le coût principal d'exploitation réside dans la consommation de carburant. Parmi les actions de régulation envisagées se trouve la régulation de vitesse. Or, pour pouvoir estimer le coût d'une telle politique, il faut être en mesure d'estimer la consommation de l'avion en fonction de la vitesse. Dans tout ce qui suit, on appellera vitesse nominale d'un avion en fonction du niveau de vol, la vitesse permettant la consommation minimale. Pour les autres vitesses on utilisera la structure de coût décrite dans la Figure 10. C'est une première approximation par rapport à la réalité, plus proche d'une parabole, mais qui en conserve les principales caractéristiques : minimum à la vitesse nominale, décroissante entre la vitesse minimale et nominale et croissante entre la vitesse nominale et maximale. L'estimation de la consommation de carburant est effectuée notamment à l'aide des données BADA (Base of Aircraft DATA) développées par le centre expérimental d'Eurocontrol (EEC).

$V_n$ : Vitesse nominale

$V_{min}$ : Vitesse minimale

$V_{max}$ : Vitesse maximale



**Figure 10) Consommation en fonction de la vitesse**

Les coûts d'usure des avions ne sont pas pris en compte directement pour cause de manque de données, ils sont proportionnels à la durée de vol qui, elle, est prise en compte dans la composante retard.

*ii) Coût des retards à l'arrivée.*

L'université de Westminster a publié en 2004, pour le compte de la commission d'examen des performances d'Eurocontrol, un rapport ([4]) évaluant le coût pour les compagnies aériennes des retards à l'arrivée. Nous sommes partis de ce rapport pour traduire le coût d'un retard en termes de quantité de kérosène consommé en utilisant de plus le prix du kérosène à la date du début des tests (1.80 \$ par Gallon à Amsterdam au 23 janvier 2006). Le but n'est pas tant d'obtenir une estimation du coût d'une trajectoire aussi proche que possible de la réalité, mais plutôt d'avoir une bonne correspondance entre le prix des retards et le prix du carburant. Cette correspondance va influencer la stratégie optimale puisqu'un carburant cher peut conduire une compagnie à privilégier la consommation à la ponctualité (la consommation augmentant avec la vitesse). A noter que les coûts des retards tirés du rapport

de l'université de Westminster et utilisés sont donnés par passager : nous avons donc pondéré le coût par le nombre de places des avions considérés. Ceci induit un biais en faveur des gros avions qu'il faudrait corriger dans la fonction coût avant une application opérationnelle.

### *iii) Coûts de la propagation des retards*

Comme on l'a déjà souligné, les retards réactifs (retards dus à la propagation durant la journée des retards dits primaires) occupent une place très importante dans les retards totaux observés en Europe. Il est important dans la modélisation de prendre en compte leurs conséquences sur le coût d'une trajectoire. Cependant, les mécanismes de propagation de ces retards sont encore mal connus et peu étudiés. Il est tout à fait irréaliste d'essayer de modéliser la chaîne complète de propagation des retards durant la journée du fait de l'incertitude générale de prévision du trafic, incertitude qui augmente notoirement avec l'horizon de prévision et avec le nombre de vols impliqués. La modélisation retenue prend en compte directement le premier niveau de cette chaîne en considérant explicitement l'ensemble des vols en correspondance pour chaque vol. On connaît l'heure de départ prévue des correspondances. Le coût de retarder un vol en correspondance est également estimé à partir du rapport ([4]) en faisant l'hypothèse que le retard au départ sera le même que le retard à l'arrivée. Enfin, on a modulé le coût des retards réactifs pour prendre en compte le fait qu'un vol retardé le matin a plus de conséquences qu'un vol retardé l'après midi. Cette modulation est effectuée à partir des travaux de Beatty et al. ([12]) qui ont proposé un multiplicateur de retard dépendant de l'aéroport et de la date d'arrivée du vol pour prendre en compte les répercussions des retards primaires. Etant donné que l'on travaille sur des horizons de temps assez courts, les vols commandables considérés sont tous de la même période de la journée et on va considérer qu'aucun vol n'appartient à la fois à l'ensemble des vols dont on cherche à déterminer la trajectoire et à l'ensemble des vols en correspondance.

## **1.3. Autres choix de modélisation**

### **1.3.a) Restriction de l'ensemble des trajectoires étudiées**

L'ensemble des trajectoires réalisables pour un avion est un ensemble continu : en effet, si le choix de l'action est discret (choisir une route aérienne parmi un ensemble limité, choisir un niveau de vol parmi ceux autorisés), les dates d'application de cette action peuvent être continues, notamment pour le changement de niveau. Il existe donc un nombre infini de trajectoires réalisables envisageables. On souhaite effectuer une restriction de l'ensemble des trajectoires considérées tout en gardant une grande liberté d'action pour ne pas nuire à la qualité de la solution réalisable obtenue. On a retenu les règles suivantes :

#### *i) Changement de route horizontale*

D'une manière naturelle et réglementaire, le changement s'effectuera au dessus d'une balise intersection entre deux ou plusieurs routes aériennes

#### *ii) Attente en l'air*

On identifie une série limitée de points où l'attente en l'air est possible et on se limite à des boucles d'attente en l'air au dessus de ces points. Les attentes en l'air sont normalisées de la même manière que pour les attentes aux aéroports : elles correspondent à un nombre variable de boucles à vitesse fixée à un certain nombre de niveaux de vol prédéfinis.

### iii) *Changement de niveau de vol*

On a décidé de ne commencer un changement de niveau de vol qu'au dessus d'une balise, à deux exceptions près. Le premier changement de niveau (à partir du moment où l'on considère le vol) pourra s'effectuer le plus rapidement possible afin de pouvoir prendre des décisions de changement de niveau immédiatement. Le début de la descente terminale qui s'effectue de manière à avoir une descente continue jusqu'à l'aéroport d'arrivée, ce changement de niveau ne constituant pas une action de régulation.

Etant donné que les changements de trajectoire verticale peuvent s'effectuer *a priori* à n'importe quel moment du vol et que l'on a limité cette liberté, ces choix ont un effet restrictif. Ils résultent d'un équilibre à trouver entre deux objectifs concurrents : limiter l'ensemble des trajectoires possibles afin de limiter l'explosion combinatoire et permettre une modélisation d'un maximum de trajectoires intéressantes.

### iv) *Modulation de vitesse*

Pour la modulation de vitesse on va fixer des dates de passage aux différentes entrées secteurs. Comme nous l'avons vu, la politique de vitesse à adopter précisément pour vérifier ces dates sera calculée dans le module de calcul des actions de régulation, alimenté par la sortie du module d'optimisation. A noter que pour s'assurer de l'existence d'un champ de vitesses compatible avec les temps de passage calculés, on considèrera les vitesses minimales et maximales admissibles de chaque vol. Par contre on ne prendra pas en compte de bornes sur l'accélération et la décélération, ce qui pourrait provoquer dans de rares cas particuliers la non-existence de champ de vitesse réel permettant d'obtenir les durées souhaitées. La probabilité d'un tel aléa est cependant faible et les temps de parcours pouvant être obtenus seraient quand même proches de ceux souhaités étant donné la durée de parcours d'un secteur.

### v) *Limitation des changements de niveau de vol*

Malgré la première limitation sur les lieux de changement de niveau de vol, décrite ci-dessus, le nombre de possibilités reste trop important comme le montre la Figure 11. Nous avons donc décidé de limiter le nombre de politiques de trajectoires alternatives possibles en nous basant sur les principes suivants : limiter le nombre de changements de niveau, ne pas faire de régulation temporelle à partir de changements de niveau (on pourrait théoriquement jouer sur la date d'arrivée en un point, non seulement en agissant sur la vitesse, mais aussi en jouant sur les niveaux de vol suivis, étant donné que la vitesse varie avec le niveau), favoriser les politiques qui mènent directement au niveau souhaité. Dans la pratique cela se traduit par le fait que l'on ne gardera dans l'exemple de la Figure 11 que la trajectoire marquée en vert (en gras), sauf si d'autres trajectoires empruntent des secteurs différents auquel cas on les étudiera également.

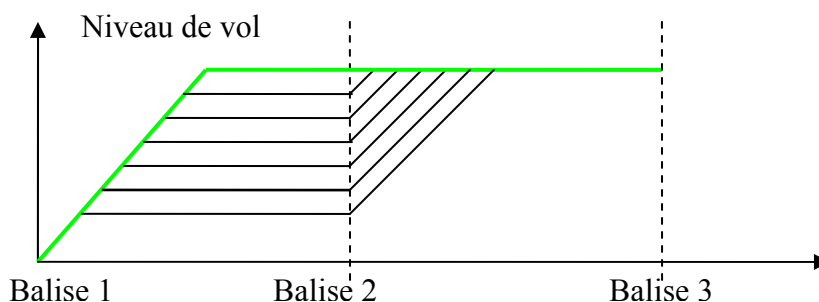


Figure 11 : Un exemple de l'explosion du nombre de politiques de niveau de vol suivi

## **II. Bibliographie sur les méthodes de résolutions candidates**

### **II.1. Introduction**

En résumé, le problème que l'on cherche à résoudre est le suivant : chercher pour chaque vol commandable (on prendra dans la pratique l'ensemble des vols régulables) une trajectoire en 4 dimensions réalisable sur le réseau, en respectant les contraintes de capacité des secteurs et en minimisant la somme des coûts des trajectoires. Ce problème d'optimisation est similaire au problème connu dans la littérature spécialisée sous le nom de TFMRP (Traffic Flow Management Rerouting Problem) mais abordé en temps réel. Ce problème et ceux de la même famille ont fait l'objet d'une étude bibliographique approfondie qui est présentée ici.

### **II.2. Taxinomie des problèmes de régulation ATFM**

#### **II.2.a) Introduction et historique de la recherche dans ce domaine**

Face à l'augmentation constante et continue du trafic (augmentation moyenne annuelle de 9% depuis 1950) et malgré l'augmentation concurrente des capacités de contrôle et des aéroports, le système global de la navigation aérienne a atteint dans les secteurs de grand trafic (Etats-Unis, Europe de l'ouest, Asie de l'est) un degré d'utilisation proche de la saturation provoquant de nombreux retards lors des périodes les plus chargées. Les premières zones victimes de la saturation ont été historiquement les aéroports. Ils restent aux Etats-Unis le principal goulot d'étranglement. Lorsque le nombre d'avions souhaitant atterrir est supérieur à la capacité de l'aéroport, les contrôleurs en charge de l'aéroport sont obligés d'imposer des attentes en l'air très coûteuses pour les compagnies. Face à ce constat, les premières recherches sur le sujet de la régulation du trafic aérien dans le milieu des années 80 ont été basées sur le principe qu'un retard au sol coûte moins cher qu'un retard en vol. Il serait alors plus économique de retarder les avions avant leur décollage lorsque l'on peut prévoir un décalage entre l'offre et la demande. Ces recherches étaient plutôt pensées pour le contexte américain où la capacité de l'espace en l'air n'est pas critique. Etendre les recherches aux cas où la capacité de l'espace en l'air est aussi limitative que la capacité des aéroports a été nécessaire pour pouvoir prendre en compte le cas européen où la saturation des espaces en route est également un problème majeur.

Au fur et à mesure que des chercheurs se sont intéressés à ce sujet et à ses extensions, un certain formalisme s'est imposé dans la désignation des différents problèmes. On présente ici une taxinomie des problèmes étudiés ainsi qu'en deuxième partie un récapitulatif des méthodes employées pour les résoudre. On s'intéressera principalement au problème nommé TFMRP (Traffic Flow Management Rerouting Problem) qui est le plus proche du problème d'optimisation que l'on cherche à résoudre dans le cas de la régulation dynamique : déterminer pour chaque avion une trajectoire espace-temps dans le réseau aérien afin de supprimer les saturations de secteur et minimiser le surcoût. Il existe cependant une différence fondamentale entre tous les articles cités ci-dessous et le problème que l'on cherche à résoudre : l'horizon d'application. La plupart des applications sont prévues pour une régulation tactique ou pré-tactique du trafic aérien alors que nous nous plaçons dans le cadre d'une régulation à court terme ce qui nous impose de nombreuses contraintes supplémentaires en termes de temps de calcul et de finesse de la modélisation notamment.

## II.2.b) Le Ground Holding Problem

Basé sur l'idée qu'un retard au sol est moins coûteux qu'un retard en l'air, ce problème cherche à allouer des retards au décollage aux avions afin d'éviter des temps d'attente dus à la capacité limitée des aéroports lors de leur arrivée. Le GHP est aussi quelquefois appelé *Slot Allocation Problem* (SAP) dans la littérature spécialisée. Les premières recherches sur ce sujet ont été effectuées aux Etats-Unis où les problèmes de saturation sont essentiellement localisés au niveau des aéroports, ce qui explique que la capacité des secteurs n'ait pas été prise en compte dans les premiers modèles. D'une manière naturelle les chercheurs se sont d'abord intéressés à la version la plus simple du problème dont la résolution pourra ensuite servir de base pour la résolution des extensions. Dans cet esprit, la version la plus ancienne et la plus simple du GHP est le *Single-Airport Ground Holding Problem* (SAGHP) dans lequel on ne considère qu'un seul aéroport d'arrivée pour un ensemble donné d'aéroports de départ. On considère au contraire un réseau entier d'aéroports dans le *Multi-Airport Ground Holding Problem* (MAGHP). A noter que dans ce cas, on introduit également l'interdépendance des vols entre eux et que les retards se propagent ainsi dans le réseau en prenant en compte le fait qu'un avion effectue généralement plusieurs vols dans la journée.

Si aux Etats-Unis les saturations sont limitées aux aéroports, ce n'est pas du tout le cas en Europe en raison de la proximité de grands pôles d'échanges et de la complexité induite par le découpage territorial. Pour pouvoir prendre en compte ces saturations, on peut introduire la capacité de l'aéroport de départ ou la capacité des secteurs en vol tant pour le SAGHP que pour le MAGHP : on parle alors de Air Traffic Flow Management Problem. On trouve des auteurs précisant plutôt « with en-route capacities ».

## II.2.c) Le Air Traffic Flow Management Problem (TFMP)

Ce modèle, aussi appelé Generalized Tactical Traffic Flow Management Problem (TFMP) est une extension des problèmes ci-dessus puisque les contraintes de capacité des secteurs sont systématiquement prises en compte. On peut de plus ajouter la possibilité de modifier la vitesse des avions pendant leur vol, voire d'imposer des délais en vol (c'est très similaire d'un point de vue de la modélisation), tout en respectant les capacités des aéroports de départ et d'arrivée et des secteurs. Enfin on peut également ajouter la possibilité de changer la route des avions : c'est le *Air Traffic Flow Management Rerouting Problem* (TFMRP).

## II.2.d) Les différentes versions

On trouve dans la littérature spécialisée de nombreuses approches de ces différents problèmes.

### i) *Statique-dynamique*

Lorsque l'on calcule une solution définitivement pour la journée le modèle est dit statique. Il est dit dynamique si on actualise au cours du temps lorsque de nouvelles données deviennent disponibles.

### ii) *Simple période- multi période*

On effectue la régulation sur une seule période ou on divise l'horizon en plusieurs périodes. A noter que cette distinction n'est quasiment jamais précisée puisqu'on utilise

naturellement des modèles multi-périodes, les simples périodes n'ayant qu'un intérêt méthodologique et encore limité.

iii) *Déterministe probabiliste*

Si les différentes capacités sont connues à l'avance on parle de modèle déterministe contre probabiliste ou stochastique si ce n'est pas le cas.

Plusieurs extensions de ces modèles ont été envisagées : on peut ainsi considérer une contrainte d'équité (fairness) entre les différentes compagnies aériennes concernées par la régulation. On peut également prendre en compte la dépendance entre les capacités de décollage et d'atterrissage des aéroports. En effet, ce sont souvent les mêmes pistes qui sont utilisées ou alors les différentes pistes se croisent interdisant une utilisation simultanée de toutes les pistes.

Dans le cadre de fonctionnement des hubs, on peut considérer qu'un vol ne doit pas être nécessairement assuré par un seul avion mais qu'il existe un ensemble d'avions qui peuvent terminer le vol après un arrêt au hub. Toujours dans ce cadre on peut également chercher à limiter le temps écoulé entre le premier vol arrivé et le dernier vol reparti (banking constraint). Ce dernier problème a fait l'objet de nombreux travaux.

II.2.e) Récapitulatif

	un seul aéroport	plusieurs aéroports	capacité des secteurs en-route	reroutement possible
SAGHP				
MAGHP				
TFMP				
TFMRP				

II.2.f) Les problèmes de routage et de reroutement dynamique

Deux types de problèmes présents dans la littérature spécialisée présentent des points communs avec le problème d'optimisation du régulateur court terme, mais abordés à des échelles de temps différentes : plus en amont pour les problèmes d'allocation de routes et de niveaux, plus en aval pour les problèmes de reroutement dynamique.

i) *Allocation de route et de niveaux aux avions*

Un problème courant dans la recherche sur les transports est le problème de routage des véhicules sur un réseau (voir notamment [64]). Dans le cas du trafic aérien ce problème consiste à déterminer pour une journée donnée les routes, les niveaux de vol, voire les créneaux de décollage alloués aux vols planifiés. Ce problème est abordé d'un point de vue planification, c'est-à-dire plutôt dans le cadre de l'ATFM pré-tactique, quelques jours avant le décollage. La fonction objectif recherche alors à minimiser les conflits potentiels de manière à diminuer la charge des contrôleurs. Ce problème vise à rationaliser le dépôt des plans de vols.



## ii) *Le reroutement dynamique*

Dans les cas où des conditions météo extrêmes et dangereuses empêchent le survol de zones complètes de l'espace aérien, le problème de reroutement dynamique cherche à trouver une trajectoire pour un avion évitant la zone de perturbation tout en minimisant la consommation supplémentaire de carburant. Ce problème se situe quasiment dans le cadre du contrôle aérien en temps réel.

## II.3. Etat de l'art sur le GHP et ses extensions

On s'intéresse au Ground Holding Problem pour ses similitudes avec le problème d'optimisation du régulateur court terme. Cependant, l'hypothèse selon laquelle seuls les aéroports ont une capacité limitée est fautive dans notre contexte. C'est pourquoi l'état de l'art est concentré sur le TFMP et ses extensions qui considèrent les capacités en route et que la résolution du GHP n'est que sommairement évoquée à titre indicatif.

### II.3.a) Le modèle GHP

Odoni (1987 [20]) a été le premier à formaliser le problème GHP. Différentes versions du SAGHP ont été étudiées entre 1987 et 1997 : de statique à dynamique probabiliste (Terrab et al., Andreatta et al., Richetta et al., Hoffman, Ball ...). Le MAGHP a été étudié de même par de nombreux auteurs (Vranas et al., Andreatta et al., Brunetta et al.). Les détails de ces travaux ne sont pas donnés pour la raison évoquée ci-dessus mais on pourra notamment se reporter à Andreatta et al (1993, [19]) pour un état de l'art détaillé du SAGHP et à Hoffman et Ball (1997, [18]) pour une étude des différents modèles du SAGHP avec des contraintes de bancs de vols (banking constraints). Récemment, les travaux concernant le GHP se sont orientés en prenant en compte les nouvelles procédures américaines basées sur le CDM (voir note 6 p.19), les calculs et les allocations de créneaux ne s'effectuant alors plus vol par vol mais pour des groupes de vols. Les modèles récents ([16]) n'ont donc plus de rapport proche avec le problème d'optimisation que l'on souhaite résoudre.

### II.3.b) Le modèle TFMP

Le problème TFMP (Traffic Flow Management Problem) a été étudié pour la première fois dans les années 1993-1995. Depuis les travaux se sont multipliés sur le sujet en augmentant à la fois la complexité des modèles et la taille des instances testées. On donne ici un bref aperçu des résultats obtenus en terme de temps de calcul (on donne les plus grosses instances traitées dans les articles) et des méthodes générales d'approche du problème. Il est à noter que les plus anciens de ces articles datent de 1994 et que les temps de calcul indiqués sont donc à considérer en gardant à l'esprit l'amélioration de la puissance de calcul des ordinateurs. Un récapitulatif de ce travail bibliographique est exposé dans le paragraphe suivant (II.3.c).

## i) *Tosic, Babic Cangalovic, Hohlacov (1994, [31])*

Cet article ne se place pas dans le cadre opérationnel mais est plutôt un travail de défrichage sur le comportement des algorithmes proposés. Les instances et les temps de calculs ne sont donc pas réalistes. Une originalité de cette approche réside dans la modélisation de la demande. L'importance du choix de la fonction coût, et donc de la politique de régulation adoptée, sur la nature des résultats et sur la répartition des retards entre les différents usagers est clairement mise en évidence.

Modélisation : programmation linéaire en nombres entiers, variables de décision  $x_{ik}=1$  si l'avion  $i$  est retardé de  $k$  unités, 0 sinon.

Résolution : par un algorithme itératif basé sur une succession de résolution de programmes linéaires où les contraintes d'intégrité sont relaxées et utilisation d'heuristiques d'arrondi pour obtenir une solution entière.

Actions possibles : On ne peut jouer que sur le retard au sol.

La demande est modélisée par unité de capacité requise par avion par secteur et par pas de temps. Comme un avion peut requérir de la capacité de plusieurs secteurs pendant la même période de temps, la marche des avions peut être plus fine que la discrétisation temporelle.

Ordinateur : 486/33 PC

Instance et temps de calculs :

4 secteurs, 162 vols, 4 heures (pas de temps 30 mn). 10 à 30 minutes de temps de calcul

ii) *Tosic, Babic Cangalovic, Hohlacov (1997, [28])*

De la même manière que dans l'article précédent, le but n'est pas ici de proposer un outil opérationnel d'aide à la décision, mais de proposer une approche prenant en compte la possibilité du reroutement. On peut ainsi évaluer sommairement le gain que l'on peut en attendre, qui est très positif sur les instances étudiées. Ce modèle est simplement une extension du modèle précédent pour prendre en compte le reroutement.

Modélisation : programmation linéaire en nombres entiers, variables de décision  $x_{ik}=1$  si l'avion  $i$  est retardé de  $k$  unités, 0 sinon. A noter que les routes alternatives sont modélisées comme des avions supplémentaires et que pour un vol tous les  $x_{ik}$  peuvent être nuls, ce qui signifie qu'une autre route a été choisie.

Résolution identique à l'article précédent.

Actions possibles : on peut jouer sur le retard au sol et le reroutement.

Instance :

6 secteurs, 1 route alternative pour 71 vol sur 162, 4 heures (pas de temps 30 mn), retard limité à 4 périodes. Pas de temps de calcul fourni (ce n'est pas l'objectif des tests).

iii) *Maugis (1995, [30])*

Cet article développe un modèle d'optimisation destiné à être intégré à un outil d'aide à la décision pré-tactique. Trois approches distinctes sont proposées mais c'est finalement la plus classique qui est retenue : la programmation linéaire en nombres entiers. Le modèle présenté est très flexible et permet de nombreuses extensions (surcharge de certains secteurs, connectivité des vols, reroutement, bancs de vols, agrégation de vols...) mais les tests ne sont effectués que pour l'affectation des retards au sol. Une étude détaillée des différentes mesures possibles d'inéquité entre les vols est effectuée.

Modélisation : Trois méthodes de résolutions sont proposées.

- Programmation linéaire en nombres entiers

Variables de décision  $u_{f,t,i}$  : 1 si l'avion  $f$  est retardé de  $t$  au dessus du  $i^{\text{ème}}$  secteur de sa route 0 sinon. (remarque  $u_{f,t,i}$  n'est pas le temps passé en  $i$  : c'est le retard accumulé quand on passe en  $i$ ). Résolution du programme relaxé et utilisation d'une technique de branch-and-bound pour obtenir une solution entière.

- Multi-flot et algorithmes d'approximation
- Recherche heuristique locale (recuit simulé et recherche tabou)

Les tests montrent que la programmation linéaire est la plus efficace des trois méthodes  
Actions possibles : Retard au sol, annulation de vols (pas sur toutes les instances testées)

Ordinateur : SPARC-5 , IBM OSL

Instance et temps de calculs :

1153 secteurs-périodes saturés, 4753 vols, une journée (pas de temps 5 mn). Plus de deux heures de temps de calcul pour les plus grosses instances.

iv) *Bertsimas and Stock (1998, [26])*

Cet article est un article de référence dans la littérature spécialisée consacrée à la résolution du TFMP. Les auteurs proposent une modélisation en programmation linéaire novatrice qui a la particularité intéressante de comporter des facettes de l'enveloppe convexe des solutions réalisables, ce qui permet de résoudre le problème par une résolution du problème relaxé et d'obtenir presque toujours des solutions entières. Les tests assez poussés effectués sur de grandes instances viennent confirmer les résultats mathématiques sur les polyèdres. A noter cependant qu'aucune garantie d'intégrité n'est donnée théoriquement et que Vranas (1997) en utilisant ce même modèle a obtenu des solutions optimales non entières. De nombreuses extensions du modèle sont proposées théoriquement (interdépendance des capacités de départ et d'arrivée des aéroports, reroutement, bancs de vols, connections multiples...) mais les tests ne sont effectués que dans le cas le plus simple et on n'a pas d'information sur les propriétés d'intégrité du modèle étendu.

Modélisation : programmation linéaire, variables de décision:  $w_{ft}^j = 1$  si l'avion  $f$  arrive ou est déjà passé au secteur  $j$  en  $t$  0 sinon.

Résolution du programme relaxé (on tombe alors presque toujours sur des solutions entières)

Actions possibles : Prend en compte la connectivité des vols et permet les attentes en vol et le contrôle de la vitesse.

Ordinateur : Sun SPARC station 10 model 41; CPLEXMIP 2.1

Instances et temps de calculs :

4 aéroports, 3 secteurs, 200 vols, 24 heures (pas de temps 5mn). 234 secondes de temps de calcul.

4 aéroports, 3 secteurs, 1000 vols, 24 heures (pas de temps 15 mn). 436 secondes

10 aéroports, 178 secteurs, 278 vols, 7 heures (pas de temps 5mn). 19 minutes pour 18 733 variables et 43 226 contraintes.

18 aéroports, 305 secteurs, 1002 vols, 8 heures (pas de temps 5 mn). 8 heures 18 minutes pour 69 497 variables et 151 662 contraintes.

v) *Bertsimas and Stock (2000, [24])*

Après avoir écrit un article sur le TFMP, Bertsimas et Stock essaient ici d'ajouter les possibilités de reroutement en utilisant une approche différente. Ils effectuent une modélisation en un multi-flot dynamique puis calculent à partir d'un algorithme de relaxation lagrangienne et d'un algorithme probabiliste un ensemble de trajectoires pour les avions, ils choisissent une de ces trajectoires par un programme linéaire en nombres entiers.

Modélisation : multi-flot dynamique qui débouche sur un programme linéaire en nombres entiers.

Résolution par un algorithme basé sur la relaxation lagrangienne des contraintes de capacité, construction d'un ensemble de chemins possibles pour chaque flot simple à partir

d'un processus probabiliste et résolution du programme linéaire choisissant entre ces chemins. On boucle jusqu'à ce qu'on ait un résultat approché satisfaisant.

Actions possibles : Prend en compte la connectivité des vols, permet le retard au sol, en l'air, la modulation de vitesse et surtout le re-routement.

Ordinateur: Sun SPARC 20 workstation avec 48 MB

Instances :

4 aéroports, 42 secteurs, 71 vols, 8 heures (pas de temps 5mn). Temps de calcul : 181 secondes de recherche de la borne inférieure et 330 secondes pour l'algorithme en lui-même pour 61 912 variables et 24 509 contraintes. 0,4% de la borne inférieure

4 aéroports, 42 secteurs, 271 vols, 8 heures (pas de temps 5mn). 86+169 secondes de temps de calcul pour 66 489 variables et 25 881 contraintes. 0,92 % de la borne inférieure.

vi) *Vranas (1997, [29])*

Cet article s'intéresse au problème du TFMP dans le cadre de la régulation tactique en Europe effectuée par la CFMU par CASA (Computer Assisted Slot Allocation). Le problème posé est alors l'allocation optimale des créneaux de décollage. L'auteur propose une nouvelle modélisation en programmation linéaire en nombres entiers qui comporte beaucoup moins de variables que la version de Bertsimas et Stock. Une étude assez complète est alors réalisée pour comparer les allocations effectuées par CASA et celles obtenues en résolvant le problème de manière optimale à partir du modèle de Bertsimas and Stock.

Modélisation : programme linéaire en nombres entiers. Variables de décision :  $p_{s,t,t'}^i$  : nombre d'avions censés passés dans le secteur  $s$  en  $t$  et en sortir en  $t'$  mais qui sont retardés de  $i$ ,  $g_f$  : nombre d'unité de temps de retard alloués au vol  $f$ . On peut facilement ajouter les contraintes de capacité des aéroports.

Actions possibles : La seule action envisagée est le retard au sol.

Ordinateur : SunSparc 5 100MHz, 192 Mb de RAM, CPLEX 4.0.

Instances :

Plusieurs scénarios ont été étudiés à partir de données réelles sur le trafic aérien européen survolant la France pendant une journée : jusqu'à 44 secteurs régulés (c'est-à-dire avec une capacité limitée) pour 2293 vols. Temps de calcul : de 5,6 minutes à 200 minutes pour trouver une première solution entière réalisable, il faut ajouter ensuite le temps de calcul pour trouver une solution optimale (jusqu'à 866 itérations de branch-and-bound). A noter qu'un problème de 22 secteurs saturés pour 177 vols a été résolu de manière optimale en 339 secondes avec une meilleure version de CPLEX

vii) *Van den Akker, Nachtigall (1997, [17])*

Cet article s'attaque au problème du TFMP sans prendre en compte les possibilités de reroutement. Le modèle utilisé est le même que celui de Maugis et les données des tests sont également les mêmes. Les auteurs utilisent en revanche un processus de génération de colonnes pour résoudre le programme linéaire posé et les résultats sont très encourageants. Ils fournissent également une heuristique pour obtenir à chaque itération du processus de génération de colonnes une solution entière qui, dans les résultats expérimentaux, est assez bonne.

Modélisations : programmation linéaire en nombres entiers, variables de décision  $x_{f,t}$  : 1 si l'avion  $f$  part à la date  $t$ , 0 sinon. Résolution par un processus de génération de colonnes et possibilité d'obtenir des solutions entières par des heuristiques d'arrondi.

Actions possibles : On ne peut jouer que sur le retard au sol.

Ordinateur : SPARC-5 , CPLEX 4.0

Instance et temps de calculs:

1153 secteurs-périodes saturés, 4753 vols, une journée (pas de temps 5 ou 1 mn). Entre 490 et 670 secondes de temps de calcul, soit dix fois moins que Maugis sur les mêmes instances mais les calculs ont été effectués deux ans plus tard.

viii) *Oussedik, Delahaye, Schoenauer (1998, [27])*

Le problème de la gestion du trafic aérien est abordé de manière différente et plus indépendante des modèles de Bertsimas et Stock. Ce n'est pas le cas des autres approches utilisant la programmation linéaire en nombres entiers. Ici, c'est la technique de l'optimisation stochastique et notamment des algorithmes génétiques qui est retenue. Outre cette différence de méthode, à noter également une différence importante d'objectifs puisque les auteurs cherchent à minimiser non pas le retard total en l'air, comme c'est le cas dans la plupart des autres articles sur le TFMP, mais la congestion. Cet article en propose une modélisation fine et, privilège de l'optimisation stochastique, pas du tout linéaire.

Modélisation : on dispose d'une flotte d'avions pour lesquels on connaît la route et le créneau de décollage ainsi que des ensembles de créneaux de décollage et de routes alternatives possibles. On cherche une allocation de créneaux et de routes de manière à réduire les pics de congestion. On code alors chaque planning en un ensemble de couples  $(r_k^n; \delta_j^n)$  : on choisit pour le vol n la route k et le créneau j parmi les ensembles de routes et de créneaux possibles pour le vol n. Les résultats sont encourageants en terme de qualité mais les temps de calculs sont importants : cette méthode serait donc plus adaptée à une utilisation dans un cadre pré-tactique.

Résolution. Algorithmes génétiques basés sur la mutation et le croisement des individus de la population précédente en adaptant les probabilités de mutations et les règles de croisements pour prendre en compte la spécificité du problème de gestion du trafic aérien.

Actions possibles : Retard au sol (limité) et reroutement parmi des routes alternatives précalculées.

Ordinateur : Pentium 300MHz

Instance et temps de calculs :

89 secteurs, 5820 vols, une journée (pas de temps 2 minutes). 4 à 6 heures de temps de calcul

ix) *Alonso, Escudero, Ortuno (2000,[25])*

Les auteurs proposent une extension du modèle de Bertsimas and Stock pour prendre en compte le caractère stochastique du système. La modélisation stochastique est faite par un arbre de scénarios. L'approche la plus prometteuse « full recourse policy » consiste à chercher une solution par scénario en respectant le principe de non anticipation : pour deux scénarios identiques jusqu'à la date t, les décisions prises jusqu'à la date t sont les mêmes. Ainsi, on ne communique les décisions qu'à la date t en fonction du parcours dans l'arbre des scénarios. Cet article complet présente des tests encourageants sur des instances de taille réaliste.

Modélisation : programmation linéaire en nombres entiers (Bertsimas & Stock) et modélisation du caractère stochastique par un arbre de scénarios.

Résolution : utilisation d'une heuristique basée sur la résolution successive de problèmes relaxés.

Actions possibles : Prend en compte la connectivité des vols, permet les attentes en vol et le contrôle de la vitesse. Le caractère stochastique des capacités est modélisé.

Processeur : pentium 200MHz 32Mb de RAM

Instances :

14 instances ont été testées avec de 40 à 160 vols, 4 aéroports, 4 à 5 secteurs, de 24 à 48 pas de temps et de 13 à 41 vols continués. L'attente en l'air a été limitée à 4 périodes de temps et l'arbre des scénarios comptait 9 scénarios possibles.

Les capacités des secteurs sont affectées par un processus aléatoire mais la capacité d'un secteur à un pas de temps dépend de la capacité de ce secteur au pas de temps précédent.

Taille maximale des instances : 32455 contraintes, 16753 variables. Temps de calcul de 71.57 secondes au minimum et de 46 minutes au maximum, 16 minutes en moyenne pour une garantie moyenne d'optimalité de 0.196 %.

x) *Barnier Brisset Rivière (2001, [23])*

On s'intéresse ici au problème d'allocation des créneaux de décollage en prenant en compte la capacité en l'air. L'approche retenue est celle de la programmation par contraintes. Plusieurs modèles sont proposés, le plus efficace est celui comportant des intervalles de temps glissants pour vérifier les contraintes de capacité. Le contexte envisagé est la régulation pré-tactique et les solutions sont évaluées en terme de qualité des résultats sans prendre en compte le temps de calcul.

Modélisation : on dispose d'une flotte d'avions pour lesquels on connaît la route et le créneau de décollage. On cherche à allouer à chaque vol un retard pour respecter les contraintes de capacité (nombre d'avions par période, généralement 30 minutes). Les variables de décision sont les retards affectés à chaque vol. On cherche de plus à minimiser le retard maximal attribué.

Résolution : Programmation par contraintes.

Actions possibles : Retard au sol (limité)

Instance et temps de calculs:

140 secteurs, 7375 vols, une journée (pas de temps 5 minutes, capacité sur une fenêtre de 30 minutes). Pas de temps de calcul fournis.

### II.3.c) Récapitulatif de l'étude bibliographique

Tous les travaux étudiés ici ont pour point commun d'aborder le problème d'allocation de créneaux de décollage et/ou de routes aériennes pour les vols de manière à respecter les contraintes de capacité des secteurs et des aéroports tout en minimisant le coût de la politique choisie, à l'exception des travaux de Oussedik et al. ([27]) qui visent une minimisation de la complexité du trafic. Une grande diversité d'outils de la recherche opérationnelle sont utilisés pour résoudre ce problème : programmation linéaire en nombres entiers (résolue par branch-and-bound ou heuristiques), programmation par contraintes, algorithmes génétiques, méta-heuristiques. On a résumé les résultats de ces articles dans le Tableau 1. Malgré le nombre relativement élevé de travaux concernant le TFMP, aucune approche proposée ne peut s'appliquer directement à la résolution du problème d'optimisation du régulateur court terme pour une raison principale : la différence du contexte de régulation, à court terme plutôt que tactique ou pré-tactique. Par ailleurs, la modélisation et la détermination des trajectoires alternatives sont les principaux points d'achoppement réduisant l'applicabilité de ces travaux à notre problème. Cependant, la structure mathématique du problème est la même et les enseignements tirés des travaux précédents ont été utilisés dans le processus de résolution que nous avons développé.

En dehors du TFMP, les problèmes de routage et de reroutement dynamique présentent certains points communs avec le problème que l'on cherche à résoudre. Il existe cependant des différences structurelles qui sont abordées dans le paragraphe suivant.

Auteurs	Modélisation, contraintes	Contraintes Objectifs	Actions possibles	Instance testée	Horizon/ discrétisation	Temps de calcul
<b>Tosic &amp; al. (1995)</b>	PLNE : heuristiques	capacités	retard au sol (non limité)	4 secteurs ,162 vols	4 heures /30 mn	10 à 30 mn
<b>Tosic &amp; al. (1995)</b>	PLNE : heuristiques	capacités	retard au sol (4 périodes) reroutement	46 secteurs ,162 vols, 71 ayant une route alternative	4 heures /30 mn	non donné
<b>Bertsimas &amp; Stock (1995)</b>	PLNE, relaxation des contraintes d'intégralité	capacités, vols continués	retard au sol, retard en l'air <sup>1</sup>	18 aéroports, 305 secteurs, 1002 vols	8 heures/5 mn	8 heures 18 mn
<b>Vranas (1997)</b>	PLNE, relaxation puis branch & bound	capacités, vols continués	retard en l'air <sup>2</sup> tests sur le modèle B-S	22 secteurs saturés (+ les autres), 1 777 vols	1 journée/ 15 mn	5mn 39 secondes
<b>Oussedik &amp; al. (1998)</b>	Algorithmes génétiques	capacités, minimisation de la congestion	retard en l'air (entre 60 et 90 mn) reroutement	89 secteurs 5820 vols	1 journée/ 2mn	4 à 6 heures
<b>Bertsimas &amp; Stock (2000)</b>	multi-flot : génération de chemins 4D puis PLNE pour choisir	capacités, vols continués	retard au sol, retard en l'air, reroutement <sup>3</sup>	4 aéroports, 42 secteurs, 71 vols	8 heures/ 5 mn	4 mn
<b>Alonso &amp; al. (2000)</b>	PLNE et résolution par relaxations et heuristiques	capacités stochastiques , vols continués	retard au sol, retard en l'air <i>4 périodes de retard possible</i>	4 aéroports, 5 secteurs, 160 vols, 9 scénarios de capacité	48 pas de temps (4 heures/ 5 mn par exemple) <sup>3</sup> périodes pour les variations de capacité <sup>4</sup>	16 mn en moyenne, 31 mn pour celle-ci
<b>Maugis (1995)</b>	PLNE et relaxation des contraintes d'intégralité	capacités en route	retard au sol (240 mn max) annulation de vols	1153 secteurs-périodes saturés, 4753 vols	1 journée/ 5mn	plus de deux heures <sup>5</sup>
<b>Akker &amp; Nachtigall</b>	PLNE et génération de colonnes	capacités en route	retard au sol (240 mn max)	1153 secteurs-périodes saturés, 4753 vols	1 journée/ 5ou 1 mn	entre 7 et 10 mn

<sup>1</sup> pas d'indications sur le nombre de retard possibles

<sup>2</sup> on cherche le retard max minimal pour lequel il y a une solution réalisable

<sup>3</sup> pas d'indications sur le nombre de retard possibles, ni sur le nombre de chemins spatio-temporels possibles (dans la solution retenue de l'ordre de 3 routes spatiales distinctes utilisées)

<sup>4</sup> c'est-à-dire qu'il n'y a pas de variabilité stochastique des capacités pour les pas de temps à l'intérieur d'une même période

<sup>5</sup> sans les possibilités d'annuler les vols

**Tableau 1) Récapitulatif de l'état de l'art sur le TFMP**

### II.3.d) Les problèmes de routage et de reroutement dynamique

Ces deux problèmes diffèrent du problème d'optimisation étudié ici d'un point de vue contexte et les contraintes et objectifs sont différents. Les méthodes employées pour les résoudre ne peuvent que partiellement s'appliquer au problème d'optimisation du régulateur court terme. Le premier problème se situe dans le cadre de la planification pré-tactique et vise à aider au dépôt des plans de vols tandis que le second concerne plutôt le contrôle aérien en vol. Le problème d'allocation de routes et de niveaux de vol cherche à minimiser une quantité de conflit sans prendre en compte de capacité de secteurs tandis que le problème de reroutement dynamique s'affranchit des routes aériennes pour calculer une trajectoire

respectant des règles de séparation minimale, le point important étant la modélisation de la perturbation orageuse. Enfin le degré de finesse de modélisation requis étant très différent, les indications bibliographique concernant ces problèmes renvoient simplement à des publications pouvant servir de porte d'entrée à une bibliographie plus détaillée.

Les problèmes d'allocation de routes et de niveaux de vol peuvent être étudiés d'un point de vue statique qui se modélise alors comme un problème de coloriage de graphe (Constans et al. [32], Barnier et al. [34], Fondacci et al. [35]). Le reroutement dynamique quant à lui est souvent abordé par des processus de programmation dynamique, on pourra notamment consulter Nilim et al. ([33]).

L'objet de la prochaine partie est de détailler les raisons qui nous ont conduits au choix de la méthode de la génération de colonnes comme base de notre processus de résolution.

### **III. Choix de la méthode de génération de colonnes**

#### **III.1. Pourquoi la programmation linéaire en nombres entiers ?**

##### **III.1.a) Distinction entre les deux parties du problème d'optimisation**

Le problème d'optimisation du régulateur court terme comporte deux parties qui bien qu'étroitement liées peuvent être considérées séparément : la détermination de trajectoires alternatives et ensuite l'allocation à chaque vol d'une trajectoire parmi celles proposées. Les articles s'intéressant au problème TFMP se sont généralement focalisés sur ce second problème d'allocation alors que le problème de détermination des trajectoires est central dans notre contexte.

##### **III.1.b) Les enseignements de la littérature spécialisée**

Supposons tout d'abord que les trajectoires alternatives aient été déterminées et que le problème consiste à allouer à chaque vol une trajectoire parmi celles possibles. L'étude bibliographique nous a appris que la méthode de résolution la plus efficace parmi celles testées est la programmation linéaire en nombres entiers (notamment les travaux de Maugis [30] qui comparent trois approches différentes). La PLNE présente également l'avantage d'une très grande souplesse de modélisation permettant d'introduire facilement les extensions souhaitées (notamment la prise en compte des correspondances). Enfin la conjonction de l'augmentation de la puissance de calcul des ordinateurs et de l'amélioration des logiciels spécialisés dans la résolution des programmes linéaires rend possible la résolution de très grands programmes linéaires en un temps raisonnable. Etant donné que la taille des instances résolues dans certains des articles étudiés est déjà proche de la taille du problème à résoudre, une approche basée sur la programmation linéaire n'est pas, a priori, irréaliste.

##### **III.1.c) Les progrès de la résolution des programmes linéaires (classiques ou mixtes)**

La progression des logiciels commerciaux de résolution de programmes linéaires est illustrée par Bixby et al. ([38], [40]). Ainsi, pour résoudre une instance du problème de programmation linéaire dit « Patient Distribution System » comprenant 49 944 contraintes, 177 628 variables pour 393 657 coefficients non nuls, il faut aux différentes versions du logiciel de résolution de programmes linéaires CPLEX sur un ordinateur à 296 MHz:



CPLEX 1.0 (1988) 57 840 secondes (16 heures)  
CPLEX 3.0 (1994) 4 555 secondes (1 heure 15 minutes)  
CPLEX 5.0 (1996) 3 835 secondes (1 heure 3 minutes)  
CPLEX 6.5 (1999) 165 secondes (2 minutes 45 secondes).

Sur un ordinateur à 666 MHz il ne faut plus que 39 secondes pour résoudre ce problème. Sur ce problème précis on a divisé le temps de calcul (à machine égale) entre CPLEX 1.0 et CPLEX 7.1 par 406,4. Sur des problèmes plus grands l'amélioration a été de 1695 fois, sans compter tous les problèmes qui n'étaient pas résolubles par CPLEX 1.0. A ces progrès algorithmiques et de programmation, il faut également ajouter les vitesses des machines. Sur un algorithme de type simplexe, l'amélioration entre 1988 et 2002 (667 MHz maintenant largement dépassé) est de 900. Sur un algorithme de type barrière l'amélioration est de 13000. Ces références datent de 2002 et la dernière version de CPLEX est maintenant CPLEX 10.0, annoncée 35% plus rapide sur des problèmes de programmation linéaire mixtes (variables entières et non entières). Les avancées de la technique et de l'algorithmique rendent donc tout à fait plausibles la résolution du problème d'allocation de trajectoires sur des grandes instances en un temps compatible avec le contexte de la régulation court terme. Il faut cependant résoudre auparavant le problème de détermination des trajectoires alternatives.

### III.2. Pourquoi la génération de colonnes ?

#### III.2.a) Pas d'ensemble de trajectoires précalculées fixe

L'un des objectifs de la régulation court terme est de pouvoir répondre à des situations d'exception telle que la baisse subite de la capacité d'un secteur du fait de mauvaises conditions météorologiques ou de difficultés techniques, par exemple. Dans ce contexte, il est important que la liberté d'action du régulateur court terme soit la plus grande possible en ce qui concerne les possibilités de reroutement notamment. De plus, la régulation s'effectue pour des avions déjà en l'air qui peuvent se situer n'importe où sur le réseau et leur position n'est pas limitée à un nombre restreint de points comme c'est le cas pour les avions au sol qui sont encore aux aéroports. Pour ces raisons, il n'est pas envisageable de restreindre les reroutements possibles à un ensemble de trajectoires aériennes précalculées.

#### III.2.b) Nécessité d'une modélisation fine des trajectoires

Les trajectoires alternatives doivent prendre en compte la possibilité de changer de niveau et respecter toutes les contraintes listées précédemment (routes aériennes, performances, confort...) et cela demande une modélisation fine du réseau et une description des trajectoires en 4 dimensions. De même, les dates d'entrée secteurs correspondant à une trajectoire doivent être connues avec la plus grande précision possible afin de remplir les objectifs de respect des contraintes de capacité et de lissage de la charge. Etant donné toutes ces contraintes et le grand degré de liberté laissé aux trajectoires, leur énumération complète est impossible. Il convient donc de limiter l'ensemble des trajectoires parmi lesquelles on cherche des trajectoires alternatives, sans pour autant en oublier de potentiellement intéressantes.

#### III.2.c) La génération de colonnes

La méthode de génération de colonnes est une technique itérative pour résoudre des problèmes linéaires (classiques ou mixtes). La principale caractéristique de cette méthode est de ne pas considérer explicitement toutes les variables de décision possibles. Elle est décrite précisément dans le chapitre suivant (chapitre III).

La modélisation la plus naturelle du problème d'allocation des trajectoires consiste à attribuer à chaque trajectoire une variable binaire de décision. La méthode de la génération de colonnes va alors permettre de limiter le nombre de trajectoires considérées, évitant une coûteuse énumération de toutes les trajectoires possibles. Cette méthode permet de combiner l'utilisation des ressources prometteuses de la programmation linéaire et la description fine d'un nombre restreint de trajectoires. Elle permet de compenser le handicap apporté par la grande finesse de modélisation nécessaire. C'est une méthode itérative qui, menée à son terme, conduit à une solution optimale. Elle fournit de plus généralement une solution réalisable intermédiaire si elle est stoppée avant la fin, solution dont il est possible d'évaluer la qualité. C'est un point intéressant dans notre contexte global qui nous oblige à trouver une solution réalisable en un temps réduit. Pour toutes ces raisons, nous avons décidé de modéliser le problème par un programme linéaire mixte et la résolution s'effectuera à partir de la méthode de génération de colonnes. Il est intéressant de noter que dans des travaux très récents consacrés à la formulation d'un modèle général de planification du système aérien ([1]) à horizon tactique ou stratégique, Sherali et al. proposent de calculer les plans de vol alternatifs par la méthode de génération de colonnes, sans toutefois l'implémenter puisque leur travail est focalisé sur la détermination des paramètres globaux du modèle.

Le chapitre suivant est consacré à la description de la modélisation mathématique du problème par un programme linéaire mixte et de l'architecture du processus de résolution basé sur la génération de colonnes.



# chapitre III Résolution du problème d'optimisation principal par la méthode de génération de colonnes

*Je me suis bâti sur une colonne absente. (Henri MICHAUX)*

*On présente dans la première partie la méthode de génération de colonnes pour résoudre des programmes linéaires classiques ou mixtes (à variables de décision entières et réelles) ainsi que les notations utilisées dans la suite de ce document. Dans la seconde partie, on écrit le problème d'optimisation principal du régulateur court terme à partir des choix de modélisation expliqués dans le chapitre précédent et on décrit le processus ayant conduit au programme linéaire mixte représentant ce problème d'optimisation principal. Ses différentes composantes sont expliquées et on montre que ce problème est NP-difficile. La troisième partie est consacrée à la description de la méthode de résolution globale du problème à partir de la combinaison d'un algorithme de génération de colonnes et de la méthode de branch-and-bound, combinaison appelée branch-and-price. Les liens entre les résolutions de plusieurs instances successives du problème d'optimisation, c'est-à-dire à des pas de régulation successifs, sont mis en lumière. Enfin, l'initialisation du processus et le choix des paramètres de l'algorithme de branch-and-bound (règles de branchement, évaluation de bornes inférieures, heuristique d'arrondi) sont décrits dans la troisième partie, le chapitre IV étant dévolu au sous problème de tarification.*

## **I. Notions générales sur la génération de colonnes et notations utilisées**

### **I.1. La méthode de génération de colonnes**

#### **I.1.a) Historique de la méthode de génération de colonnes**

En 1958, Ford et Fulkerson ([67]) suggérèrent de résoudre un problème de multi-flots sans en énumérer explicitement toutes les variables. En 1960, Dantzig et Wolf ([47]) développèrent cette idée en décrivant un processus ajoutant au fur et à mesure les colonnes nécessaires à la résolution d'un programme linéaire. Enfin, en 1961, cette technique que l'on allait appeler génération de colonnes était utilisée pour la première fois par Gilmore et Gomory ([45],[46]) pour résoudre un problème de découpe (Cutting Stock Problem), qui reste le problème de référence de cette technique. La génération de colonnes a depuis permis de résoudre un grand nombre de programmes linéaires comportant un nombre important de variables. L'extension de la technique à la résolution des programmes linéaires en nombres entiers, initiée par Desrosiers et al. ([43]) qui combinèrent la technique de génération de colonnes à un processus de branch-and-bound, a encore accru la popularité de cette méthode qui a depuis été employée avec succès dans un grand nombre de champs d'application : problèmes de routage, de découpes, de coloration de graphes, de détermination de plannings, etc. (voir Lübbecke et Desrosiers [37] ou Wilhelm [39] pour des revues récentes des applications de la génération de colonnes). D'une manière générale, la méthode de la génération de colonnes est d'ailleurs très efficace pour résoudre des programmes linéaires en nombres entiers.

### I.1.b) Principe de la génération de colonnes

La génération de colonnes est initialement une méthode de résolution des programmes linéaires de grande taille. Si le programme linéaire admet une solution optimale, il en existe une dans laquelle un grand nombre de variables ont une valeur nulle (les variables hors base de l'algorithme du simplexe). Le principe général de la génération de colonnes est alors de ne pas énumérer explicitement toutes les variables, mais de travailler à partir de leurs coûts réduits pour ne garder que des variables potentiellement intéressantes. Ainsi, lorsque la méthode du simplexe recherche à chaque itération une variable de coût réduit négatif pour la faire entrer dans la base, ce qui améliorera la valeur de la fonction objectif, la génération de colonnes va de la même manière chercher des variables de coût réduit négatif à chaque itération, mais sans énumérer explicitement ces variables. La structure de l'algorithme est la suivante.

#### i) *Résolution de programmes linéaires*

On cherche à résoudre un programme linéaire appelé *problème maître* (Master Problem), on résout tout d'abord un programme linéaire obtenu en ne considérant qu'une restriction des variables, programme appelé *problème maître restreint* (Restricted Master Problem ou RMP). Cette résolution permet d'obtenir les valeurs prises à l'optimal par les variables du problème dual du problème maître restreint. Ces valeurs associées à chaque contrainte du problème maître restreint sont appelées valeurs duales. Un sous problème nommé *sous problème de tarification* (pricing subproblem) ou *oracle* va alors chercher à partir de ces valeurs duales, une ou plusieurs variables qui ne sont pas encore dans le sous ensemble de variables considérées et qui ont un coût réduit négatif. Ces variables sont ajoutées au RMP qui est résolu à nouveau. C'est un processus itératif qui continue tant que le sous problème trouve des nouvelles variables à coût réduit négatif. Lorsqu'il n'en trouve aucune nouvelle, la dernière solution optimale du RMP est la solution optimale du problème maître. La génération de colonnes menée jusqu'à son terme est donc une méthode exacte.

#### ii) *Résolution de programmes linéaires en nombres entiers*

La méthode peut être combinée à un processus de branch-and-bound pour résoudre un programme linéaire en nombres entiers. La méthode branch-and-bound classique consiste à résoudre une succession de programmes linéaires obtenus en ajoutant progressivement des contraintes au problème initial de manière à obtenir une solution comportant de moins en moins de valeurs non entières. Chaque sous programme linéaire peut alors être résolu par génération de colonnes en prenant soin de bien effectuer à chaque nœud de l'arbre de branch-and-bound le processus de génération de colonnes de manière à ne pas oublier de variables potentiellement intéressantes.

## **II. Modélisation du problème maître par un PLNE**

### II.1. Formulation mathématique du problème d'optimisation

#### II.1.a) Notations

Les notations utilisées dans ce document sont rassemblées dans le Tableau 2.

NOTATIONS	Symbole
<b>Ensembles et éléments</b>	
Ensemble des vols commandables, un vol de cet ensemble	$\mathcal{F}, f$
Ensemble des éléments pour lesquels on assure le respect de la capacité (secteurs et aéroports), un élément	$\mathcal{S}, s$
Ensemble des périodes de temps pour la définition de la capacité, une période	$\mathcal{P}, p$
Ensemble des trajectoires (4D) possibles pour le vol $f$	$\mathcal{R}_f$
Sous ensemble de $\mathcal{R}_f$ retenu dans le problème maître restreint	$\mathcal{R}'_f$
Une trajectoire	$r$
Ensemble des dates discrétisées de l'horizon de prévision, une date	$\mathcal{T}, t$
<b>Caractérisation des vols</b>	
Date d'arrivée initialement prévue pour le vol $f$	$a_f$
<b>Caractérisation des correspondances</b>	
Ensemble des vols en correspondance, un vol en correspondance	$\mathcal{C}, g$
Ensemble des correspondances du vol $f$	$\mathcal{C}_f$
Date de départ initialement prévue du vol $g$	$d_g$
Coût de retarder le vol $g$ d'une unité de temps	$c_g$
Temps nécessaire entre l'arrivée d'un vol $f$ et le départ de sa correspondance $g$	$k_{f,g}$
<b>Variables de décision</b>	
Choix de la trajectoire $r$ pour le vol $f$ (variables binaires du problème primal : 1 si la trajectoire est choisie, 0 sinon)	$x_{f,r}$
Retard pour le vol $g$ correspondance d'un vol $f$ de $\mathcal{F}$	$r_g$
Variables duales associées aux contraintes de capacité	$\pi_{s,p}$
Variables duales associées aux contraintes d'unicité de vol	$\pi_f$
Variables duales associées aux contraintes de répercussion de l'arrivée du vol $f$ sur un vol en correspondance $g$	$\pi_{f,g}$
Valeurs prises par les variables duales dans la solution optimale du problème maître restreint (valeurs duales)	$\widehat{\pi}_{s,p}, \widehat{\pi}_{f,g}, \widehat{\pi}_f$
Critère du programme linéaire	$z$
<b>Paramètres du réseau</b>	
Capacité du secteur $s$ pendant la période $p$	$C(s,p)$
<b>Caractérisation des trajectoires</b>	
Utilisation de capacité de contrôle du secteur $s$ pendant la période $p$ par le vol $f$ empruntant la trajectoire $r$	$\delta_{f,r,s,p}$
Date d'arrivée du vol $f$ par la trajectoire $r$	$a_{f,r}$
Coût global de la trajectoire $r$ du vol $f$	$c_{f,r}$
Coût réduit de la trajectoire $r$ du vol $f$	$c'_{f,r}$
Coût d'exploitation de la trajectoire $r$ du vol $f$	$c^e_{f,r}$
Coût du retard à l'arrivée de la trajectoire $r$ du vol $f$	$c^a_{f,r}$
Coût du retard à l'arrivée du vol $f$ par unité de temps de retard	$c^a_f$
<b>Matrices</b>	
Matrice des contraintes du programme maître restreint	$A$
Colonne de la matrice $A$ correspondant à la variable $x_{f,r}$	$A_{f,r}$
Colonne de la matrice $A$ correspondant à la variable $r_g$	$A_g$

Tableau 2) Liste des notations utilisées

## II.1.b) Formulation mathématique par un programme linéaire mixte

On écrit tout d'abord le problème d'optimisation correspondant au problème d'optimisation du régulateur court terme puis on décrit les transformations nécessaires pour le transformer en un programme linéaire que l'on nommera problème maître (master problem), qui est le terme utilisé dans la littérature consacrée à la méthode de génération de colonnes. Les explications générales sur le modèle sont présentées après la formulation définitive, les premières étapes servant à décrire plus particulièrement la prise en compte des coûts des retards imposés au vol en correspondance dans le modèle.

### Première formulation du problème d'optimisation

$$\text{Minimiser } z = \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in C} c_g r_g$$

Sous

$$\sum_{f \in F} \sum_{r \in \mathcal{R}_f} \delta_{f,r,s,p} x_{f,r} \leq C(s,p) \quad \forall s \in S, p \in \mathcal{P} \text{ contraintes de capacité (1)}$$

$$\sum_{r \in \mathcal{R}_f} x_{f,r} = 1 \quad \forall f \in F \text{ contraintes d'unicité du choix de trajectoire (2)}$$

$$r_g = \text{Max}\{0; \text{Max}_{f \in F / g \in C_f} (a_{f,r} + k_{f,g} - d_g, \text{ avec } r / x_{f,r} = 1)\} \text{ expression du retard du vol } g \text{ (3')}$$

$$x_{f,r} \in \{0,1\} \quad \forall f \in F, \forall r \in \mathcal{R}_f \text{ contraintes d'intégrité des variables (4)}$$

Cette formulation peut se résumer ainsi : on veut trouver pour chaque vol  $f$  une trajectoire  $r$  parmi celles possibles de manière à minimiser une fonction objectif  $z$ , comprenant la somme des coûts des trajectoires retenues (première partie de l'expression de  $z$ ) et le coût des retards des vols en correspondance (deuxième partie de l'expression  $z$ ). Il faut de plus choisir une et une seule trajectoire pour chaque vol (équations (2) et (4)) et les contraintes de capacité des secteurs doivent être respectées (équations (1)).

$r_g$  représente le retard du vol  $g$  : l'équation (3') exprime le fait que le retard du vol  $g$ , toujours positif, est le maximum des retards imposés par les vols dont  $g$  est correspondance. La fonction maximum n'étant pas linéaire, on va modifier cette première formulation de manière à ne plus l'utiliser. Pour cela, on considère maintenant les retards  $r_g$  comme des variables de décision dont on assure qu'elles seront égales à ce maximum à l'aide d'un ensemble de contraintes (notées 3'' dans la deuxième formulation ci-dessous).

## Deuxième formulation du problème d'optimisation

$$\text{Minimiser } z = \sum_{f \in \mathcal{F}} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in \mathcal{C}} c_g r_g$$

Sous

$$\sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}_f} \delta_{f,r,s,p} x_{f,r} \leq C(s,p) \quad \forall s \in \mathcal{S}, p \in \mathcal{P} \text{ contraintes de capacité (1)}$$

$$\sum_{r \in \mathcal{R}_f} x_{f,r} = 1 \quad \forall f \in \mathcal{F} \text{ contraintes d'unicité du choix de trajectoire (2)}$$

$x_{f,r} (a_{f,r} + k_{f,g}) - r_g \leq d_g \quad \forall f \in \mathcal{F} \quad \forall r \in \mathcal{R}_f \quad \forall g \in \mathcal{C}_f$   $r_g$  est supérieur au retard imposé par le vol  $f$  dont  $g$  est correspondance (3'')

$$x_{f,r} \in \{0,1\} \quad \forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f \text{ contraintes d'intégrité des variables (4)}$$

$$r_g \geq 0 \quad \forall g \in \mathcal{C} \text{ positivité des retards}$$

Etant donné que  $x_{f,r}$  est binaire l'ensemble des équations (3'') impose que les  $r_g$  soient supérieurs à  $a_{f,r} + k_{f,g} - d_g$  pour la trajectoire  $r$  effectivement retenue pour le vol  $f$  dont  $g$  est correspondance:

$$r_g \geq \text{Max}_{f \in \mathcal{F} / g \in \mathcal{C}_f} (a_{f,r} + k_{f,g} - d_g, \text{ avec } r / x_{f,r} = 1)$$

De plus comme l'on considère que les coûts des retards  $c_g$  sont strictement positifs, pour une allocation donnée des trajectoires la valeur des  $r_g$  sera minimisée dans la solution optimale et on a bien:

$$r_g = \text{Max}\{0; \text{Max}_{f \in \mathcal{F} / g \in \mathcal{C}_f} (a_{f,r} + k_{f,g} - d_g, \text{ avec } r / x_{f,r} = 1)\}$$

La formulation que l'on a obtenue est linéaire et prend en compte la propagation des retards. On va effectuer cependant une nouvelle modification directement liée à la méthode de résolution que l'on souhaite utiliser. On a vu en effet que la méthode de génération de colonnes ne considère pas explicitement toutes les variables de décision mais travaille à partir de leur coût réduit. Celui-ci s'obtient par la formule suivante :

$c'_{f,r} = c_{f,r} - \pi^t A_{f,r}$  avec  $\pi^t$  représentant la transposée du vecteur des variables duales et  $A_{f,r}$  la colonne correspondant à la variable  $x_{f,r}$ . Dans la deuxième formulation, il y a une contrainte de type 3'' pour chaque triplet {vol, trajectoire, correspondance}. Les valeurs duales correspondant aux contraintes (3'') dépendent donc du triplet { $f,r,g$ } et ne peuvent être utilisées que pour la trajectoire  $r$ . On va alors modifier les contraintes (3'') d'une manière qui ne change rien à la solution optimale du problème maître mais qui permettra d'obtenir une seule variable duale  $\pi_{f,g}$  par couple { $f,g$ } indépendamment de la trajectoire  $r$ . De cette manière dans la détermination du coût réduit d'une trajectoire, on pourra prendre en compte la valeur duale  $\widehat{\pi}_{f,g}$  qui nous donne une indication sur les conséquences du vol  $f$  sur le retard du vol  $g$  dans la dernière solution optimale du RMP. La modification effectuée consiste à regrouper dans une seule équation toutes les contraintes de propagation des retards pour toutes les trajectoires d'un vol. Les contraintes (3) de la formulation finale qui remplacent les contraintes (3'') pour un vol  $f$  et sa correspondance  $g$  s'écrivent alors :

$$\sum_{r \in \mathcal{R}_f} (a_{f,r} * x_{f,r}) - r_g \leq d_g - k_{f,g} \quad \forall f \in \mathcal{F} \quad \forall g \in \mathcal{C}_f$$

Cela ne change rien dans le cas du problème maître puisque toutes les variables  $x_{f,r}$  sont nulles exceptée une et l'équation (3) s'écrit alors:  $a_{f,r} + k_{f,g} - r_g \leq d_g \quad \forall f \in \mathcal{F} \quad \forall g \in \mathcal{C}_f$  où  $r$  est la trajectoire retenue pour le vol  $f$ . La formulation finale du problème maître est alors :



## PROBLEME MAITRE

$$\text{Minimiser } z = \sum_{f \in \mathcal{F}} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in \mathcal{C}} c_g r_g$$

Sous

$$\sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}_f} \delta_{f,r,s,p} x_{f,r} \leq C(s,p) \quad \forall s \in \mathcal{S}, p \in \mathcal{P} \text{ contraintes de capacité (1)}$$

$$\sum_{r \in \mathcal{R}_f} x_{f,r} = 1 \quad \forall f \in \mathcal{F} \text{ contraintes d'unicité du choix de trajectoire (2)}$$

$$\sum_{r \in \mathcal{R}_f} (a_{f,r} * x_{f,r}) - r_g \leq d_g - k_{f,g} \quad \forall f \in \mathcal{F} \quad \forall g \in \mathcal{C}_f \text{ contraintes de propagation des retards (3)}$$

$$x_{f,r} \in \{0,1\} \quad \forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f \text{ contraintes d'intégrité des variables (4)}$$

$$r_g \geq 0 \quad \forall g \in \mathcal{C} \text{ positivité des retards}$$

## II.2. Détails de la formulation

### II.2.a) Les variables de décision

Le problème maître que l'on cherche à résoudre est l'affectation d'une trajectoire spatio-temporelle à chaque vol de manière à respecter les contraintes de capacité. On choisit la représentation la plus naturelle en choisissant comme variables de décision principales  $x_{f,r}$  telles que :

$$x_{f,r} = 1 \text{ si le vol } f \text{ choisit la trajectoire } r \in \mathcal{R}_f \\ = 0 \text{ sinon}$$

Il y a alors autant de variables de décision  $x_{f,r}$  que de trajectoires possibles pour chaque vol. On peut remarquer le choix des variables de décision représentant le problème n'est pas unique et l'on trouve dans la littérature d'autres approches. Nous avons retenu cette formulation pour que la méthode de la génération de colonnes se traduise par une restriction des trajectoires considérées.

Les variables  $r_g$  quant à elles représentent le retard au décollage du vol  $g$  en correspondance. Elles sont à valeur réelle.

### II.2.b) Les contraintes

#### i) La propagation des retards

Comme expliqué dans le paragraphe consacré aux choix de modélisation, le premier niveau de propagation des retards a été explicitement introduit dans le modèle. On a défini pour cela un ensemble  $\mathcal{C}$  représentant l'ensemble des vols en correspondance des vols commandables. Les relations de correspondances dont il est ici question sont impératives, c'est-à-dire, que les vols de  $\mathcal{C}$  ne peuvent pas décoller tant que tous les vols dont ils sont correspondance ne sont pas arrivés.  $k_{f,g}$  représente le temps minimal nécessaire entre l'arrivée d'un vol  $f$  et le départ de sa correspondance  $g$ . Les données sur  $k_{f,g}$  n'étant pas disponibles on a utilisé un facteur  $k$  identique pour tous les couples vol-correspondance. La démarche ayant conduit à la formulation retenue des retards a été décrite ci-dessus.

## ii) *La fonction de charge des secteurs*

La fonction de charge des secteurs sera exprimée par l'indicateur  $\delta_{f,r,s,p}$  explicitant l'utilisation de la capacité du secteur  $s$  lors de la période  $p$  par le vol  $f$  empruntant la trajectoire  $r$ . On peut noter que l'indicateur  $\delta_{f,r,s,p}$  est spécialement conçu pour la définition que l'on a retenue de la capacité, c'est-à-dire en nombre de vols entrant dans un secteur pendant une période donnée. En effet, si le vol  $f$  au cours de sa trajectoire  $r$  rentre dans le secteur  $s$  pendant la période  $p$  alors  $\delta_{f,r,s,p}$  sera égal à un et il sera égal à zéro dans le cas contraire.  $\delta_{f,r,s,p}$  pourrait cependant être utilisé pour tout autre définition retenue de la charge.

### II.2.c) La fonction objectif

On a vu dans la définition du programme maître que les retards réactifs du premier ordre sont pris en compte directement dans la modélisation par l'énumération explicite de toutes les correspondances. La fonction objectif comprend donc la somme des coûts des retards réactifs du premier ordre et la somme du coût de chaque trajectoire retenue. La fonction coût d'une trajectoire doit prendre en compte les autres composantes identifiées : la composante d'exploitation et la composante de retard à l'arrivée.

#### i) *La composante d'exploitation*

La composante d'exploitation correspond au coût immédiat d'effectuer le vol. Elle devrait comprendre la consommation de carburant et l'usure de l'avion. Cependant, pour des raisons de disponibilité des données, la composante  $c_{f,r}^e$  sera restreinte à la consommation de carburant.

#### ii) *La composante retard à l'arrivée*

Le retard à l'arrivée a des conséquences directes en terme de coût et des conséquences indirectes dues aux correspondances. Les conséquences indirectes sont prises en compte dans le modèle par le biais du coût  $c_g$  de retarder le vol  $g$ . La composante  $c_{f,r}^a$  représente uniquement le coût direct du retard et dépend du nombre de passagers du vol. Le calcul de ces coefficients est expliqué dans le chapitre II paragraphe I.2.b) page 33.

#### iii) *Les retards réactifs d'ordre supérieur*

Pour prendre en compte les retards réactifs d'ordre supérieur à 1, c'est-à-dire au-delà de la première chaîne de correspondances, on joue directement sur le coefficient  $c_g$ , c'est-à-dire le coût de retarder un vol en correspondance, en augmentant la valeur de celui-ci lorsque son départ est le matin par exemple.

## II.3. Complexité du problème maître

### II.3.a) Le problème GAP

Le problème maître peut être considéré comme une version « sophistiquée » d'un problème bien étudié dans le domaine de l'ordonnancement : le *problème d'allocation généralisé* ou *Generalized Assignment Problem* (GAP). L'objectif de ce problème est d'affecter un certain nombre de tâches à un certain nombre de machines tout en respectant les contraintes de capacité des machines et en minimisant le coût de l'ordonnancement. La formulation classique du GAP est la suivante :

Soit  $I=\{1\dots m\}$  un ensemble de machines et  $J=\{1\dots n\}$  un ensemble de tâches. Chaque tâche  $j$  allouée à une machine  $i$  consomme  $a_{i,j}$  unités de capacité de la machine  $i$  pour un coût de  $c_{i,j}$ . Chaque machine a une capacité de  $b_i$ . Les variables de décision binaires  $x_{i,j}$  prennent la valeur 1 si la tâche  $j$  est allouée à la machine  $i$ , 0 sinon. Le problème s'écrit alors :

$$\text{Min } \sum_{i=1}^m \sum_{j=1}^n c_{i,j} * x_{i,j}$$

Sous

$$\sum_{j=1}^n a_{i,j} x_{i,j} \leq b_i \quad \forall i \in I$$

$$\sum_{i=1}^m x_{i,j} = 1 \quad \forall j \in J$$

$$\text{et } x_{i,j} \in \{0,1\} \quad \forall i \in I; \forall j \in J$$

Le problème de décision associé au problème GAP s'énonce ainsi. Soit  $K$  une valeur arbitraire, existe-t-il un vecteur des variables de décision  $X_{i,j}$  vérifiant les contraintes de GAP et tel que  $c_{i,j} X_{i,j} \leq K$  ? Le problème GAP est un problème NP-difficile ([66]) et le problème de décision associé est NP-complet. Une réduction du problème de décision associé vers le problème de décision associé au problème maître suffit donc à démontrer que le problème maître est NP-difficile.

### II.3.b) Réduction du GAP vers le problème maître.

A partir d'une instance du problème de décision associé à GAP, on construit facilement une instance du problème de décision associé au problème maître du module d'optimisation du régulateur court terme.

On pose  $\mathcal{F}=J$  (une tâche représente un vol),  $S=I$  (une machine représente un secteur),  $P=\{1\}$  (une seule période),  $C=\emptyset$  (pas de correspondances). Les trajectoires possibles pour le vol  $f$  consistent alors à survoler un seul secteur (il y a donc  $m$  trajectoires possibles par vol) et le coût d'une telle trajectoire est  $c_{i,j}$  où  $j$  est la tâche associée au vol  $f$  et  $i$  la machine correspondant au secteur traversé par la trajectoire. Le paramètre  $K$  est alors le même dans les deux problèmes de décision associés.

On a alors  $\delta_{f,r,s,p}=0$  si la trajectoire  $r$  ne correspond pas au survol du secteur  $s$   
 $=a_{i,j}$  si la trajectoire  $r$  correspond au survol du secteur  $s$ .  $f$  est le vol associé à la tâche  $j$  et  $i$  la machine associée au secteur  $s$ . La transformation effectuée est polynomiale.

Il est clair que l'instance du problème GAP initiale admet une solution si et seulement si l'instance obtenue par réduction admet une solution. On en conclut que le problème de décision associé au problème maître est NP-complet et que le problème maître est donc NP-difficile.

Le paragraphe suivant est consacré à la description du processus de résolution de ce programme linéaire mixte à partir de la méthode de génération de colonnes couplée à un algorithme de branch-and-bound, méthode appelée branch-and-price dans la littérature spécialisée.

### **III. Architecture du système de résolution**

#### III.1. Architecture globale du processus de résolution

##### III.1.a) Description générale

Résoudre le problème consiste pour chaque vol à choisir une trajectoire (en 4 dimensions) parmi un ensemble de trajectoires possibles. Chaque trajectoire correspond à une variable de décision binaire dans le modèle en programmation linéaire (1 si l'on choisit la trajectoire, 0 sinon). On relaxe tout d'abord les contraintes d'intégrité des variables qui deviennent alors réelles et comprises dans l'intervalle  $[0,1]$  et l'on résout le problème en ne considérant qu'un nombre limité de variables de décision (c'est-à-dire de trajectoires). En plus de la solution optimale de ce problème, on obtient les valeurs duales associées aux contraintes du programme linéaire primal. Ces valeurs duales qui, schématiquement, indiquent le coût marginal d'utilisation des ressources de contrôle, permettent alors de calculer par un sous problème de nouvelles variables (trajectoires) dont l'ajout au problème initial provoquera une amélioration de la solution obtenue. Les valeurs duales ayant alors changé, on peut de nouveau résoudre le sous problème pour trouver de nouvelles trajectoires intéressantes. Si de telles trajectoires existent, on les ajoute de nouveau au problème et une nouvelle boucle commence, dans le cas contraire on arrête le processus qui a alors fourni une solution optimale au programme linéaire relaxé, solution qui sera la base de départ de la recherche d'une solution réalisable entière. Si la solution obtenue n'est pas entière, on ajoute une nouvelle coupe (processus de branchement) de manière à exclure la solution courante non entière et l'on résout les sous problèmes ainsi obtenus par la même méthode de génération de colonnes. Au cours du processus de résolution des programmes linéaires relaxés par génération de colonnes, on peut obtenir des bornes inférieures à la solution optimale. Ces bornes sont utilisées pour couper prématurément des branches de l'arbre de branch-and-bound sans avoir besoin de résoudre à l'optimalité chaque programme linéaire correspondant à une feuille de l'arbre de branchement.

##### III.1.b) Articulation des différents modules et schéma récapitulatif

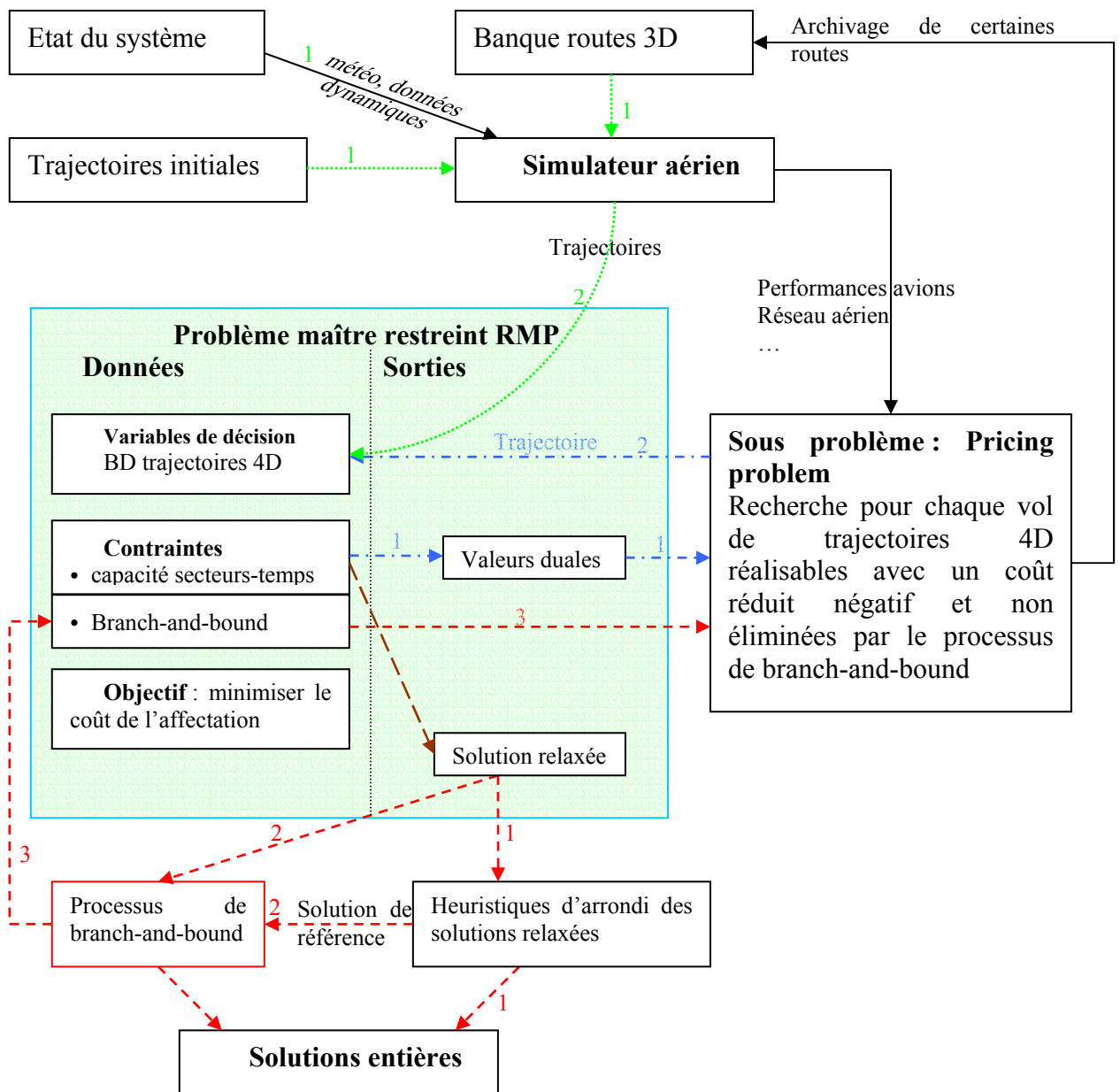
###### *i) Résolution du problème relaxé*

L'architecture du système de résolution est donc basée sur cette boucle : résolution d'un problème maître restreint relaxé, puis à partir des valeurs duales obtenues qui nous indiquent le coût marginal d'utilisation des ressources de capacité, résolution d'un sous problème pour trouver les nouvelles trajectoires à ajouter, si elles existent. Cette boucle peut s'arrêter plus tôt si la borne inférieure du programme linéaire en cours est supérieure à la valeur de la meilleure solution entière obtenue jusqu'à présent. Etant donné que l'on souhaite obtenir des trajectoires directement utilisables par les vols, le sous problème doit disposer de données fournies par un simulateur aérien sur les performances des avions, le réseau aérien et les exigences pour déclarer une trajectoire admissible. C'est un des rôles du simulateur aérien inclus dans le processus de résolution. La résolution du problème global devant être effectuée fréquemment, les enseignements tirés des précédentes résolutions devront être capitalisés afin de faciliter l'obtention d'une nouvelle solution réalisable : c'est le rôle de la banque de données qui garde en mémoire un certain nombre de routes aériennes (en 2 dimensions, avec éventuellement un niveau de vol de référence), routes à partir desquelles on pourra obtenir avec l'aide du simulateur aérien de nouvelles trajectoires (en 4 dimensions, 4D) utilisables lors de l'initialisation d'une nouvelle instance du problème.

*ii) Processus d'obtention d'une solution entière*

Lorsque l'on ne trouve plus de trajectoires intéressantes, on démarre de la solution du problème relaxé pour trouver une ou plusieurs solutions entières, tout d'abord à l'aide d'heuristiques d'arrondi, puis par un processus de branch-and-bound si les résultats obtenus ne satisfont pas à des exigences de garantie d'approximation d'optimalité prédéfinies. Les heuristiques d'arrondi ont également pour but de donner une solution réalisable initiale de référence au processus de branch-and-bound afin de couper plus facilement des branches de l'arbre de décision généré par cette méthode.

Les différents modules du processus de résolution et leurs articulations sont détaillés dans le schéma de la Figure 12 explicitant les trois phases de la méthode : l'initialisation du premier ensemble de trajectoires considérées dans le problème restreint, la génération de colonnes (c'est-à-dire de trajectoires) par le sous problème de tarification et l'obtention d'une solution réalisable entière par un processus de branch-and-bound couplé avec des heuristiques d'arrondi. Chaque phase est détaillée dans la partie suivante (IV).



- Programme linéaire relaxé**
- ⋯→ *Processus d'initialisation*
  - - - -> *Première phase du processus de génération de colonnes : boucle tant qu'il y a des nouvelles trajectoires*
  - - -> *Deuxième phase du processus de génération de colonnes : il n'y a plus de trajectoires nouvelles, la solution relaxée est optimale pour le RMP*
  - - -> *Obtention d'une solution entière: heuristiques d'arrondi de la solution relaxée et branch-and-bound : ajout de contraintes pour se rapprocher d'une solution entière ; résolution du nouveau problème par génération de colonnes. ( - - - -> )*

Figure 12) Architecture détaillée du processus de résolution

## IV. Formulation des différents modules de résolution

### IV.1. Le problème maître restreint (RMP)

#### IV.1.a) Formulation et résolution du RMP

On restreint le problème en ne considérant plus qu'un sous ensemble des trajectoires possibles pour le vol  $f$  et en relaxant les contraintes (4) d'intégrité des variables de décision  $x_{f,r}$ . On note  $\mathcal{R}'_f$  l'ensemble des trajectoires possibles considérées dans le problème maître restreint pour le vol  $f$ . Le programme maître restreint (RMP pour Restricted Master Problem) s'écrit alors :

#### PROBLEME MAITRE RESTREINT

$$\text{Minimiser } \sum_{f \in \mathcal{F}} \left( \sum_{r \in \mathcal{R}'_f} c_{f,r} x_{f,r} \right) + \sum_{g \in \mathcal{C}} c_g r_g$$

Sous

$$\sum_{f \in \mathcal{F}} \sum_{r \in \mathcal{R}'_f} \delta_{f,r,s,p} x_{f,r} \leq C(s,p) \quad \forall s \in \mathcal{S}, p \in \mathcal{P} \quad (1)$$

$$\sum_{r \in \mathcal{R}'_f} x_{f,r} = 1 \quad \forall f \in \mathcal{F} \quad (2)$$

$$\sum_{r \in \mathcal{R}'_f} a_{f,r} * x_{f,r} - r_g \leq d_g - k_{f,g} \quad \forall f \in \mathcal{F} \quad \forall g \in \mathcal{C}_f \quad (3)$$

$$x_{f,r} \geq 0 \quad \forall f \in \mathcal{F}, \forall r \in \mathcal{R}'_f \quad (4')$$

$$r_g \geq 0 \quad \forall g \in \mathcal{C}$$

On remarque que la contrainte (4') relaxée devrait s'écrire  $0 \leq x_{f,r} \leq 1$ . La contrainte  $x_{f,r} \leq 1$  est superflue du fait de la contrainte d'unicité de trajectoire par vol (2) et de la positivité des variables de  $x_{f,r}$ . On ne la conserve pas dans la formulation du problème maître restreint car, si elle n'a pas d'incidence sur le problème primal, elle peut en avoir sur le problème dual puisqu'il faudrait rajouter la variable duale associée à cette contrainte, variable qui pourrait être non nulle dans la solution optimale du dual.

#### IV.1.b) Les valeurs duales

On définit alors les familles de variables duales associées aux contraintes du problème maître restreint:

$\pi_{s,p}$  associée à la contrainte (1) de capacité sur le secteur  $s$  pendant la période  $p$ .

$\pi_f$  associé à la contrainte (2) d'unicité de trajectoire pour le vol  $f$ .

$\pi_{f,g}$  associée à la contrainte (3) de propagation du retard du vol  $f$  au vol  $g$  en correspondance.

On peut maintenant écrire le programme dual associé au problème maître restreint.

## PROGRAMME DUAL ASSOCIE AU PROBLEME MAITRE RESTREINT

$$\text{Maximiser } \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} C(s,p) * \pi_{s,p} + \sum_{f \in \mathcal{F}} \pi_f + \sum_{f \in \mathcal{F}} \sum_{g \in C_f} (d_g - k_{f,g}) \pi_{f,g}$$

Sous

$$\sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \pi_{s,p} + \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,g} \leq c_{f,r} \quad \forall f \in \mathcal{F}, r \in \mathcal{R}'_f$$

$$\sum_{f/g \in C_f} -\pi_{f,g} \leq c_g \quad \forall g \in C$$

$$\pi_{s,p} \leq 0 \quad \forall s \in \mathcal{S}, p \in \mathcal{P}$$

$$\pi_{f,g} \leq 0 \quad \forall f \in \mathcal{F} \quad \forall g \in C_f$$

On peut noter qu'à partir du moment où les coefficients  $c_{f,r}$  sont strictement positifs, on peut remplacer la contrainte (2) du problème primal par  $\sum_{r \in \mathcal{R}'_f} x_{f,r} \geq 1 \quad \forall f \in \mathcal{F}$  sans changer la

solution optimale puisque l'objectif poursuivi est de minimiser le coût total. On peut en déduire que dans la solution optimale duale la variable  $\pi_f$  sera positive ou nulle. Les valeurs  $\widehat{\pi}_{s,p}$  prises par les variables duales  $\pi_{s,p}$  dans la solution optimale duale permettent de quantifier le coût d'utilisation de la ressource de contrôle du secteur  $s$  pendant la période  $p$  au voisinage de la solution actuelle. Les valeurs  $\widehat{\pi}_{f,g}$  prises par les variables duales  $\pi_{f,g}$  dans la solution optimale permettent quant à elles de quantifier les conséquences du retard à l'arrivée du vol  $f$  sur le vol  $g$ .

### IV.1.c) Résolution du RMP

Le problème maître restreint est un programme linéaire classique qui peut devenir de grande taille mais ne présente pas de difficultés particulières. La résolution de ce problème se fera dès lors à l'aide de logiciels spécialisés dans la résolution de programmation linéaire, capables de résoudre de très grands programmes en un temps très restreint (voir le chapitre II paragraphe III.1.c). Ces logiciels permettent d'accéder directement aux valeurs duales nécessaires à la résolution du sous problème de tarification. De plus, dans le cas de l'utilisation de l'algorithme du simplexe primal et dual, on peut garder en mémoire la dernière base obtenue lors de la résolution du problème maître restreint : cette base sert de départ à la résolution avec les nouvelles variables ajoutées par le processus de génération de colonnes, ce qui présente l'avantage d'accélérer la résolution.

## IV.2. Le processus d'initialisation du RMP

### IV.2.a) Objectifs du processus d'initialisation

L'initialisation du processus de génération de colonnes doit répondre à deux exigences complémentaires.

Tout d'abord, pour pouvoir obtenir des valeurs duales, le problème maître restreint initial doit comporter une solution réalisable. Cette exigence est fondamentale pour pouvoir amorcer le processus. Ensuite, l'objectif de cette initialisation est d'obtenir une « bonne » solution réalisable dès la première étape afin de minimiser à la fois le nombre de trajectoires



potentiellement intéressantes dans le sous problème de tarification et le nombre d'itérations nécessaires à l'obtention d'une solution optimale.

#### IV.2.b) Trajectoires remarquables initiales

Il y a une famille de trajectoires qui doivent faire partie par défaut de la banque de trajectoires du problème maître restreint initial : la trajectoire courante de chaque vol. Elle est obtenue à partir du plan de vol de chaque avion, daté grâce au simulateur aérien. Les autres trajectoires que l'on ajoutera devront répondre à l'un ou l'autre des objectifs énoncés de l'initialisation : trouver une solution réalisable et limiter le nombre d'itérations de la génération de colonnes.

Dans un premier temps, afin de favoriser à peu de frais calculatoires la diversité des trajectoires et ainsi la possibilité de trouver une solution réalisable, on dérive de chaque trajectoire initiale les trajectoires correspondant à un certain nombre de boucles d'attente au premier point survolé par la trajectoire où l'attente en l'air est possible. Le nombre maximal de boucles d'attente est fonction de la réserve de carburant de chaque avion. Enfin si l'on raisonne d'un point de vue mise en œuvre opérationnelle, on peut ajouter des trajectoires issues des routes standard fournies par la CFMU pour les principales paires origine-destination.

#### IV.2.c) Banque de routes

##### *i) Le principe de la banque de routes*

La résolution du problème maître s'inscrit dans le contexte de la commande prédictive à horizon glissant ce qui implique de recalculer à chaque pas de régulation une nouvelle solution en fonction des nouvelles données disponibles. Afin de bénéficier des calculs effectués lors des résolutions précédentes, une banque de routes garde en mémoire les routes les plus intéressantes calculées à une étape donnée de la résolution afin de servir à l'initialisation de  $\mathcal{R}'_f$  lors de la résolution du problème courant. Le travail effectué lors de cette thèse, notamment d'un point de vue programmation, s'est focalisé sur la résolution du problème d'optimisation à un pas de régulation fixé et l'intégration informatique d'un régulateur court terme utilisant les résultats obtenus pour modifier une simulation du trafic n'a pas été effectuée. Aussi il n'est pas possible à ce stade de définir précisément les règles de fonctionnement de la banque de données de trajectoires mais une réflexion sommaire a été menée sur les grands principes sur lesquels elle se fonde.

##### *ii) Fonctionnement de la banque de routes*

Au cours de la résolution d'une instance du problème d'optimisation, le sous problème de tarification va calculer pour chaque avion des trajectoires 4 dimensions qui seront ajoutées à la banque de données de trajectoires. Le parcours temporel de la trajectoire 3 dimensions dépend du type et de la charge de l'avion et des conditions météo. Lors d'un pas de temps ultérieur il est peu probable que tous ces paramètres soient les mêmes que ceux lors du calcul de la trajectoire en 4 dimensions, garder en mémoire cette trajectoire telle quelle n'a donc que peu d'intérêt puisqu'il n'y aura que très peu de cas où on pourra l'utiliser. Par contre, si l'on ne considère que l'origine (c'est-à-dire le point où se situe l'avion au moment du calcul de la trajectoire, ramené en fait à un point discret de notre modélisation du réseau aérien : une balise ou un point d'entrée secteur par exemple) et la destination (un aéroport), il y aura dans les pas de résolution ultérieurs une forte probabilité que l'on cherche une trajectoire pour un

avion ayant la même origine et la même destination. Le principe de la banque de routes est donc d'extraire de certaines trajectoires 4D intéressantes la route simplifiée (successions de balises et un niveau de vol de référence) et de la stocker. Lorsque l'on retrouvera un avion ayant la même origine (ou survolant un point de la trajectoire) et la même destination, on pourra utiliser le simulateur aérien et les données actualisées sur le réseau et sur le vol pour en déduire une (ou plusieurs) trajectoire 4D réalisable par le vol.

Une fois ce principe posé, il faut trouver le juste équilibre pour avoir une banque de routes suffisamment riche afin de répondre aux objectifs de l'initialisation (obtention d'une solution réalisable, limitation du nombre d'itérations de générations de colonnes) tout en limitant sa taille pour des raisons de capacité de stockage. Une banque comportant trop de trajectoires pourrait alourdir la résolution du problème restreint en lui fournissant trop de variables. Il faut donc définir un critère d'entrée dans la banque et éventuellement un critère de sortie. On peut définir un critère d'entrée de la manière suivante : on garde toutes les routes correspondant aux trajectoires ayant eu une valeur non nulle (ou supérieure à un nombre fixé) dans une solution optimale intermédiaire du problème maître restreint. Le critère de sortie non indispensable est à définir plus expérimentalement : par exemple sortir une route qui n'a pas été utilisée dans une solution optimale du problème complet depuis plus d'une semaine ou d'un mois.

#### IV.2.d) Trajectoires artificielles ou technique du « grand M »

La présence d'une solution réalisable pour le premier problème maître restreint est une condition indispensable à la poursuite de la résolution. La technique dite du « grand M » (où M est le coût d'une trajectoire artificielle ajoutée au problème) consiste à créer pour chaque vol une trajectoire artificielle permettant de rallier la destination sans utilisation de la capacité de contrôle mais avec un coût arbitrairement grand. Ainsi, on est assuré de trouver une solution réalisable quelles que soient la capacité du réseau et la demande de trafic. Les itérations de la génération de colonnes élimineront alors si possible de la solution les trajectoires artificielles, puisque d'un coût trop élevé par rapport aux trajectoires réelles. On peut remarquer aussi que du fait de ce coût très élevé, la solution optimale de la première itération favorisera les trajectoires réelles tant que les contraintes de capacité ne seront pas violées : les valeurs duales obtenues seront alors pertinentes et permettront de lancer le processus de génération de colonnes efficacement. Si l'on a choisi un coût artificiel suffisamment important et que la solution comporte encore une proportion de trajectoires artificielles, alors le problème posé n'a pas de solutions réalisables mais on dispose d'informations précieuses : les secteurs-périodes dont la saturation est critique et les vols auxquels on n'a pas réussi à allouer une trajectoire réalisable sans dépassement de la capacité. Les trajectoires artificielles resteront dans le problème maître restreint à tous les nœuds du processus de branch-and-bound.

### IV.3. Le sous problème de tarification

#### IV.3.a) Nature du sous problème de tarification

Le sous problème de tarification, ou oracle, est un élément primordial du processus global de résolution. Son rôle premier est de trouver une variable de coût réduit négatif à ajouter au problème maître restreint. Dans notre contexte, les variables correspondent aux trajectoires des avions. Formellement le sous problème est dès lors un problème similaire aux problèmes de plus court chemin avec contraintes : trouver un chemin de coût réduit inférieur à une borne (zéro) tout en respectant la contrainte de réserve de carburant. Cependant, les contraintes

supplémentaires sur la faisabilité de la trajectoire (performances des avions, règles de navigation aériennes, confort des passagers, modélisation fine du réseau aérien...) conduisent à une modélisation complexe empêchant une utilisation telle quelle des algorithmes de plus court chemin.

#### IV.3.b) Objectifs du sous problème de tarification

Comme on l'a vu, dans le cadre théorique de la génération de colonnes, le rôle du sous problème de tarification est de trouver une ou plusieurs variables de coût réduit négatif à ajouter au problème maître restreint. Généralement, le sous problème fournit même la variable avec le plus petit coût réduit mais cela n'est pas indispensable à l'algorithme pour trouver la solution optimale : cela modifie principalement la convergence de l'algorithme global de génération de colonnes. Dans notre cas, il y a un sous problème de tarification par vol. Ces différents sous problèmes sont indépendants les uns des autres dès que les variables duales sont fixées. Les objectifs assignés à chacun de ces sous problèmes seront alors : trouver un ensemble de trajectoires de coût réduit négatif (car on peut obtenir un ensemble de trajectoires pour un effort de calcul supplémentaire limité par rapport à l'obtention d'une seule trajectoire) ; trouver une borne inférieure à la valeur du coût réduit des trajectoires, car la connaissance de l'ensemble des bornes pour chaque vol permettra de calculer une borne inférieure pour la fonction objectif du programme maître restreint ; prendre en compte les contraintes liées à l'ajout de coupes lors du processus de branch-and-bound qui est décrit ci-dessous. La modélisation adoptée et les algorithmes développés sont décrits dans le chapitre IV entièrement consacré à la définition et à la résolution du sous problème de tarification.

## **V. Le processus d'obtention d'une solution entière**

### V.1. La méthode branch-and-price dans le cas général

#### V.1.a) Les algorithmes de séparation et évaluation progressive

Les algorithmes de la famille séparation et évaluation progressive (SEP ou branch-and-bound) ont pour but de résoudre des problèmes généraux du type : minimiser  $\{f(x) ; x \in X\}$  où  $X$  est un ensemble discret généralement très grand. Le principe général de la méthode consiste à partitionner l'ensemble  $X$  en sous ensembles  $X_1, X_2$  etc. pour lesquels on dispose d'une fonction d'évaluation minorante  $\varphi$  telle que  $\varphi(X_i) \leq \min \{f(x), x \in X_i\}$ . Si l'on dispose d'un élément  $x$  de  $X_i$  tel que  $f(x) = \varphi(X_i)$  l'évaluation est dite exacte. La propriété fondatrice de la méthode est alors la suivante : si pour un sous ensemble donné la valeur de la fonction minorante est supérieure à la valeur d'une solution réalisable connue, ce sous ensemble ne contient pas de meilleure solution que celle déjà connue et on peut le supprimer d'une recherche de solution optimale. L'algorithme va donc consister à séparer l'ensemble des solutions réalisables en sous ensembles et à étudier ces sous ensembles à l'aide de la fonction d'évaluation. Quatre cas sont alors possibles :

- L'évaluation est exacte, on dispose alors de la solution optimale du sous ensemble considéré. On la compare avec la meilleure solution connue pour le moment et on garde en mémoire la meilleure des deux (ainsi que sa valeur). On n'a pas besoin de plus diviser le sous ensemble.
- L'évaluation est supérieure ou égale à la valeur de la meilleure solution connue : on arrête l'exploration du sous ensemble considéré.

- L'évaluation est strictement inférieure à la valeur de la meilleure solution connue : on divise de nouveau le sous ensemble considéré en plus petits sous ensembles et on applique la même méthode jusqu'à ce qu'il n'y ait plus de sous ensembles en attente d'être examinés.
- Le sous ensemble est vide : on arrête l'exploration

Si le principe général de l'algorithme est simple, il existe beaucoup de latitude dans sa mise en œuvre et notamment dans le choix de la partition et dans le choix de l'ordre d'exploration des différents sous ensembles, ces choix conditionnent totalement son efficacité.

#### V.1.b) Résolution de programmes linéaires en nombres entiers par branch-and-bound

Les programmes linéaires mixtes en nombres entiers (où certaines variables sont réelles tandis que d'autres sont entières) sont un champ d'application privilégié de la méthode de branch-and-bound. On dispose en effet dans ce cas d'une bonne fonction minorante : la résolution du programme linéaire dans lequel on a relaxé les contraintes d'intégrité, type de problèmes pour lesquels on dispose d'algorithmes efficaces (simplexe primal-dual, méthodes de points intérieurs...). Cela fournit une borne inférieure car toutes les solutions réalisables du problème avec les contraintes d'intégrité sont solutions réalisables du problème sans les contraintes d'intégrité. La valeur de la solution optimale de ce dernier problème est donc une borne inférieure du problème avec les contraintes d'intégrité. Plus formellement (voir Figure 13) :

Soit un programme linéaire en nombres entiers  $P$ . On note  $\hat{P}$  le programme linéaire relaxé associé et  $\tilde{P}$  l'ensemble des solutions réalisables de  $P$ .  $\overline{z_{opt}}$  est la valeur de la meilleure solution de  $P$  actuellement connue (obtenue par exemple par une heuristique d'arrondi avant le début du processus de branch-and-bound). On sépare  $P$  en ajoutant une inéquation linéaire de manière à obtenir deux sous problèmes  $A$  et  $B$  dont la réunion des ensembles de solutions réalisables est égale à  $\tilde{P}$  :  $\tilde{A} \cup \tilde{B} = \tilde{P}$ . On dispose d'une fonction minorant de  $\tilde{A}$  et de  $\tilde{B}$  : la résolution du problème associé relaxé de ses contraintes d'intégrité :  $\hat{A}$  et  $\hat{B}$ . Si cette résolution fournit une solution réalisable (entière) de  $P$  et meilleure que  $\overline{z_{opt}}$  on actualise celui-ci. Une solution est dite fractionnaire si certaines variables ont des valeurs non entières. Il existe alors trois conditions d'arrêt au processus d'ajouts de coupes à partir des sous problèmes :

- si la solution relaxée est non fractionnaire et donc réalisable
- s'il n'y a pas de solutions
- si la borne inférieure obtenue est moins bonne ou égale à la meilleure solution connue

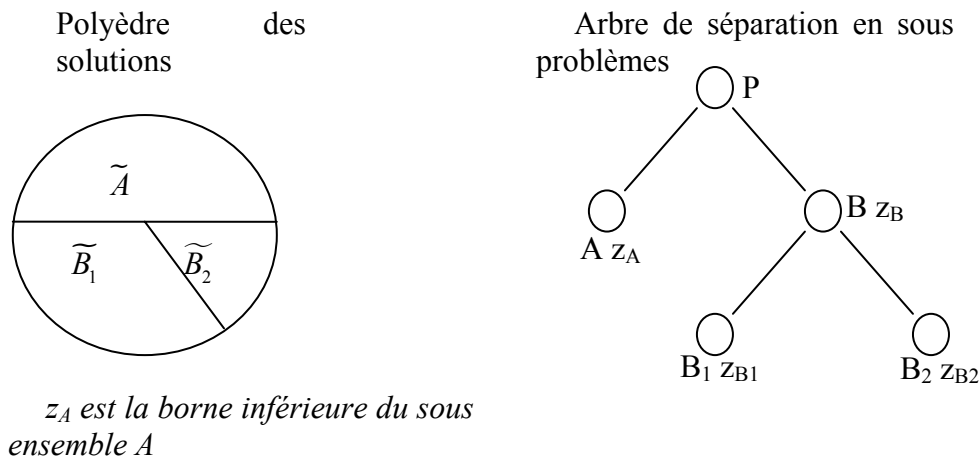
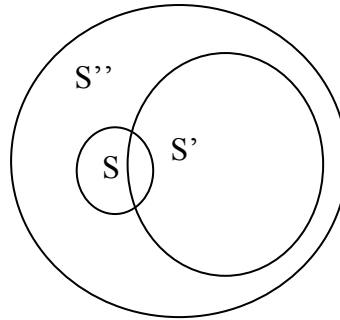


Figure 13) processus de branch-and-bound appliqué aux programmes linéaires mixtes

Ce processus mené à son terme permet de trouver la solution optimale de P.

#### V.1.c) Résolution d'un PLNE par génération de colonnes et branch-and-bound.

Dans le cas de la génération de colonnes le processus est un peu modifié du fait que la solution entière optimale du programme maître ne fait pas forcément partie de l'ensemble des solutions réalisables du programme maître restreint. Ainsi notons  $S$  l'ensemble des solutions réalisables du problème maître,  $S'$  l'ensemble des solutions réalisables du problème maître restreint relaxé et  $S''$  l'ensemble des solutions réalisables du problème maître relaxé. Dans le cas général (hors génération de colonnes) la recherche de la solution optimale s'effectue à partir de  $S''$  et on a la certitude qu'elle appartient bien à  $S''$  car  $S \subset S''$ . Mais dans le cas de la génération de colonnes, cette recherche s'effectue à partir de  $S'$  et on n'a pas de garantie au début du processus que la solution optimale soit dans  $S' \cap S$ . On sait simplement que la solution optimale de  $S''$  fait partie de  $S'$  (voir Figure 14). En résumé, des variables qui ne font pas partie de la solution optimale du problème maître relaxé et qui n'ont pas été générées par le processus de génération de colonnes peuvent faire partie de la solution optimale du problème maître non relaxé : lorsque l'on ajoute une nouvelle coupe il faut vérifier s'il n'y a pas de nouvelles variables de coût réduit négatif non incluses dans le problème maître restreint obtenu.



$S$  ensemble des solutions réalisables du problème maître (entières)  
 $S'$  ensemble des solutions réalisables du problème maître restreint relaxé  
 $S''$  ensemble de solutions réalisables du problème maître relaxé  
 $S \cap S'$  : solutions réalisables entières du problème maître restreint relaxé  
 $S \setminus S'$  solutions entières qui utilisent des trajectoires non retenues par le processus de génération de colonnes.  
 $S' \setminus S$  solutions non entières du problème maître restreint relaxé

**Figure 14) Illustration des domaines de solutions réalisables dans le cas de la génération de colonnes**

Le processus de branch-and-bound dans le cas de la génération de colonnes est alors le suivant (voir Figure 15). On suit le même schéma général que dans le processus classique mais à chaque étape, c'est-à-dire pour chaque problème intermédiaire, il faut vérifier après la résolution du programme relaxé que la solution obtenue resterait la solution optimale si l'on considérait toutes les variables. En effet des variables qui avaient été négligées dans le processus de génération de colonnes peuvent faire partie de la nouvelle solution optimale du sous problème obtenu en ajoutant des contraintes supplémentaires. Aussi faut-il vérifier grâce au sous problème de tarification qu'il n'y a pas de variables exclues du problème maître restreint qui aient un coût réduit négatif. Cette vérification doit prendre en compte les valeurs duales associées aux contraintes supplémentaires ajoutées lors du processus de branch-and-bound. S'il en existe, ces variables doivent être ajoutées à la base de données de trajectoires en quatre dimensions, notamment pour la prochaine étape du branch-and-bound dans laquelle le problème courant sera de nouveau divisé.

En fait cela revient à résoudre chaque problème intermédiaire issu de la séparation par la méthode de la génération de colonnes, l'initialisation étant alors assurée par les résultats des problèmes déjà résolus à ce stade.

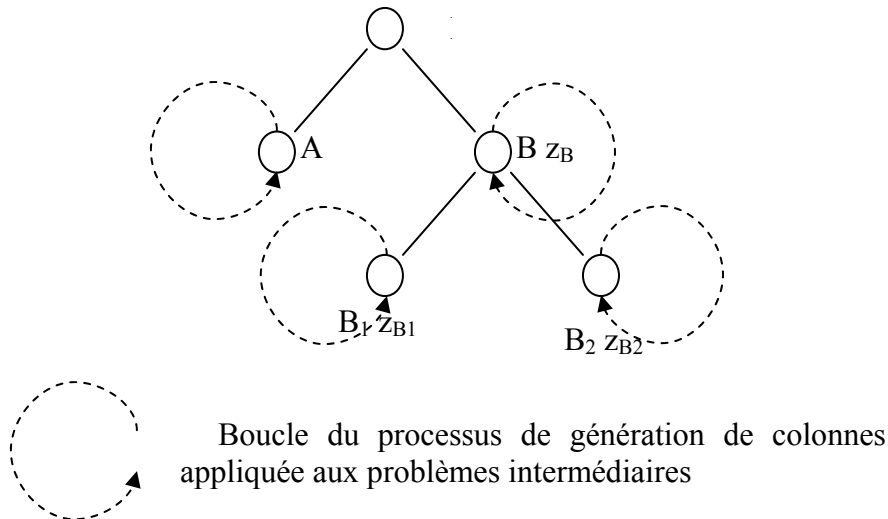


Figure 15) processus de branch-and-bound dans le cas de la génération de colonnes

## V.2. Application de la méthode de branch-and-price

L'application de la méthode générale de branch-and-bound combinée à un processus de génération de colonnes pour résoudre un programme linéaire mixte est nommée *branch-and-price* dans la littérature spécialisée (voir notamment [41]). Cette application contient deux éléments principaux : les règles de branchement à appliquer pour séparer les ensembles de solutions réalisables et la détermination de bornes inférieures pour tous les problèmes à chaque nœud de l'arbre de branch-and-bound. De plus, il est souvent très intéressant de combiner le processus de branch-and-price avec des heuristiques d'arrondi qui permettent d'obtenir une solution réalisable plus rapidement autorisant ainsi à couper certaines branches de l'arbre de branch-and-bound. Ces trois éléments sont décrits ici.

### V.2.a) Règles de branchement

Les règles de branchement ont pour but de séparer l'espace des solutions réalisables de manière à exclure la solution fractionnaire courante, sans exclure la solution optimale entière et de manière à assurer la finitude de l'algorithme. Dans le cas du branch-and-price, il faut de plus vérifier la compatibilité du branchement avec le sous problème de tarification, c'est-à-dire que les colonnes exclues ne doivent pas être générées de nouveau. La structure du sous problème ne doit pas être trop modifiée de manière à permettre sa résolution efficace. C'est à cette aune que sont évaluées les différentes stratégies possibles de branchement qui sont décrites ici et pour lesquelles on détaille les conséquences sur le sous problème de tarification et notamment sur la structure des coûts réduits.

#### i) *Branchement sur les variables.*

- Règle de branchement correspondante

Une approche classique pour la résolution de programmes mixtes consiste à séparer le problème à partir de variables de décision fractionnaires dans la solution optimale non entière.

Dans le sous problème fils 1 on ajoutera la contrainte  $x_{f,r}=0$  et dans le sous problème 2 la contrainte  $x_{f,r}=1$ . Le choix de la variable concernée peut se faire selon plusieurs logiques : on peut prendre une variable dont la valeur dans la solution optimale du problème père est proche de 1 en espérant éliminer rapidement la branche de l'arbre correspondant à  $x_{f,r}=0$ . On peut au contraire choisir une variable très fractionnaire (dont la valeur est éloignée de 0 et de 1) afin de progresser plus vite vers une solution entière.

- Conséquences sur le sous problème de tarification

Si l'on impose  $x_{f,r}=1$ , on impose une trajectoire au vol : il faut donc soustraire de la capacité globale la capacité utilisée par ce vol empruntant la trajectoire et supprimer toutes les variables de décision correspondant au vol  $f$ . Moyennant ces adaptations, la structure du sous problème de tarification ne change pas et on peut donc appliquer le processus de génération de colonnes pour le sous problème correspondant à cet ajout de contrainte.

Si l'on impose  $x_{f,r}=0$ , on peut sortir la variable correspondante de l'ensemble des variables considérées en spécifiant au sous problème de tarification que cette variable ne doit pas être ajoutée. D'une manière plus mathématique, on peut d'ailleurs remarquer que si l'on impose une contrainte  $x_{f,r}=0$ , on ajoute également au problème dual une nouvelle variable qui jouera un rôle uniquement dans le calcul du coût réduit de la variable  $x_{f,r}$  empêchant celle-ci d'être négative. Si l'on garde exactement la même structure de coût réduit sans introduire la valeur duale correspondant à la nouvelle contrainte, cette variable risque d'être générée à nouveau : il faut alors chercher la deuxième variable avec le plus petit coût réduit. Ainsi au second niveau de l'arbre de branch-and-bound le sous problème de tarification va alors se transformer en recherche du 2<sup>ème</sup> plus court chemin puis 3<sup>ème</sup> et finalement  $k^{\text{ème}}$  etc. Cette approche n'a pas été retenue car le sous problème obtenu en interdisant une trajectoire n'est pas assez différent de son problème père : en effet de nombreuses variables (trajectoires) ont des caractéristiques proches (du fait de la grande liberté d'action, notamment sur la vitesse) et le fait d'interdire une trajectoire va simplement conduire au choix d'une trajectoire similaire sans changements majeurs pour le système.

- ii) *Branchement sur les caractéristiques des trajectoires*

- Règle de branchement correspondante

Dans la méthode de génération de colonnes, il est utile de considérer la structure du sous problème de tarification pour choisir les règles de branchement. Ici, les variables représentant des trajectoires, une grande famille de séparation possible consiste à imposer pour un seul vol donné  $f$  des contraintes sur la trajectoire empruntée : passage (ou pas de passage) par un point donné, par une trajectoire en 3 dimensions fixée ou par un secteur-période. D'une manière générale, cette séparation consiste à séparer les trajectoires possibles en deux sous ensembles :  $\mathcal{R}_1$  et  $\mathcal{R}_2$  de telle manière à ce que  $\mathcal{R}_f = \mathcal{R}_1 \cup \mathcal{R}_2$  et que  $\mathcal{R}'_f = \mathcal{R}'_1 \cup \mathcal{R}'_2$  si l'on considère la restriction de ces ensembles aux trajectoires retenues par le processus de génération de colonnes.  $\mathcal{R}_1$  peut par exemple être l'ensemble des trajectoires du vol  $f$  qui passent par la balise  $b$  et  $\mathcal{R}_2$  celles qui n'y passent pas.

- Expression mathématique des contraintes de séparation et conséquences sur le dual

Une telle séparation peut s'écrire formellement dans le programme linéaire en ajoutant la contrainte  $\sum_{r \in \mathcal{R}_1} x_{f,r} = 1$  pour obtenir le sous problème 1 et  $\sum_{r \in \mathcal{R}_2} x_{f,r} = 0$  pour obtenir le sous



problème 2. Dans le nouveau sous problème 1, la contrainte  $\sum_{r \in \mathcal{R}_f} x_{f,r} = 1$  est devenue superflue

et on peut la supprimer afin de faciliter l'écriture du problème dual et des coûts réduits. On note  $\widehat{\pi}'_f$  la variable duale associée à la nouvelle contrainte  $\sum_{r \in \mathcal{R}_f} x_{f,r} = 1$ . Dans le sous problème

1, le coût réduit d'une variable associée à une trajectoire  $r \in \mathcal{R}_1$  s'écrit alors à partir de la formule  $c'_{f,r} = c_{f,r} - \pi^t A_{f,r}$ <sup>8</sup>, soit

$$c'_{f,r} = c_{f,r} - \widehat{\pi}'_f - \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \widehat{\pi}_{s,p} - \sum_{g \in \mathcal{C}_f} a_{f,r} \widehat{\pi}_{f,g}$$

et le coût réduit d'une variable associée à une trajectoire  $r \in \mathcal{R}_2$  s'écrit simplement  $c'_{f,r} = c_{f,r} - \sum_{s \in \mathcal{S}} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \widehat{\pi}_{s,p} - \sum_{g \in \mathcal{C}_f} a_{f,r} \widehat{\pi}_{f,g}$ . Or les variables

duales  $\pi_{s,p}$  et  $\pi_{f,g}$  sont négatives ou nulles alors que  $c_{f,r}$  est positif. On en déduit que pour  $r \in \mathcal{R}_2$ , le coût réduit  $c'_{f,r}$  est strictement positif et qu'elles ne seront donc jamais ajoutées au programme maître restreint lors du processus de génération de colonnes. D'un point de vue pratique il n'y a pas besoin d'ajouter de nouvelles contraintes au programme linéaire : il suffit de supprimer du RMP toutes les trajectoires du vol  $f$  ne satisfaisant pas ces contraintes et on résout le programme linéaire obtenu.

- Conséquences sur le sous problème de tarification

Dans le sous problème de tarification, on ne considère ensuite que les trajectoires satisfaisant les contraintes définissant  $\mathcal{R}_1$  et  $\mathcal{R}_2$ . Le problème devient alors une recherche de chemin de coût réduit inférieur à une borne et satisfaisant une contrainte supplémentaire. Cette contrainte peut par exemple être de la forme suivante : passer par un point donné, éventuellement pendant une fenêtre de temps donnée. Quel que soit l'algorithme choisi pour résoudre le sous problème de tarification, cette contrainte supplémentaire sur les chemins nécessite des adaptations plus ou moins lourdes selon le type de séparation et la structure de l'algorithme. L'architecture de l'algorithme de résolution du sous problème de tarification développé au cours de ce travail se prête bien à l'ajout de contraintes supplémentaires. Ce point sera étudié dans le chapitre IV.

On peut enfin remarquer que la première approche de séparation est un cas particulier où l'on a  $\mathcal{R}_1 = \mathcal{R}_f$  et  $\mathcal{R}_2 = \mathcal{R}_f \setminus \{r\}$ .

- iii) *Branchement sur un secteur-période*

- Règle de branchement

Le branchement sur un secteur-période est un cas particulier du branchement sur les trajectoires. Etant donné que les contraintes de capacité sont définies par secteurs-périodes, une règle de branchement prometteuse consiste à diviser l'ensemble des trajectoires en fonction du passage ou non dans ce secteur-période : la séparation de  $\mathcal{R}'_f$  s'écrit alors :

$$\mathcal{R}'_1 = \bigcup_{r \text{ passe par } (s,p)} \{r\} \text{ et } \mathcal{R}'_2 = \bigcup_{r \text{ ne passe pas par } (s,p)} \{r\}.$$

Il reste alors à définir à partir de quel secteur-période la séparation va s'effectuer. Ryan et Foster ([44]) ont proposé une stratégie de branchement pour des problèmes de partitions d'ensembles, stratégie qui se révèle souvent efficace dans le cadre de la génération de colonnes. Appliquée à notre cas, elle consiste à choisir deux variables fractionnaires pour un vol. Du fait qu'il n'y a pas de variables dupliquées, il existe au moins un secteur-période qui est utilisé par la première trajectoire sans

<sup>8</sup> La formulation des coûts réduits est détaillée dans le chapitre IV paragraphe II.2.a) p.85

l'être par la deuxième : on choisit alors un de ces secteurs-périodes pour effectuer le branchement. On décrit ci-dessous le choix du vol sur lequel on va travailler ainsi que celui du secteur-période parmi ceux proposés la méthode de Ryan et Foster.

On propose la procédure suivante pour choisir ce secteur-période. On choisit comme première variable celle qui est la plus fractionnaire (la plus proche de 0.5). Elle correspond alors à un vol  $f$  et à une trajectoire pour ce vol. Parmi les variables concernant le même vol  $f$ , on considère ensuite la deuxième variable la plus fractionnaire. Enfin, parmi l'ensemble des secteurs-périodes utilisés par la trajectoire 1 (correspondant à la variable 1) mais pas par la trajectoire 2, on choisit celui ayant la valeur duale la plus basse (la plus haute en valeur absolue), synonyme de son importance dans le passage du flot de trafic. On dispose alors d'un vol et d'un secteur-période sur lesquels effectuer le branchement.

### V.2.b) Détermination de bornes inférieures au programme linéaire en cours

La deuxième base du processus de séparation et évaluation progressive réside dans l'évaluation du sous problème. Dans le cas de la programmation linéaire en nombres entiers elle est généralement faite par la résolution complète du programme linéaire relaxé. Lorsque l'on résout par la méthode de génération de colonnes, on s'approche très rapidement de la solution optimale mais la convergence finale peut être assez longue (tailing off effect). Pour économiser du temps de calcul, il est alors intéressant de disposer d'une borne inférieure pour le problème fils que l'on est en train de résoudre. Cette borne peut éventuellement permettre de stopper le processus de génération de colonnes avant son terme si la borne inférieure obtenue est supérieure à la meilleure solution réalisable entière actuelle.

Deux bornes ont alors été utilisées, les démonstrations sont présentées en Appendices, Annexes, II.1. La première borne a l'avantage de pouvoir être calculée facilement à partir des bornes inférieures sur les coûts réduits.

Si l'on note à une étape donnée de la génération de colonnes  $\bar{c}_f^* = \text{Min}(c'_{f,r} \forall r \in \mathcal{R}_f)$  le minimum pour chaque vol de la valeur des coûts réduits des trajectoires possibles du vol  $f$  et  $\bar{z}$  la valeur du critère à cette étape alors  $\bar{z} + \sum_{f \in \mathcal{F}} \bar{c}_f^*$  est une borne inférieure du problème maître.

Farley ([42]) propose une borne demandant un effort de calcul supplémentaire et qui une fois adaptée à notre problème peut s'écrire :

$K_1 \bar{z}$  est une borne inférieure du problème maître avec

$$K_1 = \text{Min}\left(\frac{c_{f,r}}{\lambda_{f,r}}; \forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f / \lambda_{f,r} > 0\right) \text{ et } \lambda_{f,r} = \pi^T A_{f,r} = \sum_{s,p} (\delta_{f,r,s,p} \pi_{s,p}) + \pi_f + \sum_{g \in \mathcal{C}_f} a_{f,r} \pi_{f,g}$$

$A_{f,r}$  est la colonne correspondant à la variable  $x_{f,r}$  dans la matrice des contraintes du problème maître restreint (habituellement notée  $A$ ).

Cette borne n'est toutefois pas utilisable en l'état car si la résolution du sous problème de tarification permet de calculer les  $\bar{c}_f^*$ , elle ne permet pas de calculer simplement  $K_1$ . On utilise donc une borne simplifiée mais plus accessible :

$K_2 \bar{z}$  est une borne inférieure du problème maître avec

$$K_2 = \text{Min}\left(\frac{c_f}{\pi_f + \sum_{g \in \mathcal{C}_f} a_f \pi_{f,g}}, f \in \mathcal{F} / \pi_f + \sum_{g \in \mathcal{C}_f} a_f \pi_{f,g} > 0\right)$$

et  $c_f = \text{Min}(c_{f,r}; \forall r \in \mathcal{R}_f)$ ,  $c_f$  est donc le minimum du coût réel des trajectoires pour le vol  $f$ .

L'efficacité de ces deux bornes est comparée dans le chapitre V concernant la mise en œuvre expérimentale des algorithmes.

### V.2.c) Heuristiques d'arrondi

#### i) *Rôle des heuristiques d'arrondi*

Le traitement de chaque branche de l'arbre de branch-and-bound conduit à une solution réalisable du problème maître restreint. Dans le cas où cette solution est fractionnaire, une heuristique d'arrondi va chercher à en tirer une solution réalisable entière. Elle peut ainsi permettre d'améliorer la meilleure solution entière actuellement connue et ainsi de couper éventuellement certaines branches de l'arbre de branch-and-bound. Les heuristiques d'arrondi ne sont cependant pas indispensables puisqu'en cas d'échec de l'heuristique, le processus de branch-and-bound conduit à une solution entière : les heuristiques permettent d'accélérer le processus.

#### ii) *Description d'une heuristique d'arrondi*

On propose ici une heuristique basée notamment sur la décomposition de la recherche d'une trajectoire en deux étapes : détermination de la trajectoire en trois dimensions puis attribution d'heures de passage aux entrées secteurs. Cette décomposition est d'ailleurs également utilisée pour la résolution du sous problème de tarification.

- Détermination d'une trajectoire de référence pour chaque vol

Soit  $f$  un vol dont les variables de décision associées ( $x_{f,r} \in \mathcal{R}'_f$ ) sont fractionnaires dans la solution optimale du problème relaxé : on repère les trajectoires dont les variables sont non nulles dans la solution optimale:  $\mathcal{R}''_f = \{x_{f,r} \in \mathcal{R}'_f / x_{f,r} > 0\}$ . On évalue chaque trajectoire spatiale (déterminée par la succession de secteurs traversés et que l'on note  $ts_{f,i}$ ) utilisée par ces trajectoires en additionnant la valeur de toutes les variables qui utilisent cette trajectoire :

$$\text{valeur (trajectoire spatiale } ts_{f,i}) = \sum_{\substack{x_{f,r} \in \mathcal{R}''_f \\ x_{f,r} \text{ parcourt } ts_{f,i}}} (x_{f,r})$$

Enfin, on choisit pour chaque vol la trajectoire spatiale qui maximise cette valeur. On ne considère maintenant plus que les trajectoires qui utilisent cette trajectoire spatiale. On définit alors une date d'entrée moyenne dans chaque secteur en faisant la moyenne pondérée par  $x_{f,r}$  des dates d'entrées dans le secteur de chaque trajectoire considérée.

- Allocation de retards supplémentaires pour respecter les contraintes de capacité

On a déterminé une trajectoire de référence pour chaque vol : il faut maintenant vérifier si les contraintes de capacité sont respectées. Si elles ne le sont pas, on utilise l'heuristique basée sur le principe du premier arrivé premier servi pour allouer à chaque vol des retards en l'air. Pour chaque secteur-période saturé, on établit une liste de vols traversant ce secteur-période, ordonnée par les dates d'entrée secteur fournies par les trajectoires de référence. On remplit alors le secteur-période jusqu'à saturation en suivant cette liste, puis on retarde les vols excédentaires. Si un vol est retardé plusieurs fois du fait de plusieurs secteurs-périodes, on lui attribue le retard maximum parmi les différents retards qui lui ont été alloués. Cette

heuristique similaire à l'allocation de créneaux par le système CASA de la CFMU a également été appliquée par Van den Akker et Nachtigall ([17]) pour la résolution du problème d'allocation de créneaux de décollage (sans considérer de routes ou de retards en l'air). Son principal inconvénient est de ne pas garantir l'obtention d'une solution entière réalisable : du fait des limitations de réserve de carburants l'attente en l'air est limitée, ce qui n'est pas le cas des retards au sol. On a cependant vu que les heuristiques d'arrondi sont des catalyseurs du processus de branch-and-bound mais qu'elles ne sont pas indispensables. Le point faible de l'heuristique présentée ici n'est donc pas rédhibitoire.

Le chapitre suivant est consacré à la résolution du sous problème de tarification ou oracle avec une modélisation réaliste du réseau aérien.



## chapitre IV Modélisation et résolution du sous problème de tarification

*Le sous problème de tarification qui apparaît dans le cadre du processus de résolution global est explicité dans ce chapitre : il s'agit d'un problème particulier de recherche de chemins avec contraintes dans un contexte dynamique parmi un ensemble continu de solutions. Une bibliographie spécialisée sur les problèmes de plus courts chemins présentant au moins un point commun avec ce problème est présentée et un algorithme de résolution est proposé. Celui-ci recherche dans un premier temps des trajectoires en 3 dimensions selon un mécanisme inspiré des algorithmes de marquage. Cette recherche s'appuie sur une modélisation originale de l'espace aérien en « feuilles d'espace » permettant de prendre en compte les changements de niveaux et les exigences particulières dues au contexte (routes aériennes, connaissance des dates de passage dans les différents secteurs...). Dans un second temps, un algorithme de programmation dynamique est appliqué pour trouver pour chaque trajectoire 3D les dates de passage minimisant le coût réduit. La résolution de l'équation de Hamilton-Jacobi-Bellman permettant de résoudre ce problème de manière efficace en prenant en compte la structure particulière de la fonction coût est décrite à la fin de ce chapitre (les démonstrations étant elles-mêmes en annexe).*

### I. Description détaillée du sous problème de tarification

#### I.1. Entrées/sorties du sous problème de tarification

L'objectif général du problème d'optimisation est d'obtenir des trajectoires opérationnelles et applicables à des vols réels. Les contraintes inhérentes à cet objectif (trajectoires choisies réalisables) ne sont pas intégrées dans le programme linéaire maître afin de pouvoir le résoudre plus rapidement. Par conséquent ces contraintes seront prises en compte dans le sous problème qui ne devra proposer que des trajectoires opérationnelles.

Nous allons décrire ici précisément les caractéristiques de ce sous problème destiné à chercher des trajectoires alternatives supplémentaires pour enrichir  $\mathcal{R}'_f$  de manière efficace. On cherche, d'un point de vue mathématique, des trajectoires qui ont un coût réduit négatif qui amélioreront ainsi la solution actuelle si elles sont retenues.

##### I.1.a) Entrées du sous problème

Les données fournies en entrée du problème sont :

- les valeurs duales associées à toutes les contraintes du problème maître restreint qui permettront de calculer les coûts réduits des trajectoires
- les informations fournies par le processus de branch-and-bound : secteurs-périodes obligatoires ou interdits
- les données aériennes sur le réseau (longueur, disponibilité...) et sur les avions (performances, coûts...)
- la position des avions et leur destination finale

### I.1.b) Contraintes à satisfaire

On souhaite obtenir des trajectoires en 4 dimensions satisfaisant les contraintes suivantes :

- leur coût réduit doit être négatif
- la consommation sur le trajet doit être inférieure à la réserve de carburant
- les trajectoires proposées doivent être réalisables par l'avion concerné dans des conditions normales d'utilisation
- les contraintes imposées par le processus de branch-and-bound doivent être respectées
- le nombre de changements de niveau doit être limité

A noter que pour obtenir une trajectoire de coût réduit négatif il est naturel de minimiser le coût réduit sans avoir cependant l'obligation de trouver la trajectoire de plus petit coût réduit. On a donc un problème de minimisation pour lequel on ne cherche pas forcément l'optimum.

### I.1.c) Sorties du sous problème de tarification

Le sous problème doit fournir un ensemble de trajectoires satisfaisant toutes les contraintes listées ci-dessus. Le nombre de trajectoires fournies résulte d'un compromis entre diminution du nombre d'itérations du processus de génération de colonnes et rapidité de la résolution du sous problème. Les valeurs utiles au calcul des bornes inférieures du programme linéaire (voir chapitre III paragraphe V.2.b) doivent être fournies comme sous produit de la résolution du sous problème de tarification. Enfin lorsque aucune trajectoire n'est trouvée le sous problème doit fournir un certificat d'absence de trajectoires de coût réduit négatif satisfaisant toutes les contraintes, et ce pour tous les vols.

## I.2. Lien avec les problèmes de plus courts chemins

### I.2.a) Similarités avec un problème de plus court chemin

Le problème de plus court chemin canonique qui consiste à trouver une séquence d'arcs sur un graphe permettant de joindre un sommet source (origine) à un sommet puits (destination) à coût minimum fait partie des problèmes d'optimisation les plus étudiés par la communauté scientifique, comme on peut le vérifier dans la bibliographie du paragraphe I.3. En conséquence, il existe un grand nombre d'algorithmes et de schémas de résolution efficaces pour résoudre ce type de problème. Le sous problème de tarification, qui cherche à trouver une trajectoire dans l'espace temps vérifiant un certain nombre de contraintes et survolant les routes aériennes, présente des caractéristiques qui peuvent le situer dans la famille des problèmes de plus courts chemins. Tout d'abord, le réseau de routes aériennes constitue une structure de graphe évidente avec les balises comme sommets et les routes aériennes reliant les balises comme arcs. De plus, on recherche à minimiser une quantité qui est, au moins en partie, décomposable dans l'espace et additive, c'est-à-dire par exemple que la consommation totale est la somme de la consommation sur chaque section de la trajectoire parcourue. Le caractère décomposable et additif des coûts sur les arcs est une caractéristique principale d'un problème de plus court chemin. Du fait de ces similitudes et afin de pouvoir profiter du corpus d'algorithmes existants, on cherche à se ramener à ce cadre bien connu de la théorie des graphes. Le caractère dynamique du réseau et les contraintes supplémentaires sur la trajectoire (consommation, nombre de changements de niveaux limité) font partie des

extensions naturelles du problème de plus court chemin et ne posent pas de difficultés particulières de modélisation, ce n'est pas le cas en revanche des autres spécifications de notre sous problème de tarification qui sortent du cadre général existant.

### I.2.b) Particularités non caractéristiques d'un problème de plus court chemin canonique

La principale limite à l'identification du sous problème de tarification à un problème de plus court chemin canonique réside dans l'impossibilité de définir explicitement un graphe composé de sommets et d'arcs bien identifiés. En effet, les changements de niveau ne se modélisent pas facilement sous forme de sommets et d'arcs : lors du changement de niveau, la trajectoire de l'avion peut entrer dans un secteur mais le lieu géographique et la date de l'intersection ne sont connues qu'en effectuant une modélisation de la montée de la trajectoire prenant en compte les caractéristiques de l'avion (performances, charges) et la définition du secteur pendant la période considérée. Une modélisation trop simplifiée du réseau aérien en 3 dimensions conduirait à des dates d'entrée secteurs peu précises et non compatibles avec le contexte de la régulation court terme. Par ailleurs, il est impossible de précalculer pour toutes les routes aériennes les trajectoires pour tous les avions dans toutes les caractéristiques de charge possibles afin de connaître leurs caractéristiques géométriques, les temps de parcours et les coûts associés à chaque portion de la trajectoire. Pour ces raisons on ne dispose pas d'un graphe complètement explicité *a priori* avec des sommets, des arcs, des coûts et des temps de parcours associés à ces arcs.

## I.3. Bibliographie et taxinomie des problèmes de plus courts chemins pertinents

### I.3.a) Introduction

On estime que plus de 2000 travaux scientifiques<sup>9</sup> concernant directement un problème de la famille des plus courts chemins ont déjà été publiés depuis la fin des années 50 et les applications concernant le champ des transports sont particulièrement nombreuses (voir le travail spécifiquement consacré aux problèmes de plus courts chemins dans les modèles de transport [52]). Il n'est pas question ici de faire une étude exhaustive des travaux sur ce sujet, même restreinte à un certain nombre de problèmes de plus court chemin spécifiques les plus proches de notre approche. On souhaite simplement être capable d'identifier les grandes familles d'extensions auxquelles notre problème se rapporte et les grandes familles d'algorithmes utilisables pour les résoudre et de donner pour chaque ensemble une porte d'entrée à la bibliographie spécialisée, ainsi qu'éventuellement les articles les plus marquants.

### I.3.b) Le problème de plus court chemin général

#### *i) Description*

Le problème d'arbre des plus courts chemins (Shortest Path Tree problem ou SPT) consiste à chercher un arbre couvrant à partir d'un sommet source sur un graphe orienté tel que le coût d'un chemin quelconque (au sens classique de la somme des coûts des arcs) entre la source et un sommet du graphe soit supérieur ou égal au poids du chemin (unique) entre la source et le même sommet en utilisant uniquement les arcs faisant partie de l'arbre. Ce problème consiste

---

<sup>9</sup> d'après Pallottino et al. ([52])



donc à trouver un plus court chemin entre le sommet source et chaque sommet du graphe. Dans le cas général, la complexité théorique d'un algorithme cherchant un plus court chemin entre une source et un puits ou entre une source et tous les autres sommets est la même, même si on peut dans le cas particulier source-puits utiliser des informations supplémentaires pour accélérer le processus. La plupart des algorithmes cherchent à déterminer le *potentiel* d'un sommet qui représente dans le cas général la valeur du plus court chemin entre la source et le sommet.

ii) *Les algorithmes de marquage*

La famille d'algorithme la plus répandue est celle des algorithmes de marquage (voir [52], [58] pour une étude plus approfondie de cette famille d'algorithmes). Cette méthode recherche un plus court chemin dans un graphe à partir d'une origine fixée. Pour chaque sommet  $s$  du graphe on garde en mémoire le potentiel, le prédécesseur et l'état (*non atteint*, *marqué*, *examiné*). On initialise le potentiel pour tous les sommets distincts de l'origine à l'infini et ils sont tous marqués comme *non atteints*. Quant au sommet origine, son potentiel est initialisé à 0 et son état à *marqué*. Tant qu'il y a des sommets *marqués*, l'algorithme en choisit un, noté  $e$ , et examine un à un tous ses successeurs, notés  $s$  : si le potentiel de  $e$  additionné à la distance entre  $e$  et  $s$  est inférieur au potentiel de  $s$ , le potentiel du sommet  $s$  est changé en potentiel de  $e$  plus distance entre  $e$  et  $s$ , son nouveau prédécesseur est  $e$  et son état devient *marqué*. Lorsque tous les successeurs de  $e$  ont été ainsi traités l'état du sommet  $e$  devient *examiné*. L'algorithme continue jusqu'à ce qu'il n'y ait plus de sommets *marqués* (voir Figure 16). La valeur du plus court chemin est alors donnée par le potentiel du sommet destination et le plus court chemin est reconstruit à partir des prédécesseurs.

**Initialisation** :  $E = \{\text{sommet correspondant à la source}\}$

**Déroulement** :

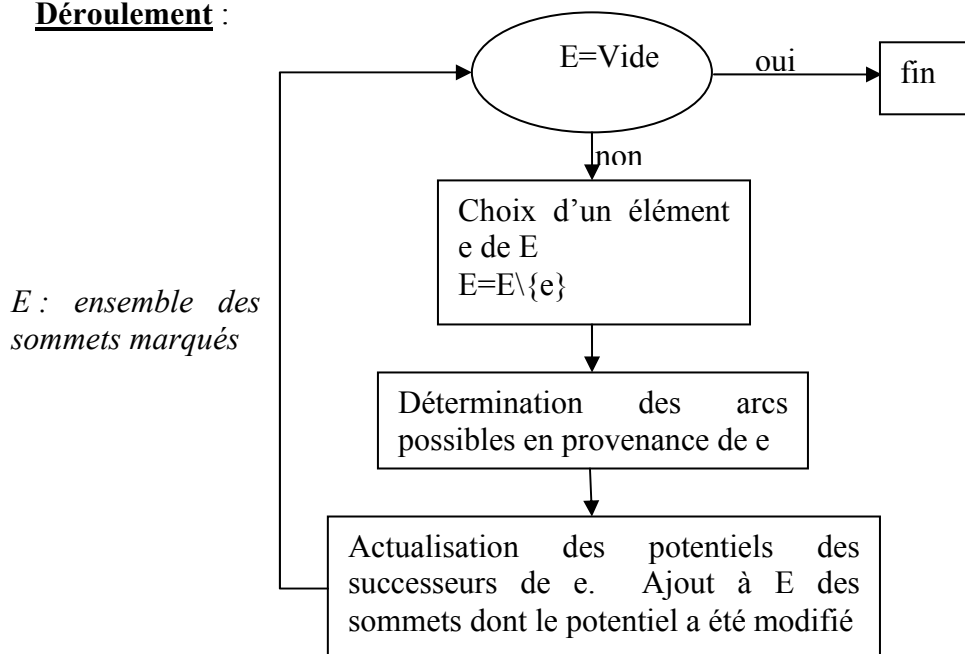


Figure 16) Principe général des algorithmes par marquage

Les deux grandes décisions qui vont caractériser alors un tel algorithme et déterminer son efficacité sont le choix du sommet à examiner et la structure de données pour stocker les éléments à examiner. L'algorithme de Dijkstra ([60]) choisit par exemple le sommet de plus petit potentiel, ce qui garantit de ne pas explorer deux fois un même sommet si les coûts sont positifs tandis que l'algorithme de Bellman et Ford ([61], [62]) choisit un sommet dont les potentiels de tous les prédécesseurs sont connus. Enfin, la structure de données choisie pour représenter la liste des sommets à explorer (piles First In First Out FIFO, arbres binaires, casiers...) est déterminante dans la complexité des algorithmes (voir [65], [56]).

### I.3.c) Les extensions du problème de plus court chemin

#### i) *Le plus court chemin avec contraintes*

A chaque arc du graphe est maintenant associé, en plus de son poids, un vecteur de consommation de ressources. La consommation totale d'une ressource par un chemin est la somme des consommations par arc emprunté. On cherche alors le plus court chemin qui vérifie aussi les contraintes de consommation de ressources maximales à ne pas dépasser (voir [51]). Ce problème appelé dans la littérature spécialisée CSP (Constrained Shortest Path problem) ou RCSP (Resource Constrained Shortest Path problem) est NP-complet même dans le cas d'une seule ressource additionnelle ([57]). Une version plus sophistiquée cherche à imposer que la consommation de chaque ressource à chaque sommet (et pas seulement à la destination) soit dans une fenêtre prédéfinie. Un cas particulier intéressant de ce problème est alors celui du plus court chemin avec fenêtre de temps qui impose que le passage aux sommets s'effectue durant une fenêtre de temps (voir notamment [54]).

Plusieurs stratégies de résolution sont proposées pour résoudre ce problème. On peut par exemple chercher les  $k$  plus courts chemins sans s'occuper des contraintes de ressources et chercher parmi ces chemins ceux qui vérifient les contraintes additionnelles, mais la détermination du nombre  $k$ , compromis entre la probabilité de trouver une solution réalisable et le temps de calcul, reste très délicate. Les outils issus de la programmation linéaire sont également utilisés pour résoudre ce problème qui s'écrit facilement comme un programme linéaire en nombres entiers. Pour résoudre ce problème, on peut alors par exemple relaxer les contraintes de ressource pour les intégrer dans la fonction objectif selon un processus nommé relaxation lagrangienne. Enfin, une approche couramment utilisée consiste à élargir la notion de potentiel afin de pouvoir utiliser les algorithmes de marquage : un sommet va alors pouvoir comporter plusieurs potentiels ou *labels* correspondant chacun à des chemins partiels jusqu'au sommet. La notion de *domination* d'un label sur un autre va alors permettre de ne pas considérer tous les labels possibles. En tant qu'algorithme de marquage cette famille est très dépendante de l'ordre dans lequel les sommets sont considérés et de la stratégie de comparaison des différents labels (ordre lexicographique, piles de priorités...).

#### ii) *Le plus court chemin dynamique*

Le facteur temps entre en considération dans les problèmes de plus court chemin dynamique. Le temps de parcours associé à chaque arc devient dépendant de la date d'arrivée à l'entrée de l'arc. On dit que la structure de temps vérifie la propriété FIFO (First In First Out) si quelles que soient deux dates  $t$  et  $t'$  telles que  $t < t'$  et quel que soit un arc  $ij$  alors

$$t + d_{ij}(t) \leq t' + d_{ij}(t')$$

Cela signifie qu'un voyageur n'a jamais intérêt à attendre à l'entrée d'un arc pour bénéficier d'un temps de trajet plus court : le temps perdu à attendre sera supérieur ou égal au

gain éventuel de temps de parcours. La recherche d'un plus court chemin sur un réseau dynamique vérifiant cette propriété est un problème pouvant être résolu par l'algorithme de Dijkstra dont la fonction d'actualisation des potentiels est légèrement modifiée : on prend comme temps de parcours d'un arc le temps nécessaire pour parcourir l'arc à la date du potentiel du sommet origine de l'arc. Lorsque la structure des temps de parcours n'est pas FIFO, le problème de plus court chemin est NP-difficile ([50]). Les problèmes dynamiques sont souvent résolus en étendant implicitement ou explicitement le réseau en un réseau espace-temps sur lequel on va appliquer des algorithmes classiques qui utiliseront le fait que le graphe espace-temps ne contient pas par construction de circuits.

*iii) Problème de plus court chemin avec origine-destination*

Lorsque l'on cherche le plus court chemin entre un sommet source et un sommet destination bien défini, on peut utiliser cette information supplémentaire pour accélérer la recherche. Le moyen le plus simple est de stopper les algorithmes classiques plus tôt lorsque le plus court chemin est trouvé pour notre sommet destination sans continuer jusqu'à la fin habituelle de l'algorithme. On peut également utiliser des estimations de distance entre les sommets du graphe et le sommet destination afin de guider la recherche, notamment dans le cas des algorithmes de marquage : c'est le principe de l'algorithme A\*. Des précalculs peuvent également être effectués pour accélérer toute requête ultérieure (voir [48]). Un autre principe de résolution consiste à effectuer un algorithme inversé en partant de la destination et en considérant les arcs dans le sens inverse et trouver la jonction entre l'algorithme normal et inversé.

I.3.d) Conclusion et choix d'un type d'algorithme de résolution

Le sous problème de tarification est donc un problème de plus court chemin origine-destination sur un réseau dynamique non FIFO (certains secteurs pouvant être fermés pendant une période et l'attente n'étant pas autorisée à tous les points du réseau aérien) avec des contraintes additionnelles (contraintes de consommation maximale et de limitation du nombre de changements de niveaux). Une particularité supplémentaire dans le cas de contraintes supplémentaires dues au processus de branch-and-bound est l'obligation de passer par un ou plusieurs secteurs-périodes prédéfinis. La famille d'algorithmes de marquage est la famille laissant la plus grande liberté pour inclure les particularités de modélisation spécifiques à notre problème. L'argument principal en faveur du choix d'un tel type d'algorithme est la possibilité de construire au fur et à mesure du déroulement de l'algorithme le graphe d'exploration sans avoir besoin de l'explicitement dès le début de l'algorithme comme c'est nécessaire dans les algorithmes basés sur la programmation linéaire. Ces raisons ont conduit au choix d'un algorithme de marquage dont les principales caractéristiques -choix du sommet, structures de données utilisées, mécanismes de domination et de comparaison des labels- sont exposées dans la suite de ce chapitre et évaluées dans le chapitre V consacré à la mise en œuvre opérationnelle. La connaissance de la destination est utilisée pour guider l'ordre d'examen des sommets et pour couper certaines branches de l'arbre d'exploration. La date d'arrivée étant inconnue, on n'a pas en revanche implémenté d'algorithme inversé partant de la destination.

## **II. Modélisation du sous problème de tarification**

### **II.1. Modélisation en un graphe**

#### **II.1.a) Les contraintes qui ont donné naissance à la modélisation**

Le système spatial sur lequel nous travaillons et dont la modélisation est maintenant nécessaire est le réseau de l'espace aérien utilisé par les avions durant leur vol. Plusieurs impératifs ont guidé la modélisation retenue : tout d'abord nous travaillons dans le contexte réel des routes aériennes (par opposition à l'hypothèse vol sans contraintes ou *free flight*) et dans ce contexte les avions suivent des routes aériennes préexistantes en respectant les niveaux de vol autorisés en fonction du segment parcouru. Sur un tel réseau, les changements de direction s'effectuent au dessus des balises qui marquent physiquement les routes aériennes. Du fait de sa nature de sommets (les balises) et d'arcs (les segments de route aérienne entre deux balises) la représentation par les graphes est la plus naturelle. Cependant les vols suivent une trajectoire en 3 dimensions et la modification du ou des niveaux de vol suivis par le vol fait partie des actions possibles envisagées dans le cadre de la régulation dynamique. Or, la représentation des changements de niveau de vol s'accorde mal avec une structure de graphe classique du fait qu'un changement de niveau de vol est continu et l'arrivée à une balise ne coïncide pas forcément avec l'arrivée à un niveau de vol (c'est-à-dire généralement les niveaux multiples de 1000 pieds).

Une solution consisterait à discrétiser la dimension verticale pour obtenir un graphe dont chaque sommet représenterait un point de l'espace en 3 dimensions. On se heurterait cependant, soit au problème de la modélisation réaliste du changement de niveau de vol si le pas de discrétisation retenu est trop grand, soit à la taille du graphe obtenu et à l'impossibilité de s'en servir de manière efficace si le pas de résolution est trop petit. Pour ces raisons, nous avons choisi de ne pas discrétiser la troisième dimension. De plus, la modélisation retenue doit permettre une modélisation assez fine des trajectoires pour avoir une bonne estimation de la capacité de ressource de contrôle aérien nécessitée par chaque trajectoire, c'est-à-dire notamment les dates d'entrée dans les secteurs. Enfin, la modélisation doit permettre de prendre en compte le caractère dynamique de l'espace aérien. Tout ceci nous a conduit à développer notre propre modélisation basée sur ce que nous appellerons les « feuilles d'espace ».

#### **II.1.b) Le graphe des feuilles d'espace**

Une feuille d'espace ou feuille est une portion d'espace bidimensionnelle entre deux balises consécutives sur une route aérienne. Ainsi sur la Figure 17, on a représenté la feuille correspondant à la portion de route aérienne entre les balises LISTO et HON, ainsi que les frontières des secteurs traversés par cette route aérienne : EGCC29G, EGDTS, EGCCSTF et EGTTCOW. Elle comprend de plus les données suivantes : les niveaux de vol autorisés sur cette portion de route (y compris les niveaux minimum et maximum), les frontières de secteurs, la disponibilité temporelle de la feuille et les feuilles suivantes pouvant être empruntées par un vol quittant la feuille courante. Chaque élément de cette description possède une validité temporelle. Ainsi, cette représentation permet de modéliser le caractère dynamique de la sectorisation ou des changements de niveaux de vol extrémaux, que ces changements soient planifiés ou non.

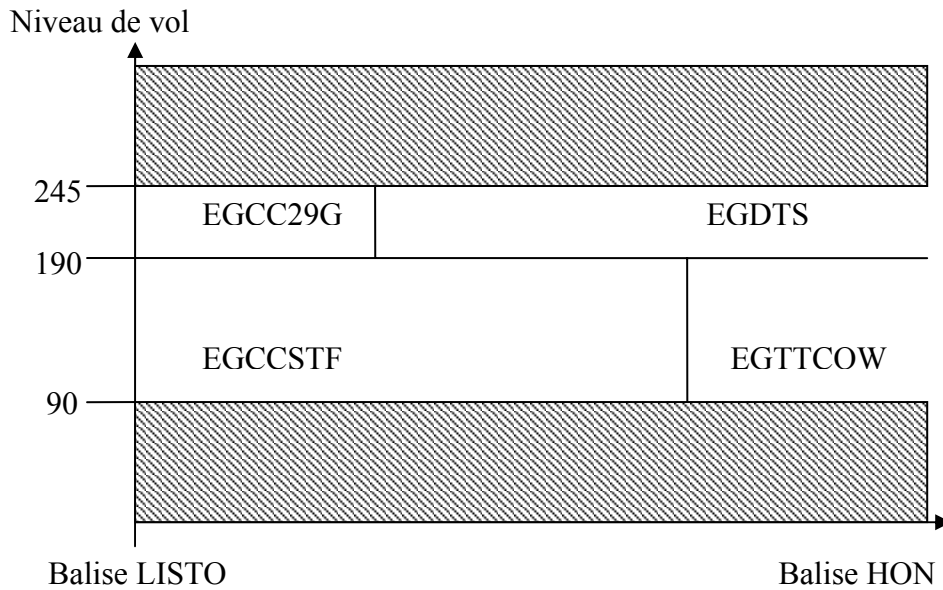


Figure 17 : feuille d'espace entre les balises LISTO et HON

L'ensemble des feuilles représente le réseau global des routes aériennes (voir Figure 18). Chaque feuille constitue le sommet d'un graphe global qui est utilisé lors du processus de résolution. La modélisation est donc basée sur la théorie des graphes, incontournable lorsque l'on recherche des plus courts chemins, avec cette différence que chaque sommet du graphe est cet objet complexe qu'est une feuille d'espace. On peut remarquer que cette modélisation de l'espace aérien en feuilles d'espace permet de prendre en compte directement les contraintes sur la trajectoire horizontale de l'avion (pas d'angle de changement de trajectoire supérieur à 90°) : en effet, lors de la définition des successeurs des feuilles d'espace il est très aisé de vérifier que cette contrainte est satisfaite et les trajectoires calculées à partir de ces feuilles la vérifieront également.

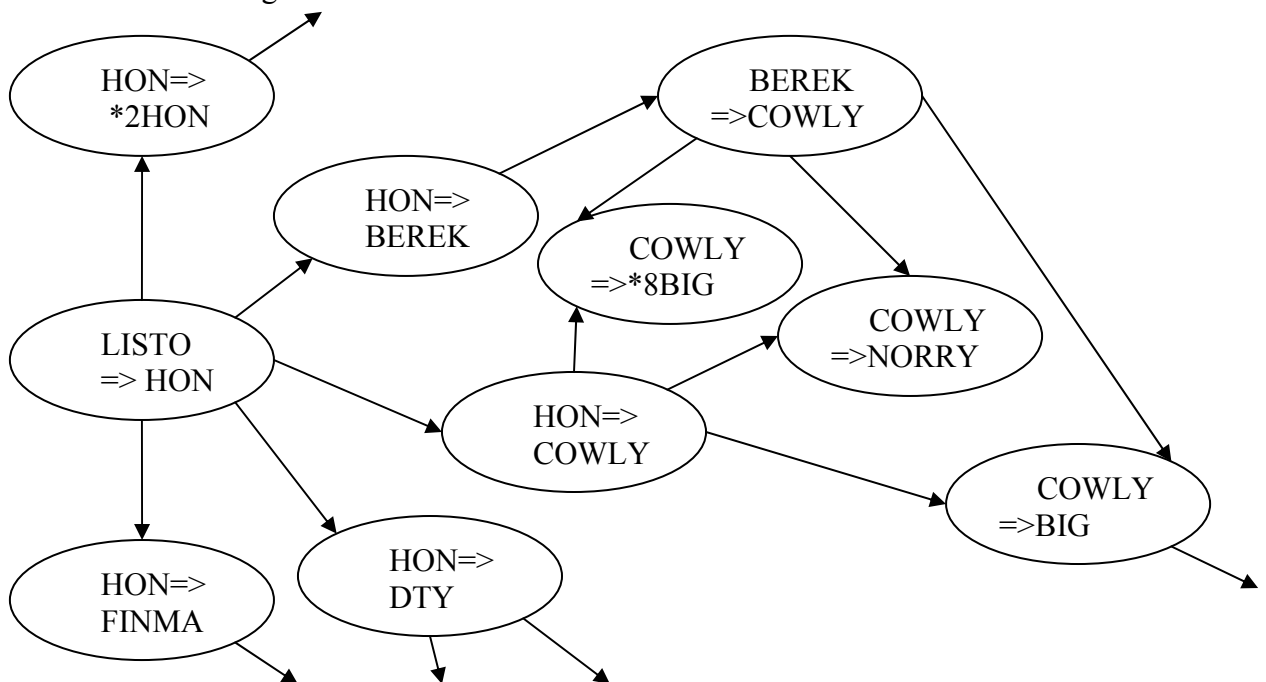


Figure 18) Portion du graphe des feuilles d'espace

## II.2. La fonction coût réduit

### II.2.a) Formulation du coût réduit

On étudie ici la structure du coût réduit d'une variable  $x_{f,r}$ . On note  $c_{f,r}$  le coût réel de la trajectoire  $r$  et  $c'_{f,r}$  son coût réduit dans la base optimale obtenue par résolution du problème maître restreint. Pour le vol  $f$  et la trajectoire  $r$  ce coût réduit est alors déduit de la formule donnant le coût réduit en fonction de la colonne de la matrice représentative du programme linéaire que l'on nomme généralement  $A$  :  $c'_{f,r} = c_{f,r} - \pi^t A_{f,r}$ .  $\pi^t$  représente la transposée du vecteur des variables duales. Dans la base optimale courante du problème maître restreint, cette formule nous permet d'écrire le coût réduit d'une variable :

$$c'_{f,r} = c_{f,r} - \widehat{\pi}_f - \sum_{s \in S} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \widehat{\pi}_{s,p} - \sum_{g \in C_f} a_{f,r} \widehat{\pi}_{f,g}$$

$$\text{avec } c_{f,r} = c^e_{f,r} + c^a_{f,r}$$

### II.2.b) Décomposition de la fonction coût réduit

On appelle une portion de trajectoire la restriction de la trajectoire à un lieu géographique donné : par exemple portion de trajectoire entre deux balises ou portion de trajectoire à l'intérieur d'un même secteur. Il faut alors distinguer parmi les composantes de la fonction coût réduit d'une trajectoire celles qui sont décomposables et additives, c'est-à-dire qui sont la somme des coûts sur les différentes portions de la trajectoire, de celles qui ne le sont pas.

#### i) Composantes décomposables de la fonction coût réduit

La composante d'exploitation  $c^e_{f,r}$  qui représente la consommation de carburant est décomposable quelle que soit la division spatiale des trajectoires et peut être actualisée au fur et à mesure que l'on construit les trajectoires dans un algorithme de construction de trajectoires portion par portion. De même, la composante  $\sum_{s \in S} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \widehat{\pi}_{s,p}$  est également décomposable : si la portion de trajectoire traverse un secteur  $s$  pendant la période  $p$ , on peut soustraire  $\widehat{\pi}_{s,p}$  au coût réduit de la trajectoire partielle.  $|\widehat{\pi}_{s,p}|$  représente le coût marginal d'utiliser le secteur  $s$  pendant la période  $p$  dans la situation décrite par la solution optimale courante du problème maître restreint.

#### ii) Composantes non décomposables de la fonction coût réduit

La composante  $\widehat{\pi}_f$  n'est pas décomposable mais elle est connue dès le début de la recherche de trajectoires,  $-\widehat{\pi}_f$  est la seule composante négative du coût réduit. Elle représente le coût marginal du vol  $f$  sur le système dans la dernière solution optimale : schématiquement elle indique combien il coûte au système d'avoir à allouer une trajectoire au vol  $f$ .

$c^a_{f,r}$  (coût du retard à l'arrivée pour le vol  $f$ ) et  $\sum_{g \in C_f} a_{f,r} \widehat{\pi}_{f,g}$  ne peuvent pas quant à elles être calculées avant la fin de la construction de la trajectoire, lorsque la date d'arrivée est connue.  $\widehat{\pi}_{f,g}$  représente le coût des conséquences de la date d'arrivée du vol  $f$  sur le vol en correspondance  $g$  dans la dernière solution retenue. Ces deux dernières composantes non décomposables ne peuvent pas être connues complètement lors de l'exploration du graphe. On

les estime au cours de la recherche de trajectoires en prenant en compte une date d'arrivée à l'aéroport au plus tôt qui permet de calculer une borne inférieure de ces deux composantes. Cela peut notamment influencer le choix des prochains sommets à explorer.

### **III. Résolution du sous problème de tarification**

#### III.1. Principe général de la résolution

##### III.1.a) Raisons de la dissociation du problème en deux phases

Les actions de régulation envisagées dans le cadre de la régulation court terme sont de deux natures différentes : de nature spatiale (changement de niveau et reroutement) et de nature temporelle (attente et modulation de vitesse). D'un point de vue de la résolution et de la recherche d'un ensemble de trajectoires pour chaque vol, cette distinction reste toujours pertinente même si elle n'est pas obligatoire (la dimension temporelle ne se distingue pas dans la théorie des graphes des autres dimensions). La principale difficulté repose dans le caractère continu de la modulation de vitesse ; une nouvelle fois, la simple discrétisation afin de pouvoir travailler sur un graphe classique n'est pas souhaitable pour éviter une explosion combinatoire. De plus, si l'échelon pertinent pour décider des changements de route est effectivement les balises, ce n'est plus le cas pour décider des changements de vitesse, l'échelon important étant l'arrivée à un nouveau secteur. Pour ces deux principales raisons la recherche de trajectoires s'effectue selon un processus à deux phases interdépendantes.

##### III.1.b) Description générale de la dissociation en deux phases

Dans le cas où cela n'est pas précisé, une trajectoire est définie en quatre dimensions. Dans la première phase on construit des trajectoires en 3 dimensions, en prenant déjà en compte la dimension temporelle pour calculer une borne inférieure de coût réduit pour toutes les trajectoires utilisant cette trajectoire 3D. Ces trajectoires 3D contiennent potentiellement des trajectoires associées réalisables. A chaque fois qu'une nouvelle trajectoire 3D est déterminée, on va effectuer une deuxième phase pour déterminer la politique optimale de vitesse, afin de trouver la trajectoire de plus petit coût réduit parmi celles suivant cette trajectoire 3D. Cette deuxième phase est basée sur un algorithme de programmation dynamique. Il y a alors trois possibilités d'arrêt.

- Toutes les trajectoires 3D potentiellement intéressantes ont été traitées
- On a déterminé un nombre de trajectoires supérieur à un nombre prédéterminé
- Le temps de calcul dépasse une borne prédéterminée

De cette manière on remplit les objectifs suivants. Tout d'abord, on peut certifier en ne fixant pas de limite de temps de calcul qu'il n'existe pas de trajectoires de coût réduit négatif lorsque l'algorithme ne trouve pas de solution. Ceci est à la base de la garantie d'optimalité du processus de génération de colonnes. Ensuite, on obtient le nombre souhaité de trajectoires, variées d'un point de vue temporel et spatial.

## III.2. Phase I : plus court chemin : recherche spatiale

### III.2.a) Introduction

L'algorithme élaboré pour la détermination des trajectoires spatiales de coût réduit négatif est développé à partir du principe général des algorithmes de marquage multi-labels. La particularité de cet algorithme est de ne construire que les portions du graphe nécessaires en s'appuyant sur le graphe des feuilles d'espace. La dimension temporelle est prise en compte par un label représentant un intervalle de temps possible. Le fonctionnement de l'algorithme est explicité dans les paragraphes suivants.

### III.2.b) Architecture de l'algorithme

#### i) Définition et composition d'une trajectoire partielle

On appelle *trajectoire partielle* d'un vol  $f$  une trajectoire entre la position courante de  $f$  et une entrée d'une feuille d'espace. Il y a équivalence entre une trajectoire partielle et une marque multi-labels puisqu'à chaque trajectoire partielle arrivant à une feuille d'espace, on peut associer un ensemble de labels. Chaque trajectoire partielle est associée à une et une seule feuille d'espace : celle au début de laquelle elle est arrivée. Les différents labels caractérisant une trajectoire partielle sont les suivants :

- Coût réduit partiel : somme des composantes décomposables et additives de la fonction coût réduit sur la portion de trajectoire parcourue, plus  $-\widehat{\pi}_f$  qui est ajoutée au départ
- Réserve de carburant actuelle
- Date d'arrivée  $t$  à la feuille courante en ayant suivi la vitesse nominale tout au long de la trajectoire partielle
- Date d'arrivée  $t_{\min}$  (resp.  $t_{\max}$ ) minimale (resp. maximale) d'arrivée à la feuille courante en ayant suivi la vitesse maximale (resp. minimale) tout au long de la trajectoire partielle
- Nombre de changements de niveau déjà effectués
- Etat actuel du vol : montée, palier ou descente
- Niveau de vol : parmi les séries de niveaux de vol autorisés si le vol est en palier ou altitude si le vol est en montée ou en descente
- Liste des frontières de secteurs déjà traversées et intervalles de temps associés

Parmi ces labels on peut distinguer ceux qui caractérisent l'exploration du graphe étendu (état du vol, intervalle de temps et niveau de vol) et ceux qui correspondent au potentiel élargi de l'algorithme de marquage : coût réduit partiel, consommation et nombre de changements de niveau effectués. Les dernières informations sont destinées à la deuxième phase de l'algorithme.

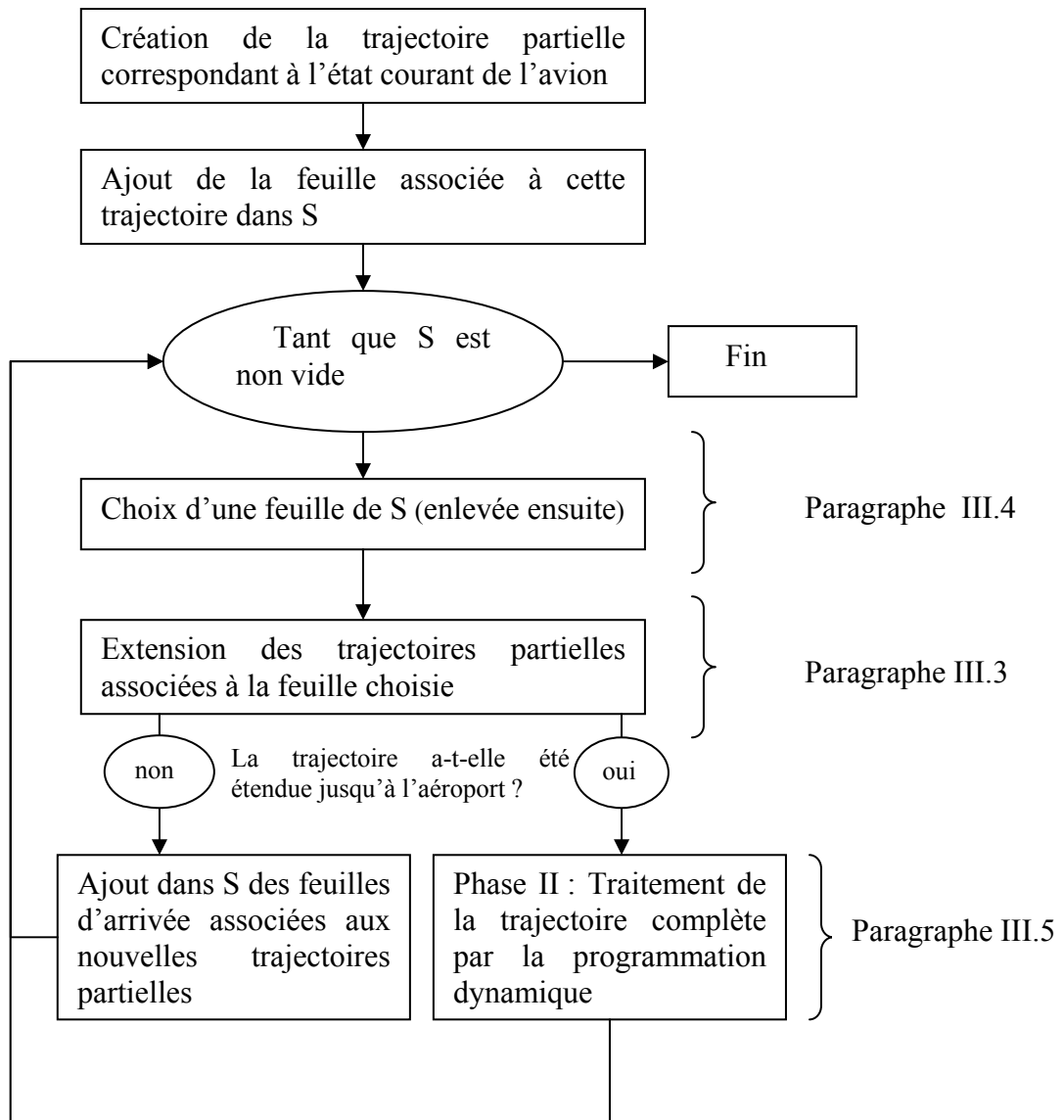
Les feuilles d'espace, qui représentent le réseau aérien, vont servir de support à l'algorithme. On regroupe les trajectoires partielles en fonction de la feuille à laquelle elles arrivent.



ii) *Principe général de l'algorithme*

Le principe de l'algorithme est alors le suivant (voir aussi la Figure 19) : création de la première trajectoire partielle correspondant à la position actuelle et initialisation des différents labels en fonction de l'historique de l'avion. La feuille à laquelle correspond cette première trajectoire partielle est alors déclarée *marquée* selon la terminologie des algorithmes de marquage et la phase d'initialisation de l'algorithme est terminée.

Lors d'une étape normale de l'algorithme, on choisit une feuille d'espace à laquelle sont associées des trajectoires partielles à étendre (c'est-à-dire une feuille marquée) et on étend toutes ces trajectoires partielles correspondantes en modélisant la traversée de la feuille d'espace choisie. Les trajectoires partielles contrevenant à l'une ou l'autre des exigences (consommation excessive, nombre de changements de niveau trop grand etc.) sont abandonnées. On crée ainsi un nouvel ensemble de trajectoires partielles associées aux feuilles successeurs de la feuille courante. La feuille courante devient maintenant *examinée*. Les trajectoires partielles atteignant l'aéroport de destination (trajectoires complètes) sont traitées par la phase II de l'algorithme afin de déterminer les temps d'entrée secteur optimaux. Si ce traitement permet de déterminer une trajectoire complète de coût réduit négatif, elle est alors ajoutée à la liste des nouvelles trajectoires calculées. L'algorithme continue tant qu'il reste des feuilles *marquées* ou tant qu'aucune condition d'arrêt n'est satisfaite. Cet algorithme fournit alors un ensemble de trajectoires réalisables de coût réduit négatif.



*S* : ensemble des feuilles contenant des trajectoires partielles à étendre

Figure 19) Architecture simplifiée de la résolution du sous problème de tarification

### III.2.c) Conditions d'arrêt supplémentaires

Si l'on continue le processus jusqu'à ce qu'il n'y ait plus de sommets à examiner, l'algorithme renvoie toutes les trajectoires satisfaisant les contraintes (exceptées celles qui sont dominées par une autre trajectoire, voir ci-après la notion de domination). Ceci n'est pas pertinent pour deux raisons principales : temps de calcul trop long des trajectoires et saturation du problème maître restreint, c'est-à-dire qu'on lui fournit trop de trajectoires alternatives et la résolution en est rendue difficile.

Pour ces raisons, on fixe un nombre de trajectoires par vol et lorsque ce nombre est atteint la recherche de trajectoires est interrompue. Avant de terminer la recherche, on termine

cependant le traitement de toutes les trajectoires partielles correspondant à la feuille courante, ce qui explique que l'algorithme peut rendre plus de trajectoires que le nombre fixé comme limite.

Pour les mêmes raisons, on fixe également un temps de calcul maximal pour la résolution d'un sous problème pour un vol donné : la recherche est stoppée lorsque ce temps de calcul est dépassé. Cette condition d'arrêt supplémentaire ne remet en cause l'optimalité globale de la résolution que dans le cas où aucune autre trajectoire n'a été trouvée pour aucun des autres vols commandables. L'arrêt pour cause de temps de calcul excédentaire est enregistré pour ne pas fausser les affirmations d'optimalité. La valeur de ces paramètres peut être ajustée en fonction du moment où le sous problème est appelé (c'est-à-dire en fonction du nombre de boucles de générations de colonnes déjà effectuées). Elle sera discutée dans la partie consacrée à la mise en œuvre expérimentale.

### III.2.d) Cas particulier des contraintes de branch-and-bound

#### *i) Conséquences des contraintes de branch-and-bound*

Du fait de la règle de branchement proposée (sur les secteurs-périodes), les contraintes de branch-and-bound se traduisent par la recherche de trajectoires passant par un ensemble de secteurs-périodes déterminés et évitant un autre ensemble de secteurs-périodes.

#### *ii) Prise en compte des secteurs-périodes interdits*

Pour les secteurs-périodes à éviter la prise en compte est immédiate : il suffit d'allouer provisoirement à ces secteurs-périodes un coût réduit infini ce qui a pour effet de disqualifier toutes les trajectoires utilisant ces secteurs-périodes.

#### *iii) Prise en compte des secteurs-périodes obligatoires*

Pour les secteurs-périodes obligatoires, la manière la plus simple de procéder consiste à ne rien modifier à la première phase de l'algorithme et à ne vérifier qu'à la fin de celle-ci si les contraintes sont respectées. Les trajectoires 3D ne passant pas par les secteurs requis sont alors éliminées directement et on peut continuer la marche normale de l'algorithme. Pour les trajectoires 3D passant par les bons secteurs, il est facile d'ajouter les contraintes de fenêtre de temps de passage dans le processus de programmation dynamique décrit dans la suite de ce chapitre. Les dates en dehors des périodes requises sont simplement interdites.

#### *iv) Accélération de l'algorithme*

La facilité d'intégration des contraintes de branch-and-bound dans l'algorithme de plus court chemin représente un avantage important des règles de branchement retenues. Quelques règles simples peuvent également être implémentées de manière à couper l'exploration des trajectoires plus tôt. Tout d'abord d'un point de vue temporel, dès que la date courante d'une trajectoire partielle dépasse la période d'un secteur-période obligatoire et que la trajectoire n'a pas traversé le secteur attendu, la trajectoire partielle en question peut être abandonnée.

Lors de l'extension des trajectoires, on calcule toujours la distance minimale jusqu'à l'aéroport afin d'avoir une estimation de la consommation minimale restante et de la date d'arrivée au plus tôt. Cette estimation de distance minimale peut être précisée en prenant en compte le fait qu'il faudra passer par le secteur donné : cette règle permet également de couper des trajectoires partielles plus tôt.

### III.3. Extension des trajectoires partielles

#### III.3.a) Principes de l'extension des trajectoires partielles

On souhaite étendre les trajectoires partielles en prenant en compte la traversée de la feuille et en minimisant les calculs. Or, pour un vol situé par exemple à l'entrée d'une feuille au niveau 130 et pouvant ensuite se rendre aux niveaux 150 ou 170, la phase de montée entre 130 et 150 est la même qu'on aille jusqu'à 150 ou que l'on monte jusqu'à 170. De plus, deux trajectoires partielles qui sont dans le même état au même niveau de vol vont avoir la même trajectoire géométrique au cours de la traversée de la feuille indépendamment de leur historique, d'où l'intérêt de travailler par feuilles. On a organisé le calcul des extensions de trajectoires de manière à ne pas faire dans la mesure du possible deux fois le même calcul. Le principe est le suivant : on dispose à l'entrée d'une feuille d'un ensemble de trajectoires partielles et on étend ces trajectoires en prenant en compte les différentes stratégies de changement de niveau de vol possibles (voir Figure 20). Ces différentes stratégies sont limitées, c'est-à-dire qu'on n'initie un changement de niveau qu'au début de la feuille à l'exception du début de la descente finale et du premier changement de niveau et qu'on privilégie les stratégies limitant le nombre de paliers intermédiaires (ces limitations sont expliquées dans le chapitre II paragraphe I.3.a) p.36).

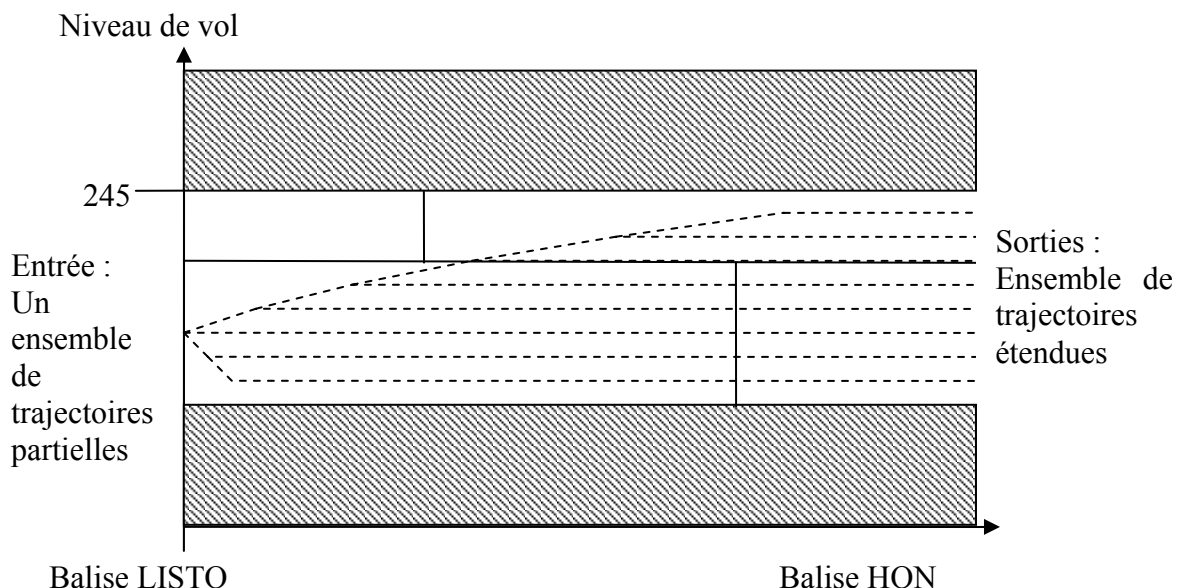


Figure 20 : Extension d'un ensemble de trajectoires partielles situées au même niveau par la traversée d'une feuille d'espace

#### III.3.b) Détails du mécanisme d'extension des trajectoires

##### i) Extension en palier

Quand une feuille est sélectionnée, on parcourt tous les niveaux autorisés et toutes les trajectoires situées au même niveau sont traitées en même temps. Pour une meilleure compréhension du mécanisme, examinons le cas d'un niveau quelconque, 200 par exemple. On calcule tout d'abord la trajectoire correspondant au vol en palier sur toute la feuille : on calcule la distance, le temps nécessaire et la consommation pour se rendre jusqu'à la prochaine frontière de secteur à l'intérieur de la feuille à la vitesse nominale ainsi que le

temps minimal et maximal et les consommations associées. Chaque trajectoire partielle arrivant au niveau 200 au début de la feuille est alors actualisée en fonction de ces données et de ses caractéristiques propres, on actualise notamment les intervalles de temps, la consommation et le coût réduit lié à l'entrée dans un secteur à une période donnée. On calcule également une borne inférieure à ce coût réduit en considérant toutes les autres périodes pendant lesquelles chaque trajectoire aurait pu croiser le secteur en changeant de politique de vitesse. On vérifie de plus la validité temporelle des frontières. Les trajectoires partielles actualisées sont éventuellement abandonnées si elles ne sont plus valides (consommation trop importante, borne inférieure de coût réduit positive par exemple). On continue ainsi jusqu'à la fin de la feuille où l'on crée de nouvelles trajectoires partielles associées aux feuilles successeurs.

*ii) Extension en montée ou descente*

Pour la montée (la démarche est ensuite la même pour la descente), on calcule déjà la portion entre 200 et 210 et l'on a ainsi de nouvelles trajectoires partielles temporaires ayant atteint le niveau 210 à partir du niveau 200 (on a pu croiser une frontière de secteur durant cette portion de montée auquel cas les trajectoires partielles ont été actualisées). Ces trajectoires partielles vont servir au calcul des trajectoires qui poursuivent en palier à partir du niveau 210 (si c'est autorisé) et de celles qui montent jusqu'au niveau 220, ces dernières servant elles mêmes de base aux trajectoires en palier au niveau 220 et celles qui continuent à monter et ainsi de suite. Arrivé à la fin de la feuille, on crée les nouvelles trajectoires partielles après une vérification de dominance sur les principaux labels (voir ci-dessous).

*iii) Remarques sur le mécanisme d'extension des trajectoires partielles*

On a ainsi une description des trajectoires modélisées assez finement (portion de droite entre chaque niveau de vol), ce qui permet une estimation précise des différentes dates d'entrée secteurs. Les frontières des feuilles sont stockées de manière à minimiser les calculs lorsque l'on cherche à savoir si une portion de droite coupe une frontière : frontières horizontales et verticales stockées séparément et classées par altitude (pour les horizontales) ou par abscisse (pour les verticales) croissantes. On peut noter que cette méthode pourrait prendre en compte une modélisation encore plus fine (portions de paraboles pour les trajectoires) avec des calculs d'intersection un peu plus compliqués mais un temps d'exécution quasiment identique. De même, des informations éventuelles sur la météo peuvent être facilement prises en compte à ce niveau là. Les trajectoires partielles arrivant en montée ou en descente au début de la feuille sont traitées selon le même principe mais séparément.

III.3.c) Détermination d'une borne inférieure au coût réduit et élimination des trajectoires partielles

*i) Utilité du calcul de la distance minimale pour rejoindre l'aéroport*

Plusieurs composantes du coût réduit d'une trajectoire n'étant pas décomposables et additives mais dépendantes de la date d'arrivée, il est utile de disposer d'une borne inférieure précise de la date d'arrivée, qui permet de calculer une borne inférieure au coût réduit et, le cas échéant, d'éliminer des trajectoires partielles (c'est-à-dire les abandonner sans continuer à les étendre). Pour cela, il faut estimer la distance minimale qu'il reste à parcourir pour l'avion

entre sa position actuelle et l'aéroport de destination. La distance minimale permet également de déterminer la consommation minimale pour rallier l'aéroport.

ii) *Précalcul de la distance minimale pour rejoindre l'aéroport*

La distance orthodromique<sup>10</sup> entre ces deux points pourrait être utilisée mais du fait que les avions suivent des routes aériennes, la différence entre la distance réelle à parcourir et la distance au plus court est significative. Pour gagner en précision, on a effectué un précalcul de distance minimale en suivant les routes aériennes entre toutes les balises intersections de routes aériennes et tous les aéroports significatifs. Pour cela, on a défini un graphe simple à partir du réseau aérien : les sommets correspondent aux balises et les arcs aux portions de routes aériennes les reliant. Les distances étant positives, on peut utiliser l'algorithme de Dijkstra pour trouver une borne inférieure à la distance entre les balises et les aéroports en suivant les routes aériennes. Ce précalcul est effectué hors ligne une seule fois pour toutes. Les disponibilités temporelles et les contraintes de changement d'angle ne sont pas prises en compte dans ce précalcul qui ne fournit qu'une borne inférieure à la distance entre la balise et les différents aéroports. On prend toutefois en compte le cap d'arrivée à la balise de départ.

iii) *Utilisations de la distance minimale précalculée*

A partir de cette distance minimale on peut alors estimer une date d'arrivée minimale et une consommation minimale pour rejoindre l'aéroport et en déduire une borne inférieure fidèle du coût réduit d'une trajectoire et de la consommation minimale. Cela permet d'éliminer le plus tôt possible des trajectoires partielles non valides. En donnant une indication sur le fait de savoir si l'on se rapproche ou si l'on s'éloigne de l'aéroport de destination, le précalcul des distances minimales est également utilisé lors du choix de la prochaine feuille à examiner

III.3.d) *Domination des trajectoires*

i) *Cas des trajectoires partielles*

Avant de créer les nouvelles trajectoires partielles, celles-ci sont comparées avec les trajectoires partielles existantes au même niveau. On ne compare que les trajectoires ayant le même parcours en terme de secteurs-périodes. On estime ensuite les trajectoires en fonction de deux paramètres : le nombre de changements de niveau et la consommation de carburant. Une trajectoire est dite dominée si elle est moins bonne qu'une autre pour les deux paramètres, auquel cas elle est abandonnée.

ii) *Cas des trajectoires complètes*

Les trajectoires complètes 4D (jusqu'à l'aéroport d'arrivée) sont également comparées avec toutes les trajectoires existantes : celles que l'on vient de calculer et celles déjà présentes dans le problème maître restreint. On ne compare de la même manière que des trajectoires ayant le même parcours de secteurs-périodes. Les caractéristiques qui permettent alors de comparer deux trajectoires sont la date d'arrivée et la consommation (toutes les trajectoires complètes ont un nombre de changement de niveau admissible). Les trajectoires déjà présentes peuvent être modifiées si l'algorithme a trouvé une amélioration. On utilise la même notion de domination pour garder ou éliminer les trajectoires complètes : une trajectoire est dominée si elle est moins bonne pour toutes les caractéristiques considérées.

---

<sup>10</sup> La distance orthodromique est la plus courte distance entre deux points à la surface d'une sphère

### III.4. Règles de sélection des feuilles à examiner

#### III.4.a) Introduction

Le choix de la feuille à examiner parmi les feuilles marquées peut s'effectuer selon plusieurs modalités décrites ci-dessous. La comparaison expérimentale entre ces différentes modalités sera également étudiée dans le chapitre V. Il est à noter que le choix de la feuille à explorer est facilement paramétrable et peut changer selon le moment où l'algorithme est appelé lors du processus de global de résolution. Ainsi, lors des premières boucles, l'objectif est de trouver rapidement de bonnes trajectoires alternatives pour avoir une bonne estimation des valeurs duales. Lors des boucles suivantes, l'objectif est de prouver qu'il n'existe plus de solution ou d'en trouver de plus atypiques.

#### III.4.b) Exploration en profondeur contre exploration en largeur

Il existe deux grandes stratégies d'exploration d'un graphe de possibilités. L'exploration en largeur étudie toutes possibilités à un niveau avant d'étudier les possibilités au niveau suivant. L'exploration en profondeur choisit au contraire une possibilité à chaque niveau et continue ainsi jusqu'à arriver à la fin de l'arbre des possibilités. Dans notre cas, une exploration en largeur consiste à explorer toutes les feuilles autour de la position d'origine et à ne s'éloigner que concentriquement de celle-ci. Une exploration en profondeur consiste au contraire à se rapprocher plus directement de l'aéroport de destination. Ces deux stratégies d'explorations ont des avantages et des inconvénients distincts.

##### *i) Avantages et inconvénients d'une exploration en largeur*

On peut remarquer que les calculs à effectuer pour étendre deux trajectoires partielles arrivant à la même position au début d'une feuille sont les mêmes, indépendamment du parcours précédent. Si les deux trajectoires sont déjà présentes au moment où l'on sélectionne la feuille concernée, ces calculs ne seront effectués qu'une fois. Par contre, si l'on sélectionne la feuille dès qu'on a associé la première trajectoire partielle alors que la deuxième n'est pas encore connue, le calcul du parcours de la feuille sera effectué deux fois. Une exploration en largeur vise donc à limiter ces cas de figures. Par ailleurs, en étudiant toutes les possibilités à un niveau avant de passer au niveau supérieur, l'exploration en largeur augmente la diversité des solutions trouvées.

L'inconvénient principal de cette stratégie est qu'elle est plus lente : elle nécessite plus de temps de calcul pour trouver un nombre fixe de trajectoires. De plus, dans le cas où l'on ne connaît pas encore beaucoup de trajectoires, l'arbre des possibilités à explorer est trop grand et l'algorithme va avoir besoin de trop de temps ou de place mémoire.

##### *ii) Avantages et inconvénients d'une exploration en profondeur*

Une exploration en profondeur va permettre de calculer très rapidement un ensemble de trajectoires complètes et directes. Par contre, l'éventail des solutions trouvées sera beaucoup plus restreint puisqu'elles ne différeront que par les dernières décisions. De plus, quand une exploration systématique est nécessaire (pour prouver qu'il n'y a plus de solution notamment), l'exploration en profondeur perd son avantage de rapidité puisque de nombreux calculs vont devoir être effectués plusieurs fois.

### III.4.c) Définition du potentiel d'une feuille

Il faut commencer par associer à chaque feuille un potentiel qui permet de comparer deux feuilles entre elles. Ce potentiel est tiré des différentes trajectoires partielles associées à la feuille. Plusieurs formes de potentiel possibles sont décrites ci-dessous.

#### *i) Réserve de carburant maximale parmi les trajectoires associées*

Le potentiel de chaque feuille est le maximum des réserves de carburant des trajectoires associées restant à étendre. Cette stratégie vise clairement à effectuer une exploration en largeur puisqu'on s'éloignera de la position d'origine de manière concentrique. Elle paraît a priori adaptée à une recherche exhaustive des trajectoires lorsqu'un certain nombre d'itérations de génération de colonnes ont déjà été effectuées.

#### *ii) Maximum de la réserve de carburant moins la consommation minimale*

Le potentiel de chaque feuille est le maximum des réserves de carburant des trajectoires associées restant à étendre moins la consommation minimale pour rejoindre l'aéroport de destination. Cette stratégie inspirée de l'algorithme A\* vise tout comme la précédente à limiter le nombre de calculs redondants tout en profitant des informations supplémentaires sur la suite de l'exploration en « visant » l'aéroport de destination.

#### *iii) Distance minimale pour rejoindre l'aéroport de destination*

Le potentiel de chaque feuille est la distance minimale pour rejoindre l'aéroport de destination. C'est évidemment une stratégie d'exploration en profondeur, indépendante des trajectoires partielles associées à la feuille. Cette stratégie cherche à trouver rapidement des trajectoires complètes : adaptée a priori aux premières itérations de génération de colonnes.

#### *iv) Coût réduit minimal des trajectoires partielles non explorées présentes dans la feuille*

Le potentiel de chaque feuille est le minimum des coûts réduits partiels des trajectoires associées restant à étendre. Cette stratégie basée sur l'algorithme de Dijkstra ne garantit pas toutefois que chaque feuille ne sera explorée qu'une seule fois comme dans l'algorithme de Dijkstra. En effet, comme on étend toutes les trajectoires d'une feuille en même temps, toutes les trajectoires partielles ne sont pas considérées dans l'ordre croissant du coût réduit. Elle est similaire à la première stratégie et paraît conseillée après un certain nombre de boucles de génération de colonnes.

#### *v) Estimation minimale des coûts réduits des trajectoires partielles présentes*

Le potentiel de chaque feuille est le minimum de l'estimation de coût réduit de chaque trajectoire associée restant à étendre. Cette estimation du coût réduit est faite à partir de la portion de trajectoire déjà déterminée (coût réduit partiel) et à partir des distances minimales précalculées entre les feuilles et l'aéroport de destination pour la portion restante de la trajectoire. Ce choix inspiré de l'algorithme A\* est similaire à la stratégie ii) tout en prenant mieux en compte les coûts réduits



vi) *Date d'arrivée minimale à la feuille parmi les trajectoires présentes*

Le potentiel de chaque feuille est le minimum des dates d'arrivée à l'aéroport (à la vitesse nominale) des trajectoires associées restant à étendre. Cette stratégie, parfois rencontrée sous le nom de *chrono-SPT* ([52]), est conseillée dans le cas des problèmes de plus court chemins dynamiques et vise également à minimiser le nombre de sélections d'un même sommet.

III.4.d) Structures de données pour déterminer la séquence des sommets dans l'algorithme

Une fois qu'on a déterminé le potentiel d'une feuille qui permet de les comparer deux à deux, il faut choisir la structure de données utilisées pour stocker les feuilles marquées en attente d'être examinées. L'objectif général est de minimiser pour chaque feuille le nombre de passage dans la liste des feuilles à examiner. Pour cela deux grandes stratégies sont possibles. On peut choisir systématiquement la feuille de plus petit potentiel et utiliser alors des structures de données telles que les piles de priorité pour accéder rapidement à l'élément de plus petit potentiel. On peut également suivre des règles de sélection plus empiriques basées notamment sur des piles auxquelles on ajoute les feuilles soit en tête soit en queue. Les deux stratégies sont décrites ci-dessous.

i) *Les piles de priorités*

Le potentiel d'une feuille est généralement défini à partir de caractéristiques des trajectoires partielles associées à la feuille. Ces caractéristiques des trajectoires ont la particularité d'avoir une évolution monotone (soit toujours en augmentation, soit toujours en diminution) lors de l'extension des trajectoires au travers d'une feuille. Ainsi, la réserve de carburant diminue alors que l'estimation de coût réduit ou le coût réduit partiel augmentent. Si l'on souhaite explorer le graphe des possibilités en largeur, il est alors naturel de choisir les feuilles dans l'ordre dicté par leur potentiel (le plus petit en premier si le potentiel augmente lors de l'extension des trajectoires partielles ou le plus grand s'il diminue). Si l'on stocke linéairement les feuilles dans leur ordre d'arrivée, il ne faut qu'une opération pour ajouter une feuille à la liste, par contre il faut  $n$  opérations pour trouver la feuille de plus petit potentiel où  $n$  est le nombre de feuilles déjà dans la liste. Les structures de données appelées *piles de priorité* permettent de trouver ce minimum en un temps proportionnel au logarithme de  $n$ . Les piles de priorité utilisées lors de ce travail sont décrites en détail dans les annexes (p.162 Annexes II.1.d).

ii) *Les règles d'Esopo-Pape et variantes à deux piles (algorithmes à seuil)*

L'algorithme dit d'Esopo-Pape ([59]) maintient deux piles : un sommet (pour nous une feuille) qui arrive pour la première fois dans les sommets marqués est stocké à la fin de la pile  $Q_2$  tandis que s'il revient, il est stocké à la fin de la pile  $Q_1$ . On considère alors les sommets de  $Q_1$  en premier puis, lorsque celle-ci est vide, ceux de  $Q_2$ . Cette méthode donne généralement de très bons résultats. Une méthode plus sophistiquée appelée méthode de seuil (*threshold algorithm* [55]) gère également deux piles : un sommet est ajouté à la fin de la pile 1 si son potentiel ne dépasse pas un certain seuil ou à la fin de la pile 2 dans le cas contraire. Lorsque la pile 1 est vide, les sommets de la pile 2 qui ne dépassent pas le seuil actuel sont alors replacés dans la pile 1 et le seuil est réajusté. Cet algorithme souffre de la difficulté de définir la valeur du seuil et n'a pas été implémenté pour cette raison. Bertsekas ([53]) propose une variante de ces algorithmes : avant d'ajouter un sommet à une pile, on compare son label avec celui du sommet situé en tête de la pile : s'il est meilleur on le met à la tête de la pile, sinon à

la fin de la pile. Les conséquences du choix de l'une ou l'autre de ses structures de données sur les performances de l'algorithme de résolution du plus court chemin sont décrites dans le chapitre V.

On décrit maintenant la deuxième phase de l'algorithme pendant laquelle on calcule pour chaque trajectoire 3D retenue la politique de vitesse associée optimale.

### III.5. Phase II : Programmation dynamique : recherche temporelle

#### III.5.a) Notations

Les notations spécifiques à la résolution par la programmation dynamique sont présentées dans le Tableau 3.

NOTATIONS	Symbole
<b>Programmation Dynamique</b>	
Le sommet du graphe état-temps correspondant au passage à la date $t$ à l'entrée du secteur $s$	$t_s$
Potentiel d'un sommet	$V(t_s)$
Coût de passer du secteur $s-1$ au secteur $s$ en une durée de $d$ secondes	$c_s^e(d)$
Durée maximale pour passer du secteur $s-1$ au secteur $s$ , les secteurs étant numérotés dans l'ordre de passage de la trajectoire étudiée	$d_{s \max}$
Durée minimale pour passer du secteur $s-1$ au secteur $s$	$d_{s \min}$
Durée pour passer du secteur $s-1$ au secteur $s$ à la vitesse nominale	$d_{s \text{ nom}}$
Aéroport de destination	$S_d$
Pente du segment croissant de la fonction de consommation (voir Figure 23)	$f_1$
Valeur absolue de la pente du segment décroissant de la fonction de consommation (voir Figure 23)	$f_2$

*Tableau 3) Notations relatives au processus de programmation dynamique*

#### III.5.b) Données calculées lors de la première phase

Lors de la première phase de recherche de trajectoires 3D, la dimension temporelle est déjà prise en compte puisqu'à chaque trajectoire partielle on associe une date de passage de référence au début de chaque feuille calculée pour la vitesse nominale de l'avion. On calcule également un intervalle de dates de passage possibles dont la borne inférieure est calculée à partir de la vitesse maximale de l'avion et la borne supérieure à partir de la vitesse minimale. Lorsqu'une trajectoire 3D est étendue jusqu'à l'aéroport de destination, on dispose donc d'une enveloppe temporelle possible en fonction de la politique de vitesse suivie. Pour gérer la dimension temporelle, le point important n'est plus le survol d'une balise mais l'entrée dans un nouveau secteur car les capacités sont définies secteurs par secteurs. Ainsi, les intervalles de temps de passage possibles que nous venons de décrire sont calculés non seulement pour les survols des balises mais aussi pour les différentes entrées secteurs. Entre deux entrées secteur on connaît donc la durée minimale de parcours, la durée « normale » et la durée maximale ainsi que la consommation de carburant associée à chaque durée.

### III.5.c) Objectif de la deuxième phase et rappel de la structure de coût réduit

#### i) *Objectif de la deuxième phase de l'algorithme*

L'objectif de la deuxième phase est de déterminer pour chaque trajectoire 3D les dates de passage optimales à chaque entrée secteur de manière à obtenir la trajectoire 4D de coût réduit minimal correspondant à cette trajectoire 3D. On rappelle que l'obtention de la politique de vitesse à choisir pour obtenir les dates de passage voulues est un problème à part entière évoqué dans le rapport du LICIT sur la régulation court terme ([3]). On a alors un problème qui consiste à déterminer des temps de passage à un certain nombre de points tout en minimisant une fonction coût réduit dépendant de ces temps de passage. Cette structure de problème est particulièrement adaptée à une modélisation par un programme dynamique à états continus et étapes discrètes que nous allons présenter ci-dessous après avoir explicité la structure de coût réduit d'une trajectoire.

#### ii) *Structure de coût réduit d'une trajectoire*

Le coût réduit d'une trajectoire dépend de trois facteurs : la consommation de carburant, la date d'arrivée et l'utilisation des ressources de capacité. Le minimum de consommation est atteint lorsque le vol utilise la vitesse nominale, par définition de cette dernière. Les coûts liés à la consommation sont convexes et les coûts liés à la date d'arrivée sont linéairement croissants en fonction de la date d'arrivée. En revanche la seule information disponible a priori sur les coûts liés à l'utilisation de la ressource de capacité est qu'ils ont une structure en escalier dont les intervalles de temps sont les intervalles retenus lors de la définition de la capacité (voir le paragraphe Capacité des secteurs et lissage de la charge p.31). Cette structure en escalier est directement liée à la fonction de charge retenue, pour une fonction de charge prenant en compte par exemple le nombre de vols dans un secteur cela ne serait plus vrai. On trouvera une illustration graphique de la structure de coût réduit ci-dessous dans le paragraphe III.5.e) de ce chapitre.

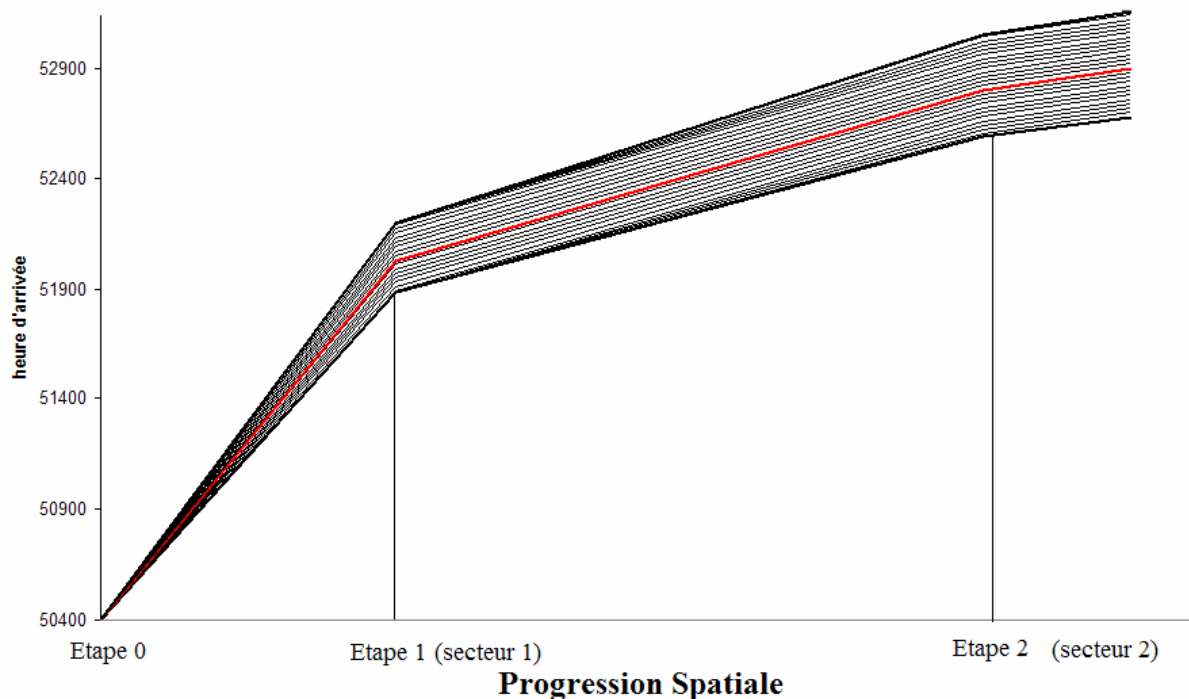
### III.5.d) Présentation de l'algorithme de Bellman

La programmation dynamique a été introduite par Bellman à la fin des années 50 ([68]) et a pour objectif premier la résolution de problèmes de contrôle. Les problèmes de programmation dynamique sont caractérisés par plusieurs éléments. Tout d'abord, le problème peut être décomposé en plusieurs *étapes* (on trouve également *temps* ou *périodes*), au début de chaque étape une *décision* est requise et le système est dans un *état* donné. La *décision* prise et l'*état* du système à l'*étape*  $n$  vont conditionner l'*état* du système à l'*étape*  $n+1$ . Le coût du passage d'un état donné à un autre état à la période suivante est donné par une *fonction de transition*. La programmation dynamique repose sur la propriété suivante que le système doit vérifier : à un état donné et à une étape donnée, la décision optimale pour toutes les étapes suivantes ne dépend ni des décisions ni des états précédents.

#### i) *Modélisation*

Dans notre cas les *étapes* sont les entrées dans un nouveau secteur, l'*état* est la date d'arrivée à ce secteur et la *décision* est le temps passé pour rejoindre le secteur suivant. On cherche alors à trouver la séquence optimale de temps passés entre deux secteurs de manière à minimiser le coût de la trajectoire. Ce problème posé sous cette forme peut être résolu par un algorithme classique de Bellman pour la programmation dynamique. La Figure 21 présente à

titre d'exemple l'enveloppe des solutions temporelles parmi lesquelles on va chercher celle de coût minimum.



**Figure 21) Enveloppe des solutions temporelles possibles pour une trajectoire spatiale donnée (remarque : l'échelle des abscisses n'est pas continue)**

ii) *Algorithme de Bellman dans le cas état discret*

L'algorithme de Bellman simple utilisé pour résoudre des problèmes de plus court chemin dans un graphe sans circuits peut servir à résoudre un problème de programmation dynamique. En effet, résoudre un tel problème consiste à trouver un plus court chemin sur le graphe états-étapes généralement appelé états-périodes ou états-temps (voir Figure 22). Les sommets de ce graphe correspondent aux couples (état ; étape) possibles. Deux sommets (état  $i$  ; étape  $j$ ) et (état  $k$  ; étape  $j+1$ ) sont reliés par un arc si et seulement si la transition est possible. Le poids de cet arc est alors donné par la fonction de transition. Le principe de l'algorithme est alors le suivant : on définit le potentiel  $V$  d'un sommet comme la valeur du plus court chemin entre l'état d'origine à l'étape 0 et le sommet. Le potentiel d'un sommet est le minimum sur l'ensemble des sommets prédécesseurs de la somme du potentiel du sommet prédécesseur et du coût de la transition du sommet prédécesseur au sommet étudié. Cette équation dite de Hamilton-Jacobi-Bellman est à la base de l'algorithme de Bellman. Si on note  $V(X)$  le potentiel d'un sommet  $X$  et  $C(X \rightarrow Y)$  le coût de la transition du sommet  $X$  au sommet  $Y$  elle s'écrit :

$$V(Y) = \text{Min} \{ V(X) + C(X \rightarrow Y), X \text{ prédécesseur de } Y \}$$

L'algorithme de Bellman va alors procéder période par période en profitant du fait que le graphe états-étapes est sans circuits et que tous les prédécesseurs d'un sommet sont compris dans la période précédente. On détermine le potentiel de tous les sommets à une période donnée, puis grâce à la formule de Hamilton-Jacobi-Bellman on peut calculer les potentiels des sommets situés à la période suivante. On continue jusqu'à ce que tous les potentiels soient

connus : on peut alors remonter le graphe de proche en proche pour trouver la solution optimale.

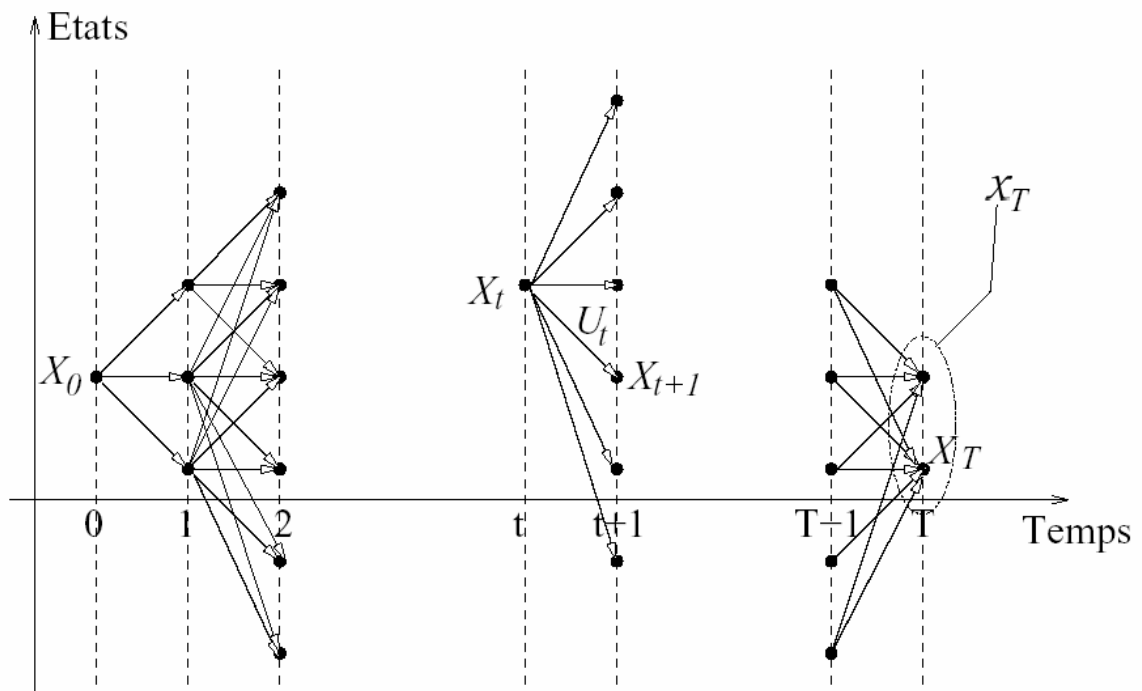


Figure 22) Graphe Etats-Etapes (ou Etats- Périodes ou Etat-Temps) caractéristique d'un problème de programmation dynamique ([63])

### iii) Algorithme de Bellman dans le cas à états continus

L'algorithme de Bellman classique peut facilement être étendu au cas où les états sont continus. Formellement on ne peut plus parler de graphe états-étapes mais la structure de l'algorithme reste inchangée et la notion de sommets états-étapes reste compréhensible. On travaille toujours étape par étape et on cherche également à chaque fois à déterminer le potentiel pour tous les états possibles. Comme les états sont continus cela se traduit par la détermination à chaque étape de la fonction potentielle  $V(X)$ ,  $X$  parcourant l'ensemble des états possibles. La fonction  $V$  est donc continue alors qu'elle est discrète lorsque les états sont discrets. La détermination de la fonction  $V$  à chaque étape se fait toujours à partir de l'équation Hamilton-Jacobi-Bellman qui devient un minimum sur un ensemble continu au lieu d'être un minimum sur un ensemble discret.

### III.5.e) Résolution de l'équation de Hamilton-Jacobi-Bellman (HJB)

Pour résoudre le problème de programmation dynamique il faut alors résoudre l'équation HJB à chaque période. Nous allons maintenant étudier la fonction de transition (notée  $C$  dans l'expression mathématique de l'équation ci-dessus). Elle comporte plusieurs composantes que nous allons détailler.

#### i) Composante coût d'exploitation

La composante du coût de d'exploitation comprend exclusivement la consommation de carburant. Etant donné les hypothèses effectuées sur la consommation de carburant en

fonction de la vitesse, la composante de la fonction de transition correspondant au coût d'exploitation entre deux secteurs consécutifs est linéaire par morceaux (voir Figure 23) et atteint son minimum pour la durée nominale de parcours.

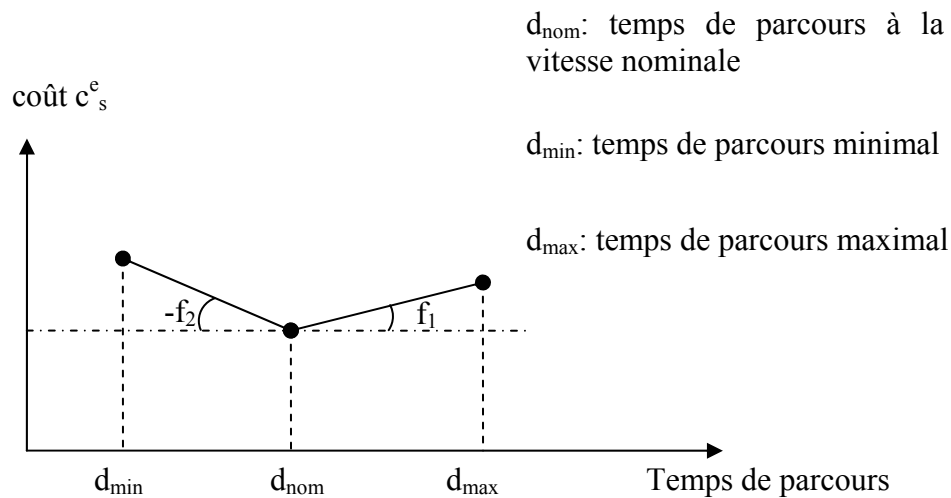


Figure 23) Composante consommation de la fonction de transition entre deux secteurs consécutifs

ii) Composante correspondant à l'utilisation de la capacité  $(-\sum_{s \in S} \sum_{p \in P} \delta_{f,r,s,p} \widehat{\pi}_{s,p})$

La composante correspondant à l'utilisation de capacité est facile à évaluer : c'est une fonction en escalier dépendant de la date d'arrivée au secteur mais pas du temps de parcours pour y arriver. Pour chaque date d'arrivée  $t$  possible à un secteur  $s$ , il faut soustraire les valeurs duales correspondant aux secteurs-périodes  $\{s ; p\}$  où  $t$  est inclus dans la période  $p$ . La structure de la composante correspondant à l'utilisation de la capacité est tracée dans la Figure 24.

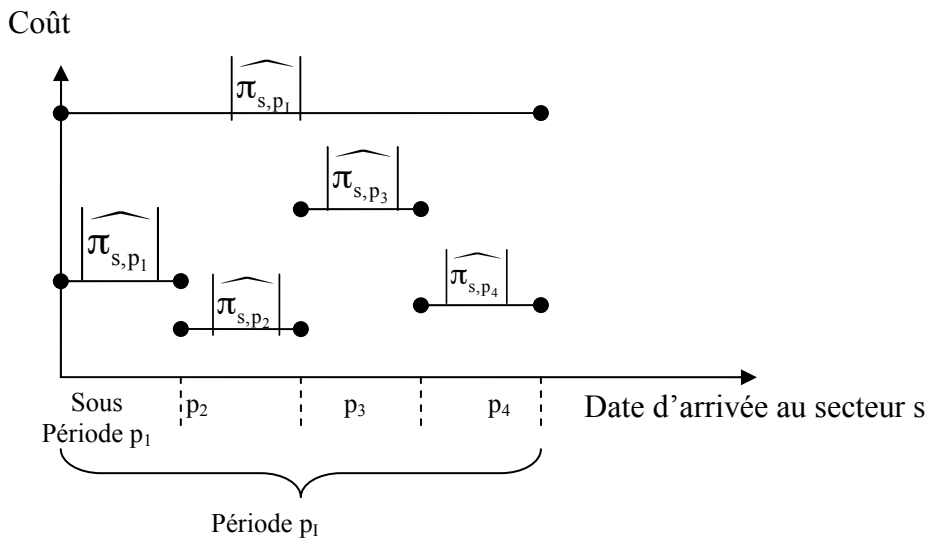


Figure 24) Composante « utilisation de la capacité » de la fonction de transition : en fonction de l'heure d'arrivée dans un secteur.

iii) Composante liée au retard à l'arrivée

Enfin la dernière composante de la fonction coût n'intervient qu'à la dernière étape lorsque l'on arrive à l'aéroport de destination et c'est une fonction linéaire par morceaux. Elle comprend la composante due au retard à l'arrivée et la composante de propagation des retards aux vols en correspondance. De la même manière elle dépend de la date d'arrivée à l'aéroport mais pas du temps de parcours pour y arriver. La pente du segment est  $\sum_{g \in C_f} |\widehat{\pi}_{f,g}| + c_{f,r}^a$ . Elle est tracée dans la Figure 25.

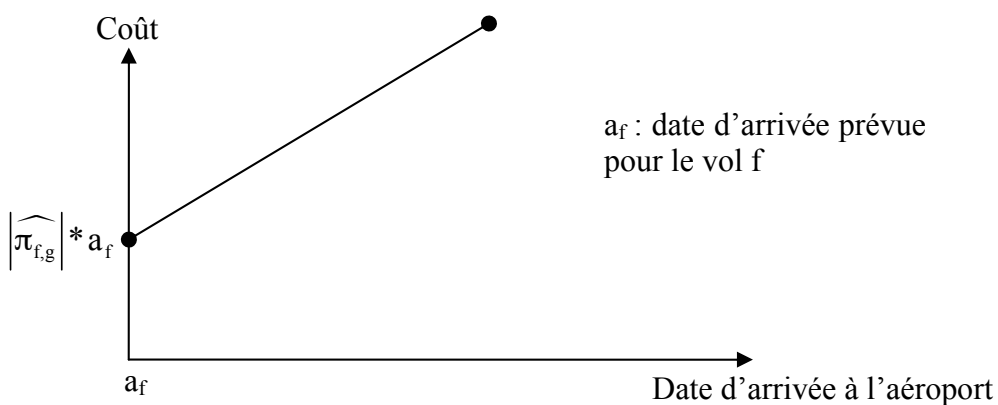


Figure 25) Composante dépendant de la date d'arrivée de la fonction de transition à la dernière étape

iv) *Expression de l'équation HJB dans notre contexte*

On note  $t_s$  une date d'arrivée au secteur  $s$ . Le potentiel de  $t_s$  s'écrit grâce à l'équation HJB.

$$V(t_s) = \underset{t_{s-1} \text{ prédécesseur de } t_s}{\text{Min}} (V(t_{s-1}) + c_s^e(t_s - t_{s-1})) - \sum_{t_s \in \mathcal{P}} \widehat{\pi}_{s,p}$$

$t_{s-1}$  est prédécesseur de  $t_s$  si  $d_{s \text{ min}} \leq t_s - t_{s-1} \leq d_{s \text{ max}}$

On remarque que le troisième membre ne dépend pas de  $t_{s-1}$  : on peut donc le sortir du minimum :

$$V(t_s) = \underset{t_{s-1} \text{ prédécesseur de } t_s}{\text{Min}} (V(t_{s-1}) + c_s^e(t_s - t_{s-1})) - \sum_{t_s \in \mathcal{P}} \widehat{\pi}_{s,p}$$

L'aéroport de destination est géré comme les secteurs en ce qui concerne les contraintes de capacité et mathématiquement il fait donc partie de l'ensemble des secteurs pour lesquels on assure le respect de la capacité. On note ce secteur particulier  $sd$ . L'équation HJB à l'aéroport d'arrivée s'écrit alors :

$$V(t_{sd}) = \underset{t_{sd-1} \text{ prédécesseur de } t_{sd}}{\text{Min}} (V(t_{sd-1}) + c_{sd}^e(t_{sd} - t_{sd-1})) - \sum_{t_{sd} \in \mathcal{P}} \widehat{\pi}_{sd,p} - \sum_{g \in C_f} t_{sd} \widehat{\pi}_{f,g} + c_f^a * (t_{sd} - a_f)$$

De même que précédemment, les trois dernières composantes ne dépendent pas de  $t_{sd-1}$  : on peut les sortir du minimum :

$$V(t_{sd}) = \underset{t_{sd-1} \text{ prédécesseur de } t_{sd}}{\text{Min}} (V(t_{sd-1}) + c_{sd}^e(t_{sd} - t_{sd-1})) - \sum_{t_{sd} \in \mathcal{P}} \widehat{\pi}_{sd,p} - \sum_{g \in C_f} t_{sd} \widehat{\pi}_{f,g} + c_f^a * (t_{sd} - a_f)$$

Dans tous les cas, pour résoudre l'équation HJB, il suffit donc de calculer ce minimum :

$$\underset{d_{s \text{ min}} \leq t_s - t_{s-1} \leq d_{s \text{ max}}}{\text{Min}} (V(t_{s-1}) + c_s^e(t_s - t_{s-1}))$$

v) *Résolution de l'équation HJB pour une étape*

$c_s^e(t_s - t_{s-1})$  est une fonction linéaire par morceaux composée de deux segments (voir Figure 23). On a montré (les calculs sont détaillés en Annexes) que la fonction  $V(t_s) = \underset{d_{s \text{ min}} \leq t_s - t_{s-1} \leq d_{s \text{ max}}}{\text{Min}} (V(t_{s-1}) + c_s^e(t_s - t_{s-1}))$  est continue, linéaire par morceaux et composée de trois segments lorsque  $V(t_{s-1})$  est un segment de pente quelconque. Lorsque  $V(t_{s-1})$  est un segment dont la pente est égale à celle d'un des segments de la fonction  $c_s^e(t_s - t_{s-1})$  alors  $V(t_s)$  n'est composée que deux segments de même pente que  $c_s^e(t_s - t_{s-1})$ . Afin de faciliter la lecture des résultats, on a tout d'abord résumé dans la Figure 26 la signification de la résolution de l'équation HJB dans le processus de la programmation dynamique : déterminer le potentiel à une étape donnée en connaissant le potentiel à l'étape précédente et la fonction de transition. Les résultats précis sont exposés dans deux figures (Figure 27 et Figure 28) qui comprennent chacune l'expression du potentiel à l'étape  $n-1$  et celle à l'étape  $n$ . Le calcul a été effectué pour la même fonction de transition  $c_s^e(t_s - t_{s-1})$ .



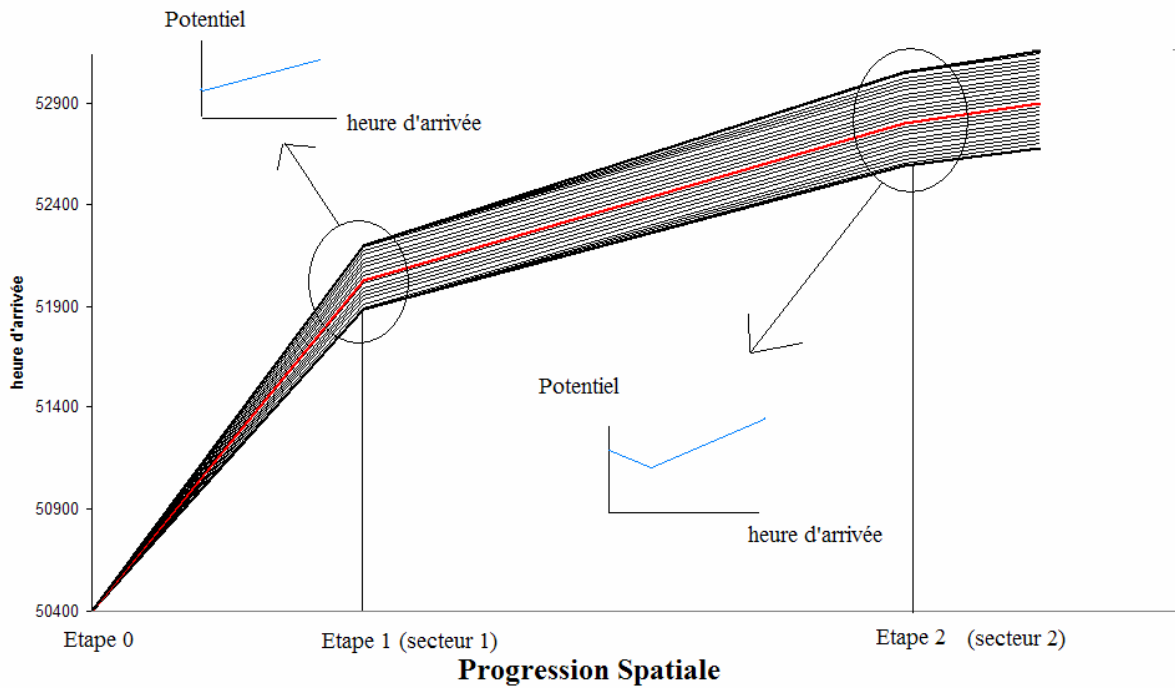


Figure 26) Résolution de l'équation HJB : potentiel à l'étape 2 en fonction du potentiel à l'étape 1 et de la fonction de transition

- - - - ● Potentiel V à l'étape n-1
- — ● Potentiel V à l'étape n
- En italique : prédécesseur optimal*

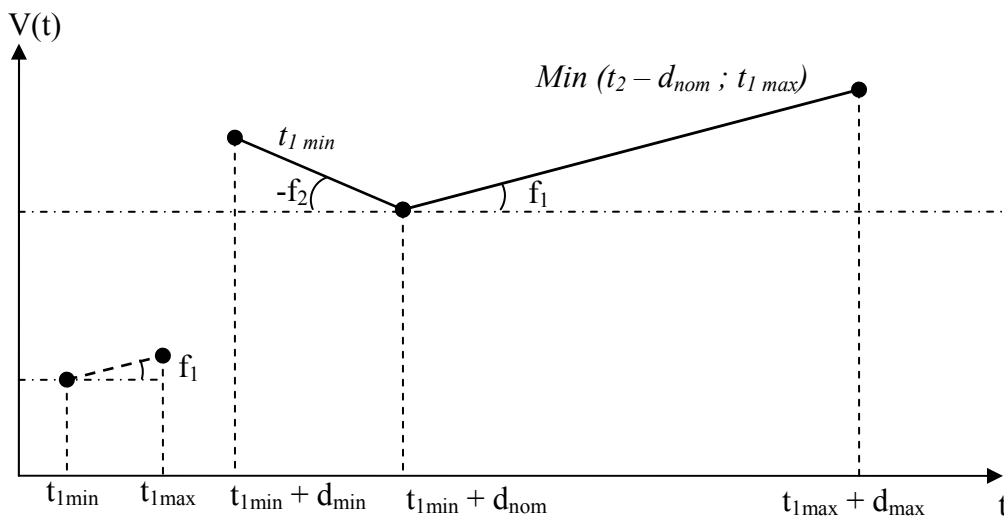


Figure 27) Solution de l'équation HJB pour une fonction potentiel segment de pente  $f_1$  à la période s-1 et la fonction de transition  $c_s^e$  (présentée Figure 23)

● - - - ● Potentiel V à l'étape n-1  
 ● — ● Potentiel V à l'étape n  
*En italique* : prédécesseur optimal

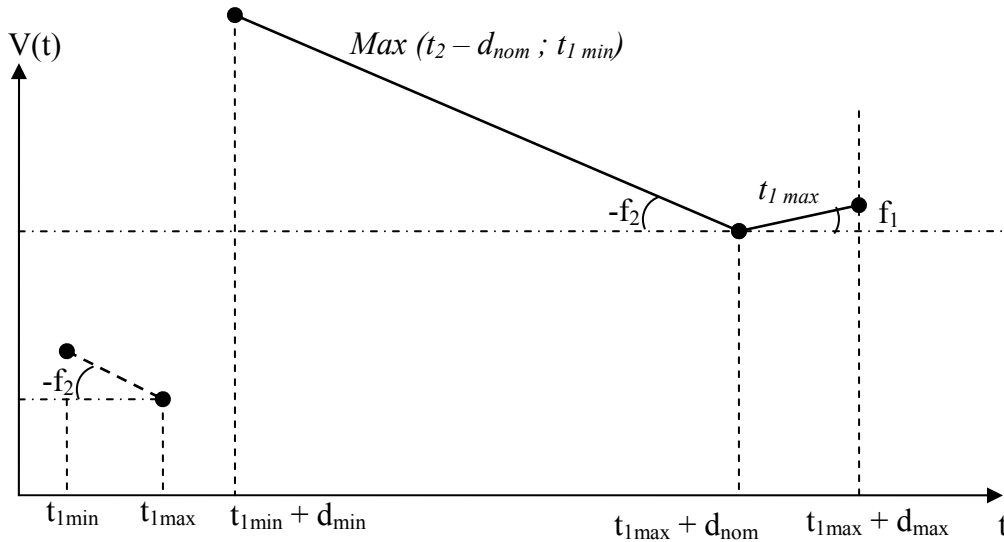


Figure 28) Résolution de l'équation HJB pour une fonction potentiel segment de pente  $-f_2$  à la période  $s-1$  et la fonction de transition  $c_s^e$  (présentée Figure 23)

vi) Première étape de l'algorithme de Bellman

L'initialisation de la méthode s'effectue par la détermination des états possibles à l'étape 0 et de la fonction potentiel associée. Ici, au départ il n'y a qu'un état possible :  $t_i$  la date initiale à partir de laquelle on cherche des trajectoires. Le potentiel initial de cette trajectoire sera alors  $-\widehat{\pi}_f$  seule partie de la fonction coût que l'on n'a pas encore considéré. Le potentiel V initial est présenté dans la Figure 29.

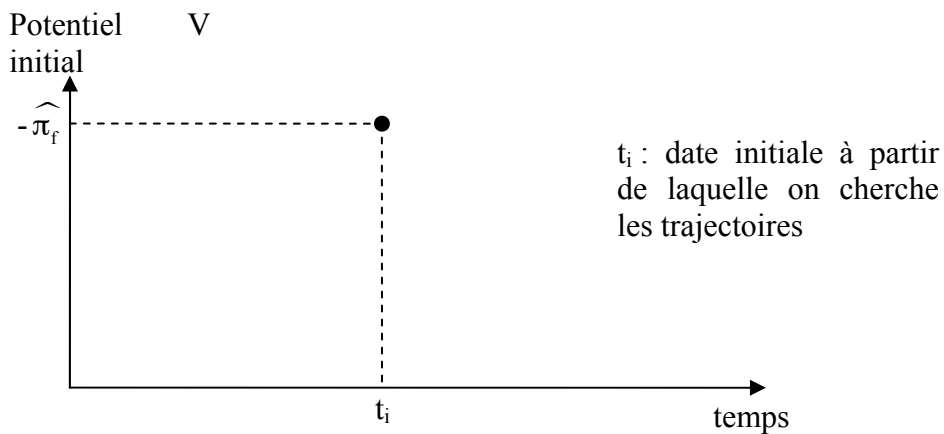
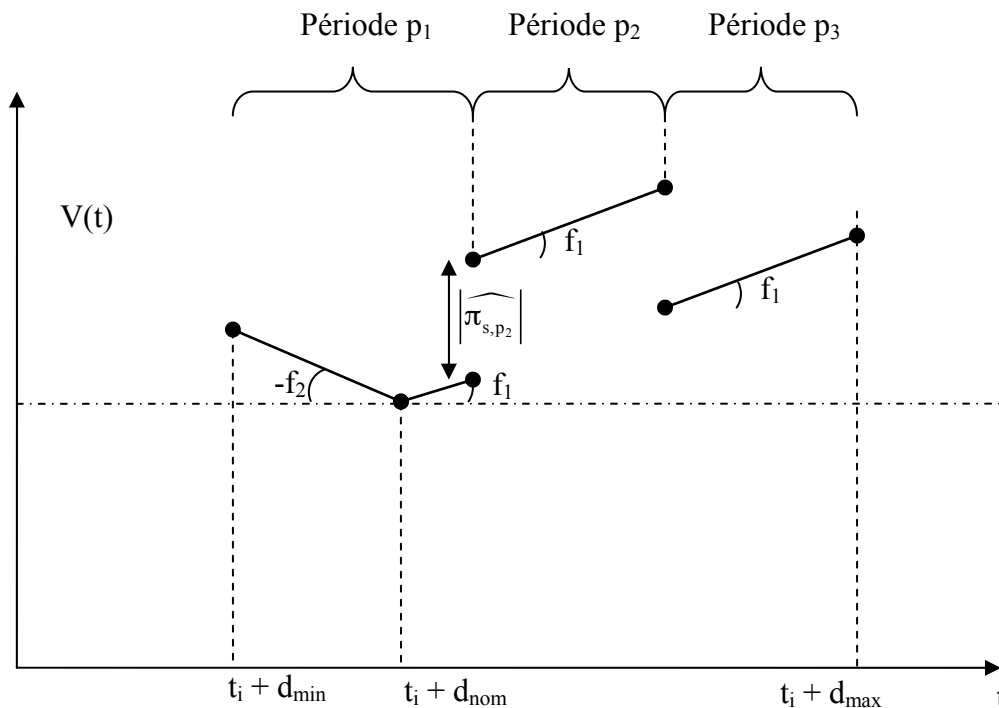


Figure 29) Fonction potentiel à la première étape : un seul état possible  $t_i$

Etant donné qu'il n'y a qu'un prédécesseur possible, le calcul de la fonction potentiel  $V(t)$  à l'étape 1 correspondant au premier croisement secteur est immédiat : consommation pour aller de la position initiale de l'avion au secteur 1 en  $t-t_i$  et soustraction des valeurs duales correspondantes à l'utilisation de capacité du secteur 1 à la date  $t$ . On a tracé l'exemple dans lequel seule la valeur duale correspondant à la période  $p_2$  est non nulle (Figure 30). La fonction potentielle est alors linéaire par morceaux puisque somme d'une fonction linéaire par morceaux et d'une fonction en escalier. On peut remarquer de plus qu'il n'y a que deux pentes possibles pour chaque morceau :  $-f_2$  et  $f_1$ .



**Figure 30) Fonction potentiel  $V(t)$  à l'étape 1 (premier croisement secteur) lorsque seule la valeur duale correspondant à la période  $p_2$  est non nulle.**

vii) *Etapas suivantes de l'algorithme de Bellman*

Pour calculer le potentiel à l'étape suivante, on va considérer la fonction potentiel morceau par morceau. Pour chaque morceau (qui est un segment de droite de pente  $-f_2$  ou  $f_1$ ) on va calculer la fonction potentiel à l'étape suivante en appliquant les formules présentées dans la Figure 27 et la Figure 28. Pour obtenir la fonction potentiel complète il faut alors garder le minimum d'un ensemble de fonctions linéaires par morceaux dont on connaît toutes les caractéristiques, ce qui ne présente pas de difficultés. Il ne reste plus qu'à ajouter la fonction en escalier correspondant à l'utilisation de capacité et on a alors déterminé entièrement la fonction potentiel à la nouvelle étape. On démontre facilement par récurrence que la fonction potentiel à n'importe quelle étape, exceptée la dernière, est une fonction linéaire par morceaux et dont les pentes sont toujours  $-f_2$  ou  $f_1$ . On peut donc calculer le potentiel à chaque étape de la même manière. A la dernière étape cependant il faut de plus rajouter la fonction linéaire correspondant aux pénalités de retard et de propagation des retards. La fonction potentiel obtenue à la fin est également linéaire par morceaux, son minimum est donc atteint à un point de discontinuité et est facile à obtenir. Tout au long du processus on a maintenu en mémoire

le prédécesseur optimal de chaque état et on peut donc retrouver la solution optimale en remontant de proche en proche.

### III.5.f) Problème de la consommation et exemples de résultats

On présente dans la Figure 31 un exemple réel, tiré des tests, de la fonction coût réduit à l'arrivée en fonction de la date d'arrivée. Une fois le minimum repéré on remonte de proche en proche pour obtenir les caractéristiques de la fonction optimale, présentées dans cet exemple dans la Figure 32.

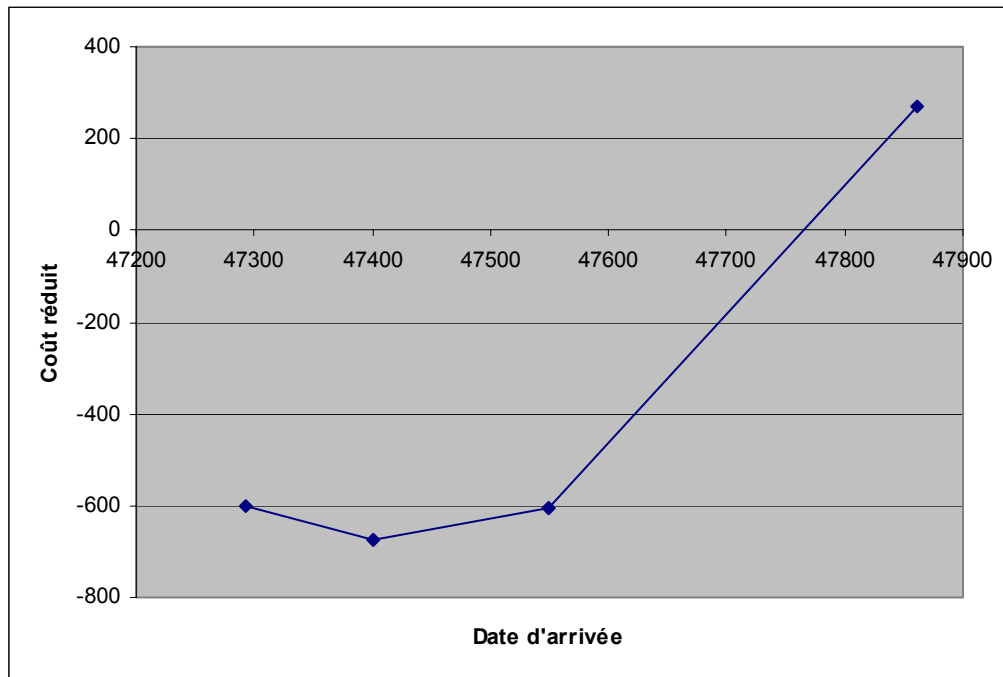


Figure 31) Fonction coût réduit à l'aéroport d'arrivée en fonction de la date d'arrivée

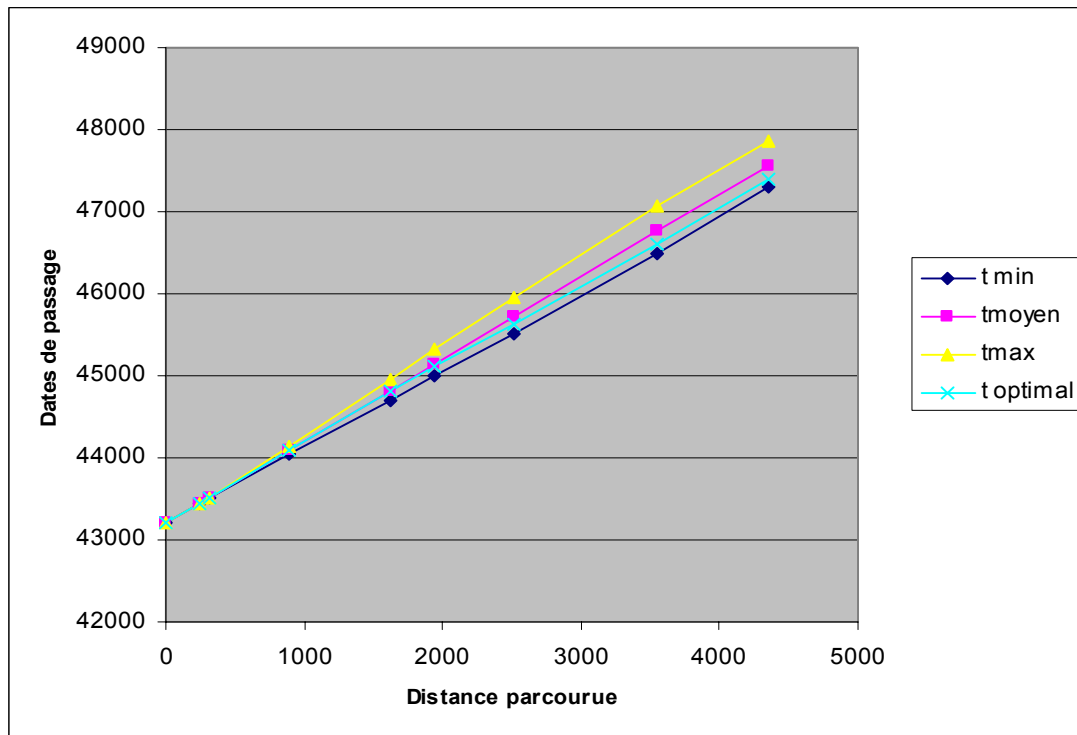


Figure 32) Dates de passages aux secteurs correspondant au minimum du coût réduit à l'arrivée

Si ce minimum correspond à une consommation inférieure à la réserve de carburant initiale, le problème est résolu et l'algorithme de programmation dynamique s'arrête ici pour cette instance. Il peut arriver cependant que le minimum de coût réduit corresponde à une consommation supérieure à la réserve de carburant et que la trajectoire ne soit donc pas valide. Il s'agit alors de trouver la trajectoire de coût réduit minimum mais vérifiant de plus la contrainte sur la réserve de carburant.

Sur chaque segment de la fonction coût réduit obtenue à la fin de l'algorithme de programmation dynamique (telle celle présentée Figure 31), la consommation est linéaire. Pour un segment donné il est donc facile de trouver la trajectoire optimale vérifiant la contrainte de carburant. Dans le cas où le minimum de coût réduit ne correspond pas à une consommation admissible, on va chercher pour chaque segment la trajectoire optimale vérifiant la contrainte de carburant puis ne garder que la meilleure si elle existe. Si enfin on ne trouve pas une trajectoire réalisable, il reste une incertitude pour savoir s'il existe une trajectoire réalisable de coût réduit négatif. Etant donné les tests effectués qui montrent que les échecs pour trouver des trajectoires réalisables pour des trajectoires 3D sont très majoritairement dus au coût réduit, nous avons décidé d'accepter cette petite incertitude et la perte d'optimalité associée. Le temps de calcul nécessaire à la résolution de cette question est disproportionné par rapport au gain attendu.

### III.5.g) Conclusion sur la résolution de la deuxième phase par programmation dynamique

Le fait de travailler sur la fonction potentiel morceau par morceau réduit grandement la complexité de l'algorithme puisque le nombre de ceux-ci est limité : à partir d'un segment de la fonction potentiel à une étape donnée, on n'en crée que deux à l'étape suivante qui souvent vont prolonger les segments adjacents. De plus, la fonction en escalier représentant l'utilisation de capacité présente un nombre de sauts limité aux périodes pour lesquelles on

fait respecter la capacité : 4 par heure dans nos applications numériques. On avait dans un premier temps résolu le problème de programmation dynamique en discrétisant le temps de manière à se placer dans le cas à états discrets. On calculait alors le potentiel d'un sommet en énumérant tous ses sommets prédécesseurs mais cette méthode a été abandonnée devant les temps de calcul trop lourds. Il est enfin à noter que la méthode de résolution serait toujours valable pour une fonction de consommation en fonction de la vitesse plus proche de la réalité. La résolution de l'équation de Hamilton-Jacobi-Bellman serait certes plus compliquée mais la complexité de l'algorithme resterait inchangée. Le poids de la phase de programmation dynamique dans le temps de calcul des trajectoires est évalué dans le chapitre suivant qui est consacré à la mise en œuvre informatique des algorithmes présentés jusqu'ici, mais ce temps de calcul est négligeable par rapport au temps de calcul nécessité par la phase I de l'algorithme.



## chapitre V Mise en œuvre expérimentale et résultats

*Ce chapitre est consacré à la description de la mise en œuvre expérimentale du processus global de résolution. On présente tout d'abord le contexte général de l'expérimentation ainsi que les objectifs poursuivis. Une importante partie décrit les tests effectués sur l'algorithme de résolution du sous problème en insistant notamment sur la détermination des paramètres régissant cet algorithme. On s'attaque ensuite à la résolution globale du problème par génération de colonnes. L'obtention d'une solution entière de bonne qualité par la résolution en nombres entiers du dernier problème maître restreint nous dispense d'appliquer la méthode prévue basée sur le branch-and-price. En revanche la persistance de trajectoires artificielles dans la solution optimale nous conduit à étudier plus précisément dans la dernière partie de ce chapitre les causes et les conséquences de leur présence.*

### I. Objectifs de l'expérimentation et vue d'ensemble

#### I.1. Objectifs de la mise en œuvre expérimentale

##### I.1.a) Objectifs

Le but premier de la mise en œuvre expérimentale est de déterminer si et dans quelles conditions la méthode basée sur la génération de colonnes développée au cours de ce travail peut être appliquée pour résoudre le problème d'optimisation qui apparaît dans le cadre de la régulation court terme du trafic aérien. A cet objectif global il faut ajouter des objectifs internes de détermination des paramètres propres à chaque algorithme.

##### *i) Objectifs internes*

- Sous problème de tarification

L'algorithme de résolution du sous problème de tarification dispose de deux paramètres principaux qui ont été décrits dans le chapitre précédent (voir le paragraphe III.4 du chapitre IV p.94) : la structure de données utilisée pour stocker les feuilles d'espace au cours de l'algorithme et le label qui permet de comparer deux trajectoires partielles entre elles. En ce qui concerne la résolution du sous problème de tarification les tests ont alors pour objectif de déterminer les couples {structure de données ; label} qui donnent les meilleurs résultats en termes de temps de calcul et de nouvelles trajectoires trouvées en fonction du contexte dans lequel l'algorithme a été appelé. La recherche de trajectoires pouvant être interrompue avant la fin de l'algorithme lorsque l'on dépasse un certain temps de calcul ou un nombre fixé de nouvelles trajectoires construites, l'influence de ces nombres limites doit donc également être étudiée.

- Itérations de génération de colonnes

En ce qui concerne l'algorithme global de génération de colonnes, le paramètre principal à étudier est le nombre de boucles de la génération de colonnes effectuées avant d'arrêter l'algorithme lorsqu'il ne s'arrête pas de lui-même à l'optimalité en un temps de calcul raisonnable. L'efficacité des bornes inférieures est donc également étudiée dans ce cadre. De plus, l'initialisation du processus se fait à l'aide de variables artificielles selon la technique du grand M. La présence de variables artificielles dans la solution finale doit être analysée en fonction notamment des caractéristiques de ces trajectoires artificielles.



- Branch-and-bound

Le processus d'arrondi par branch-and-bound ou par heuristique pourrait comporter de nombreux paramètres à étudier et déterminer. Cependant les premiers résultats donnant des solutions relaxées peu fractionnaires, il a été décidé de porter l'effort principal d'expérimentation sur les autres « briques » du processus de résolution (sous problème de tarification et génération de colonnes) et d'utiliser si nécessaire une règle de branchement inspirée d'une règle éprouvée : la règle de Ryan et Foster ([44]). La comparaison des différentes règles de branchement évoquées dans le paragraphe consacré à ce sujet (chapitre III paragraphe V.2.a) représente une étude à part entière mais est similaire à d'autres travaux déjà effectués sur ce sujet ([41] par exemple) alors que l'algorithme de résolution du sous problème de tarification étant totalement caractéristique de notre problème, on ne peut pas se rapporter à d'autres travaux existants pour son étude.

- ii) *Objectifs externes*

Les expériences doivent permettre d'évaluer l'efficacité de l'algorithme dans des conditions diverses représentant différents états du trafic aérien en fonction notamment de l'heure et de la date considérée. Il faut étudier son comportement dans des situations de charge normales ou exacerbées. Les critères à examiner sont alors le coût global de la solution obtenue par rapport à la situation initiale, la faculté à supprimer les surcharges et les temps de calcul nécessaires pour obtenir une solution réalisable.

- I.1.b) Philosophie de l'expérimentation

L'objectif principal de l'expérimentation étant de vérifier si l'algorithme permet de résoudre le problème d'optimisation posé dans un contexte opérationnel, les tests ont été effectués dans des conditions aussi proches que possible des conditions réelles sans simplification abusive dénaturant le problème. Toutes les données disponibles ont été utilisées afin de représenter fidèlement le système aérien à réguler. Les instances étudiées représentent des journées de trafic réelles sur le territoire européen géré par la CFMU.

- I.2. Contexte de la mise en œuvre expérimentale

- I.2.a) Contexte et données disponibles

- i) *L'« atelier aérien »*

Le laboratoire LICIT dispose d'un environnement logiciel appelé « atelier aérien » qui rassemble un simulateur de trafic aérien, des données réelles sur le système aérien et des logiciels développés par le LICIT pour évaluer les différentes méthodes de régulation et d'optimisation sur lesquelles l'équipe du laboratoire travaille. Les tests pour évaluer la méthode de génération de colonnes ont logiquement été effectués dans cet environnement.

## *ii) Les données disponibles utilisées au cours de l'expérimentation*

Les données utilisées au cours de ces tests sont des données réelles fournies par la CFMU, elles sont regroupées par AIRAC<sup>11</sup>, les données des tests proviennent de l'AIRAC 247 couvrant la période allant du 04 septembre 2003 au premier octobre 2003 inclus. Les données utilisées au cours des tests peuvent être regroupées en trois sous ensembles : les données sur les performances des avions (consommation, taux de montée, vitesse...), les données sur le réseau aérien européen (routes aériennes, niveaux de vol, ouvertures et fermetures des routes et secteurs, balises, secteurs, capacités, aéroports...) et les données sur le trafic aérien (plans de vols déposés, régulés et observés).

## *iii) Détermination des correspondances d'un vol*

Pour prendre en compte le coût des retards à l'arrivée on a besoin d'informations sur les correspondances. Aucune des correspondances (avion, équipage, passagers) ne fait partie des données disponibles directement : il faut donc les obtenir par d'autres sources ou les évaluer à partir des données disponibles. Une première possibilité consiste à se placer dans une approche CDM (Collaborative Decision Making voir note 6 p.19) du partage d'informations sur les correspondances, c'est-à-dire supposer l'existence d'une plateforme par laquelle les compagnies pourraient en direct indiquer les correspondances des vols concernés. Si une telle structure est imaginable, elle n'existe pas réellement et la collecte et l'exploitation de telles données est très délicate. Par ailleurs les passagers organisent souvent eux-mêmes des correspondances multi-compagnies qui n'apparaîtraient pas de cette manière et il faudrait envisager d'autres méthodes, notamment économiques, pour en évaluer le coût.

Une autre manière d'estimer les correspondances à partir de données existantes et disponibles consiste à considérer que tout avion de la même compagnie partant de l'aéroport d'arrivée d'un vol f dans une plage de temps commençant à l'arrivée prévue du vol f et durant par exemple deux heures sont des correspondances potentielles du vol f et doivent être pris en compte dans le calcul de coût de retard à l'arrivée. On a procédé de cette façon pour la première implémentation tout en sachant que si des informations plus précises sur les correspondances existent, elles pourront facilement être intégrées dans le schéma d'expérimentation sans modifier le schéma de résolution.

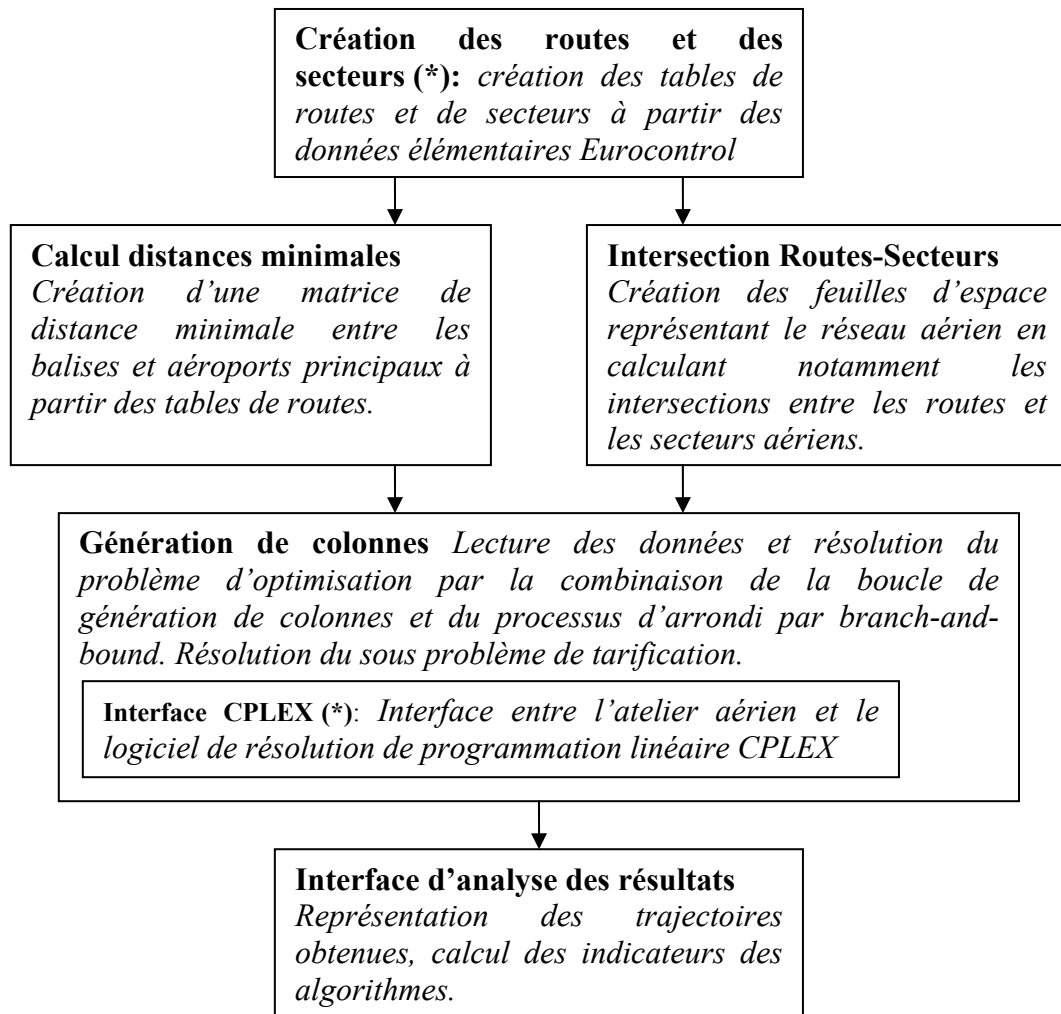
## **1.3. Vue d'ensemble sur la mise en œuvre expérimentale**

### **1.3.a) Introduction**

L'ensemble des logiciels réalisés pour la mise en œuvre expérimentale des algorithmes développés durant ce travail représente environ 20 000 lignes de code, très majoritairement en C++ (le SQL ayant été utilisé pour la manipulation des bases de données), sans compter le cadre général que constitue la plate forme de programmation du LICIT (l'atelier aérien). Ce paragraphe a pour but de donner une vue d'ensemble des programmes informatiques écrits dans le cadre du travail de thèse en soulignant notamment le travail périphérique de modélisation du système aérien et de mise en forme des données fournies par Eurocontrol. Le récapitulatif de l'architecture logicielle réalisée est exposé dans la Figure 33.

---

<sup>11</sup> AIRAC : Aeronautical Information, Regulation and Control : l'AIRAC est un système normalisé de diffusion d'informations aéronautiques avec préavis de mise en vigueur. Les AIRAC sont des cycles de 28 jours et sont numérotés par ordre croissant



(\*) Logiciels partiellement effectués dans le cadre de la maintenance informatique de l'atelier aérien

Figure 33) Vue d'ensemble des logiciels réalisés dans le cadre de l'expérimentation

### I.3.b) Les précalculs et les calculs périphériques de modélisation

#### i) Modélisation en feuilles d'espace du réseau aérien

Le premier travail préliminaire a consisté à agréger les tables de données élémentaires fournies par Eurocontrol (espaces aériens élémentaires, composition des secteurs, structure des espaces aériens...) en des tables utilisables directement : tables de routes comportant la catégorie de la route (dont dépend la disponibilité), les niveaux de vol autorisés, les balises concernées, les dates d'ouverture et de fermeture ; table de secteurs comportant la délimitation spatiale, la nature, les capacités en fonction du temps des différents secteurs. Ce travail a notamment été effectué dans le cadre de la maintenance informatique de l'atelier aérien. Une fois ces tables obtenues, le premier bloc logiciel construit les feuilles d'espace (chapitre IV paragraphe II.1.b) p. 83) en effectuant notamment un calcul assez technique des intersections entre les routes aériennes et les secteurs aériens. Ce logiciel comporte également un module de sauvegarde « en dur » des feuilles obtenues afin de faciliter les phases de

programmation suivantes. Pour la journée du 10 septembre 2003 qui sert d'instance de test principale, le réseau aérien est ainsi représenté par 21 520 feuilles d'espace.

ii) *Précalculs : distances minimales entre les balises et les aéroports*

Dans le cadre de la résolution du sous problème de tarification, on utilise une borne inférieure de la distance entre chaque balise et chaque aéroport principal en suivant les routes aériennes. Cette borne inférieure permet d'avoir une estimation précise de la consommation de carburant nécessaire pour rallier l'aéroport de destination et permet également le calcul d'une fourchette de dates d'arrivée à l'aéroport. On calcule ainsi au cours de la construction des trajectoires une borne inférieure précise de la consommation de carburant et du coût réduit des trajectoires partielles. On peut alors couper l'exploration des trajectoires dans le cas où la consommation minimale excède la réserve de carburant ou si le coût réduit devient positif. Plus la distance minimale entre les balises et l'aéroport de destination est précise et plus vite on peut couper des trajectoires partielles stériles. On diminue ainsi le temps de calcul nécessaire à la résolution du sous problème de tarification.

Le deuxième bloc logiciel calcule donc à partir des tables de routes un graphe orienté classique en deux dimensions dont les sommets représentent les balises aériennes. Deux balises sont reliées par un arc si elles sont consécutives sur une route aérienne. Le poids d'un arc représente la distance entre les deux balises en suivant la route aérienne. Le calcul des distances minimales entre les balises et les aéroports principaux s'effectue ensuite à l'aide d'un algorithme classique de Dijkstra, les distances minimales ainsi calculées sont alors intégrées aux feuilles d'espace.

I.3.c) Résolution globale

Les blocs logiciels précédents ont préparé le bloc principal qui code la résolution globale du problème d'optimisation par la boucle de génération de colonnes. Celle-ci est imbriquée dans le processus de branch-and-bound et cette partie qui représente la base de la méthode est assez restreinte, la boucle ne présentant pas de difficultés particulières si ce n'est, d'un point de vue purement technique, l'interface avec le logiciel commercial CPLEX utilisé pour résoudre le programme linéaire à chaque boucle.

La résolution du sous problème de tarification est le cœur réel de la résolution et les difficultés tant algorithmiques que techniques y sont les plus nombreuses. Le logiciel remplissant ce rôle représente environ la moitié du programme complet, contre un quart pour les calculs préliminaires et un quart pour les autres fonctions. Pour conclure cette présentation sommaire, une petite interface d'analyse des résultats a été programmée permettant de visualiser les trajectoires calculées et de calculer les différents indicateurs sur la résolution du sous problème de tarification et du problème global.

## **II. Résolution du sous problème de tarification**

### **II.1. Les différents scénarios et les paramètres de l'algorithme**

#### **II.1.a) Caractéristiques influant la recherche de trajectoires alternatives**

L'algorithme de résolution du sous problème de tarification doit trouver le plus rapidement possible un ensemble de trajectoires alternatives qui seront ajoutées au programme maître restreint de manière à faire baisser la valeur de la fonction objectif. Il est utile pour

comprendre ce qui va suivre d'avoir en tête la structure de coût réduit d'une trajectoire que nous rappelons donc ici :

$$c'_{f,r} = c_{f,r} - \widehat{\pi}_f - \sum_{s \in S} \sum_{p \in P} \delta_{f,r,s,p} \widehat{\pi}_{s,p} - \sum_{g \in C_f} a_{f,r} \widehat{\pi}_{f,g}$$

$c_{f,r}$  représente le coût réel de la trajectoire (consommation et coût direct du retard)

$\widehat{\pi}_f$  est la valeur de la variable duale associée à la contrainte d'unicité de trajectoire du vol  $f$ ,  $\widehat{\pi}_{s,p}$  celles associées aux contraintes de capacité et  $\widehat{\pi}_{f,g}$  celles associées aux contraintes de propagation des retards.

Deux caractéristiques principales vont alors influencer sur la difficulté à remplir cette tâche :

- L'état de charge du réseau

Pour un réseau loin de la saturation, peu de secteurs-périodes sont utilisés au maximum de leur capacité et ainsi peu d'entre eux auront un coût réduit associé ( $\widehat{\pi}_{s,p}$ ) non nul. Le coût réduit des trajectoires ne dépendra alors principalement que de la consommation et de la date d'arrivée à l'aéroport. Dans ces conditions les trajectoires optimales seront parmi les plus directes et l'algorithme de recherche de trajectoires n'aura pas besoin d'explorer les trajectoires plus éloignées. Dans le cas où le réseau est surchargé, il est possible que les trajectoires les plus directes traversent des secteurs-périodes surchargés dont le coût réduit associé est élevé. Dans ces conditions les trajectoires optimales se situeront peut-être sur d'autres routes, d'autres niveaux ou avec des politiques de vitesse différentes. L'algorithme devra alors explorer des trajectoires plus variées, ce qui conduira à un arbre d'exploration plus étendu. On voit ici l'importance de tester l'algorithme et d'étudier son comportement dans des conditions de charge différentes.

- Les caractéristiques des trajectoires déjà présentes

D'une manière schématique on peut écrire que la valeur de  $\widehat{\pi}_f$  représente l'étalon ou la maille auquel on va comparer les trajectoires que l'on construit au cours de l'algorithme : aucune trajectoire avec un coût supérieur à cette valeur ne sera conservée. Ainsi plus la valeur de  $\widehat{\pi}_f$  est élevée et plus le nombre de trajectoires valides augmente, pénalisant ainsi toute recherche exhaustive. Or la valeur de  $\widehat{\pi}_f$  est directement liée aux trajectoires déjà présentes dans le problème maître restreint et le nombre et la variété de celles-ci dépendent de la phase de la génération de colonnes dans laquelle on se situe. En effet lors de la première phase il n'y a que les trajectoires initiales et éventuellement certaines trajectoires remarquables. Dans les phases suivantes il faut y ajouter toutes celles qui ont déjà été calculées lors des précédents appels du sous problème de tarification. La nature et la difficulté du problème vont donc changer en fonction du nombre de passages déjà effectués dans la boucle de génération de colonnes : les expériences doivent donc évaluer l'algorithme dans ces différentes conditions.

## II.1.b) Scénarios étudiés et démarche de tests

### i) Scénarios étudiés

Dans l'architecture globale du régulateur court terme, le processus d'optimisation n'est appelé que lorsque des saturations ont été détectées dans l'horizon de prévision (voir Figure 7

p.27). On peut alors supposer que le maximum de moyens de contrôle a été mis en œuvre et que les centres de contrôle ont dégroupé au maximum la zone de l'espace aérien située sous leur responsabilité<sup>12</sup>. Pour ces raisons l'instance de test principale choisie représente l'espace aérien européen le jeudi 10 septembre 2003 à midi. Cette instance représente 1112 vols commandables. Afin de faire varier la charge des secteurs, les tests ont été effectués pour trois valeurs de la capacité des secteurs-périodes : 100% de la capacité annoncée, puis 90% et enfin 75%. On réduit artificiellement la capacité en gardant la même instance afin de bien pouvoir évaluer les conséquences de la charge du réseau sur l'algorithme sans interférence due à un changement d'instance de test.

## *ii) Démarche des tests*

Lorsque l'on a déterminé une trajectoire, l'effort de calcul nécessaire pour en trouver de nouvelles est faible du fait de toutes les trajectoires partielles qui ont déjà été partiellement étendues et qui sont en attente d'être étendues si possible jusqu'à l'aéroport. D'un point de vue du temps de calcul, il est bien plus efficace de calculer à chaque appel du sous problème de tarification un nombre conséquent de trajectoires plutôt que d'appeler souvent le sous problème pour peu de trajectoires. Etant donné le temps de calcul irréductible à chaque appel du sous problème de tarification, la résolution ne devrait pas appeler celui-ci plus de 5 ou 6 fois. Chaque appel est donc spécifique et peut être étudié en détail.

La démarche suivie est alors la suivante : on effectue une batterie de tests de manière à déterminer les paramètres à choisir pour la première recherche de trajectoires. On effectue ensuite le même travail de tests et de détermination des paramètres à la deuxième étape de la génération de colonnes en supposant que les trajectoires de la première étape ont été calculées avec la combinaison de paramètres choisie lors de la première phase de tests. On effectue le même travail pour la troisième phase. Les tests pour la quatrième phase peuvent être beaucoup plus légers que dans les phases précédentes du fait des enseignements qu'on peut tirer des tests précédents. On trouvera dans la Figure 34 un résumé de cette démarche.

La manière dont les paramètres sont considérés ainsi que les indicateurs permettant d'évaluer une combinaison de paramètres sont traités dans le paragraphe suivant.

---

<sup>12</sup> L'espace aérien européen est divisé en FIR (Flight Information Region) dont dépendent les centres assurant le contrôle dans chacune de ces régions. La France est par exemple divisée en 5 FIR (Bordeaux, Paris, Brest, Marseille, Reims). Ces espaces sont eux-mêmes divisés en secteurs de contrôles dont la responsabilité est dévolue à une équipe de contrôle aérien. La division des espaces de contrôle en secteurs aériens est variable en fonction du temps, dépendant de la charge de trafic et du nombre de contrôleurs disponibles : la liste des secteurs activés à un moment donné est appelée configuration. La nuit lorsque le trafic est faible la configuration adoptée ne contient qu'un ou deux grands secteurs alors qu'en période de pointe les configurations contiennent de nombreux secteurs plus petits.

### Espace aérien 10/09/2003 midi

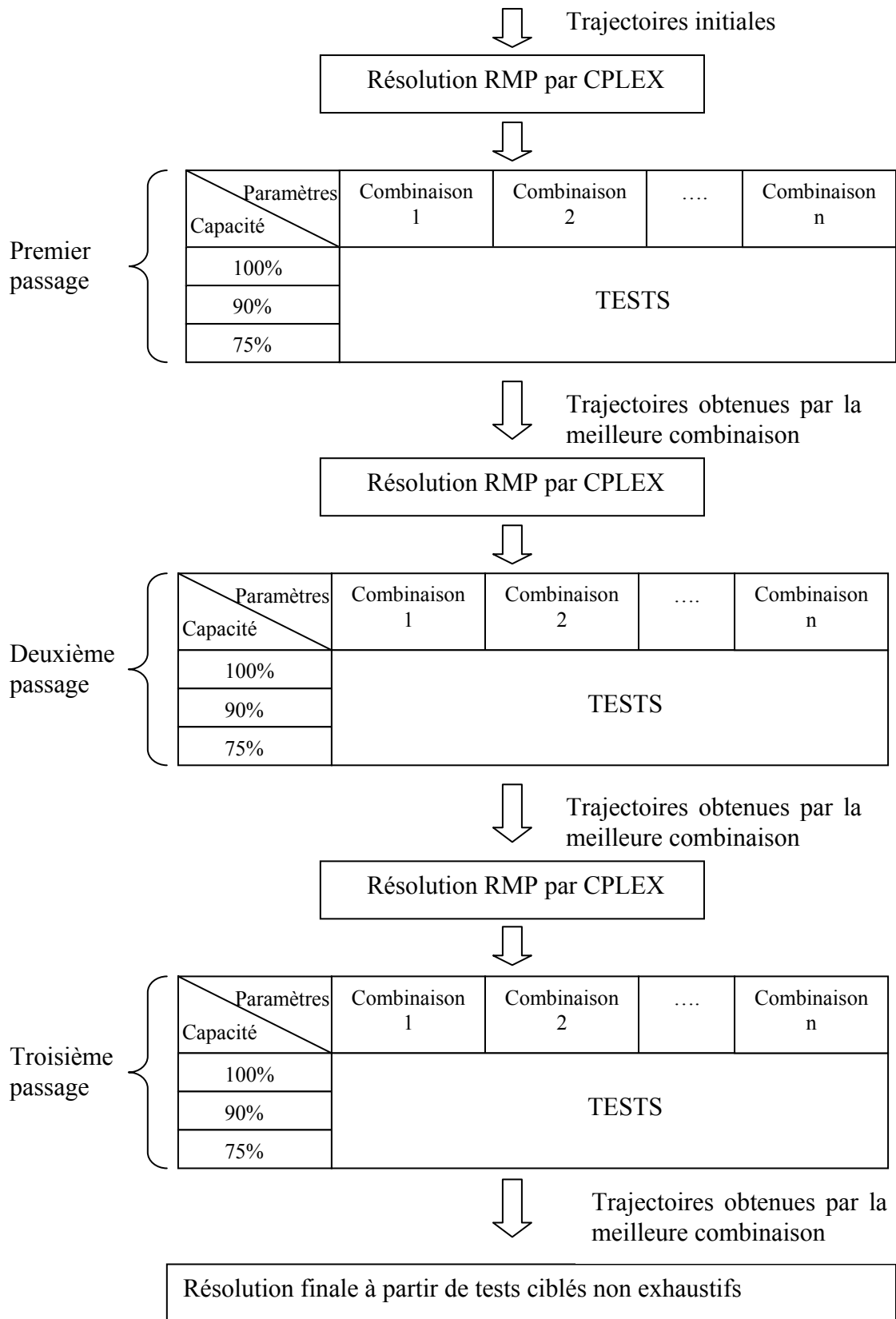


Figure 34) Démarche expérimentale de détermination des combinaisons de paramètres

## II.1.c) Les paramètres de l'algorithme

Dans le programme codant l'algorithme de résolution du sous problème de tarification il existe des paramètres internes concernant l'implémentation technique et des paramètres externes influençant directement l'algorithme. Les paramètres internes, concernant principalement la gestion mémoire des nombreuses listes dynamiques que l'algorithme doit gérer, ont été déterminés empiriquement lors de la phase de correction et de validation du programme et ne seront désormais plus évoqués ici. Les paramètres externes de l'algorithme vont principalement jouer sur l'exploration de l'arbre des possibilités.

### i) Paramètres d'exploration

Les deux premiers paramètres, la structure de données utilisée pour stocker les feuilles d'espace en attente d'être examinées et le label utilisé pour comparer deux feuilles entre elles, changent le mode d'exploration du graphe (plutôt en largeur, en profondeur, mixte etc.). On les appellera dans la suite *paramètres d'exploration*. Il y a 2 structures de données différentes (que l'on nomme piles) et 6 labels possibles, plus une structure de données ne prenant pas en compte les labels. Cela fait au total 13 combinaisons différentes.

### ii) Paramètres d'arrêt

Les deux paramètres suivants, le nombre limite de trajectoires recherchées et le temps maximal de calcul alloué à l'algorithme pour un vol donné, représentent les conditions d'arrêt de l'algorithme. On les nommera donc *paramètres d'arrêt*. Il est important de remarquer que l'arrêt se fait toujours après avoir traité toutes les trajectoires partielles d'une feuille. L'algorithme peut donc trouver plus de trajectoires que le nombre limite et parfois beaucoup plus (de l'ordre du millier de trajectoires pour un vol avec une limite de 50 trajectoires).

Le nombre de possibilités pour ces paramètres est illimité entre une et toutes les trajectoires possibles et entre une seconde et un temps de calcul illimité. Aussi, pour limiter l'explosion combinatoire des tests, il a été décidé de déterminer empiriquement et intuitivement un critère d'arrêt (50 trajectoires et 50 secondes de temps de calcul pour la première phase) avec lequel on a effectué les tests pour les 13 combinaisons de paramètres d'exploration. Une fois ces tests effectués, un premier tri est effectué parmi les combinaisons et une nouvelle batterie de tests est réalisée en changeant les critères d'arrêt et en ne considérant que les meilleures combinaisons précédemment identifiées. De cette manière on détermine une combinaison performante de tous les paramètres.

## II.2. Résultats et temps de calcul

### II.2.a) Indicateurs utilisés

Afin de pouvoir comparer deux combinaisons de trajectoires, il faut déterminer des indicateurs qui permettent d'évaluer la qualité de la résolution du sous problème. Les principaux indicateurs utilisés sont détaillés ci-dessous :

- Temps de calcul global

Temps de calcul cumulé pour la résolution du sous problème de tarification pour tous les vols commandables.

- Nombre de trajectoires nouvelles calculées



Indicateur à manipuler avec précaution puisque de nombreuses trajectoires peuvent être similaires, de plus certaines combinaisons peuvent trouver beaucoup de trajectoires pour un vol mais aucune pour un autre. Enfin ajouter un trop grand nombre de trajectoires alternatives au RMP rend sa résolution plus difficile et donc plus longue.

- Nombre de réussites

Nombre de vols pour lesquels on a calculé au moins une trajectoire nouvelle.

- Nombre d'échecs

Nombre de vols pour lesquels on n'a trouvé aucune trajectoire à la fin du temps de calcul autorisé, sans avoir démontré qu'il n'en existe pas.

- Valeur du critère à l'étape suivante

Après avoir résolu le sous problème de tarification on ajoute les trajectoires calculées au RMP que l'on résout. On connaît alors la valeur de la fonction objectif au début de l'étape suivante. La valeur du critère est donnée en équivalent kg de kérosène. Il faut cependant prendre avec précaution les chiffres d'amélioration de la fonction objectif entre deux boucles : en effet les solutions aux différentes boucles de la résolution peuvent contenir des trajectoires artificielles qui ont une consommation volontairement excessive (2 fois la réserve de carburant) afin de dissuader leur utilisation. A chaque fois que l'on remplace une trajectoire artificielle par une trajectoire réelle, on améliore grandement la fonction objectif sans avoir pour autant gagné dans la réalité la quantité de kérosène indiquée par l'amélioration de la valeur de la fonction objectif.

Ces indicateurs sont les principaux utilisés pour évaluer les combinaisons de paramètres. D'autres indicateurs peuvent être utilisés pour affiner l'analyse : place mémoire utilisée, temps de calcul consacré à la programmation dynamique, nombre de secteurs-périodes utilisés au maximum, nombre de trajectoires artificielles encore présentes et divers ratios calculés à partir de ces indicateurs. Le ratio le plus important est l'amélioration du critère divisé par le temps de calcul.

## II.2.b) Tableaux listant les instances principales de tests

Les différentes combinaisons de paramètres d'exploration ont été numérotées et sont regroupées dans le Tableau 4. La description des paramètres est disponible dans le chapitre IV paragraphe III.4. La correspondance numéro/paramètre est donnée dans le Tableau 5.

pile	capacité	100%			90%			75%		
	paramètre	passage 1	passage 2	passage 3	passage 1	passage 2	passage 3	passage 1	passage 2	passage 3
0	0	0	13	26	39	52	65	78	91	104
	1	1	14	27	40	53	66	79	92	105
	2	2	15	28	41	54	67	80	93	106
	3	3	16	29	42	55	68	81	94	107
	4	4	17	30	43	56	69	82	95	108
	5	5	18	31	44	57	70	83	96	109
1	0	6	19	32	45	58	71	84	97	110
	1									
	2									
	3									
	4									
	5									
2	0	7	20	33	46	59	72	85	98	111
	1	8	21	34	47	60	73	86	99	112
	2	9	22	35	48	61	74	87	100	113
	3	10	23	36	49	62	75	88	101	114
	4	11	24	37	50	63	76	89	102	115
	5	12	25	38	51	64	77	90	103	116

**Tableau 4) Nomenclature des combinaisons des paramètres d'exploration**

Pile 0	Pile de priorité
Pile 1	Pile Esopo-Pape
Pile 2	Variante à 2 piles de la règle Esopo-Pape
Label 0	Réserve de carburant maximale
Label 1	Maximum de la réserve de carburant moins la consommation minimale
Label 2	Distance minimale pour rejoindre l'aéroport de destination
Label 3	Coût réduit minimal des trajectoires partielles
Label 4	Estimation minimale du coût réduit des trajectoires partielles
Label 5	Date d'arrivée minimale à la feuille

**Tableau 5) Signification des différents paramètres d'exploration**

*i) Avertissement sur les résultats :*

Chaque numéro dans le tableau correspond à un test pour lequel de nombreux indicateurs ont été calculés, il n'est pas possible pour des raisons de place et de clarté d'exposer tous les résultats obtenus. On ne trouvera ici que les principaux résultats. Cette remarque est valable pour tous les tests réalisés, mais les choix ont été effectués en prenant en compte le plus possible d'indicateurs pertinents.

**II.2.c) Résultats de l'algorithme de résolution du sous problème de tarification pour la première boucle de la méthode de génération de colonnes**

*i) Détermination des paramètres d'exploration : première boucle et première évaluation des combinaisons de paramètres*

On note  $\{i,j\}$  la combinaison de la pile  $i$  avec le label  $j$ . Il y a 13 combinaisons possibles. On a fixé les paramètres d'arrêt à 50 par vol (50 trajectoires pour la limite de trajectoires et 50

secondes pour la limite de temps). En ce qui concerne le temps de calcul total, la combinaison la plus efficace est  $\{0,2\}$  avec une moyenne de 507 secondes pour les 3 instances de capacité, suivie de  $\{2,2\}$  avec 737 secondes. On passe ensuite à 1064 secondes pour  $\{0,1\}$  et 1431 secondes pour  $\{2,1\}$ . Il faut 1903 secondes à la combinaison la plus mauvaise ( $\{2,0\}$ ) pour terminer le calcul, tandis que 4 combinaisons échouent du fait d'une trop grande utilisation mémoire. Le classement est quasiment le même pour le ratio amélioration de la fonction objectif/ temps de calcul, avec une amélioration de 7735 par seconde de temps de calcul pour  $\{0,2\}$  contre 1971 seulement pour  $\{0,0\}$ .

En résumé, pour les différents critères utilisés on retrouve quasiment le classement suivant : un premier groupe composé de  $\{0,1\}$ ,  $\{0,2\}$ ,  $\{2,1\}$  et  $\{2,2\}$  dans lequel  $\{0,2\}$  est le meilleur, ensuite la combinaison  $\{1,j\}$  en 5<sup>ème</sup>, un second groupe  $\{2,0\}$ ,  $\{2,3\}$ ,  $\{2,4\}$  et  $\{2,5\}$  pour les places de 6 à 9 et enfin un dernier groupe comprenant  $\{0,0\}$ ,  $\{0,3\}$ ,  $\{0,4\}$ , et  $\{0,5\}$  systématiquement dans les 4 derniers rangs. Pour la détermination des paramètres d'arrêt de la recherche, on ne va donc considérer que les 5 premiers :  $\{0,1\}$ ,  $\{0,2\}$ ,  $\{1,j\}$ ,  $\{2,1\}$  et  $\{2,2\}$ . La combinaison d'Esopo-Pape ( $\{1,j\}$ ) est moins bonne que les 4 autres mais étant donné qu'elle est basée sur un principe différent on a décidé de pousser les tests la concernant un peu plus loin.

ii) *Détermination des paramètres d'exploration : première boucle et seconde évaluation des combinaisons de paramètres*

- Nouvelle série de tests

Afin de déterminer un couple de paramètres d'arrêt efficaces, on va tester 4 valeurs différentes pour chaque paramètre : 10, 30, 50 et 70, soit 16 possibilités (arrêt après 10 secondes de calcul ou 10 trajectoires, 10 secondes ou 30 trajectoires...). Afin d'expliquer le choix effectué pour la première phase on présente dans les trois figures suivantes (Figure 35, Figure 36 et Figure 37) l'évolution du critère en fonction du temps de calcul dans les trois instances (100%, 90% et 75% de la capacité de départ) pour différentes combinaisons de paramètres d'exploration et d'arrêts. Chaque point du graphique correspond à un test et à une combinaison de paramètres d'exploration et de paramètres d'arrêts.

- Résultats de la nouvelle série de tests

De même que dans la première série de test, la combinaison  $\{0,2\}$  est la plus efficace en termes de ratio amélioration/temps de calcul, elle surpasse toutes les autres en terme de temps de calcul et également en termes de valeur de fonction objectif. On peut cependant mettre en évidence une limite à cette combinaison : le nombre de vols auxquels a été affectée une trajectoire artificielle ne diminue pas avec l'augmentation des paramètres d'arrêt. La différence de temps de calcul et l'efficacité de la combinaison  $\{0,2\}$  nous conduisent cependant à la sélectionner pour la première phase : cela correspond à une stratégie dans laquelle on cherche le plus rapidement possible des trajectoires alternatives efficaces et plutôt directes, la recherche de trajectoires plus éloignées se faisant alors aux étapes suivantes. A noter que les autres indicateurs étudiés et non représentés ici plaident également pour la combinaison  $\{0,2\}$  et notamment le taux d'échec qui est le plus bas pour cette combinaison. On peut rappeler que la combinaison  $\{0,2\}$  correspond au choix de la feuille d'espace la plus proche possible de l'aéroport de destination parmi les feuilles contenant des trajectoires partielles à étendre. Une fois la combinaison d'exploration retenue, il faut choisir la valeur à affecter aux paramètres permettant de stopper l'exploration.

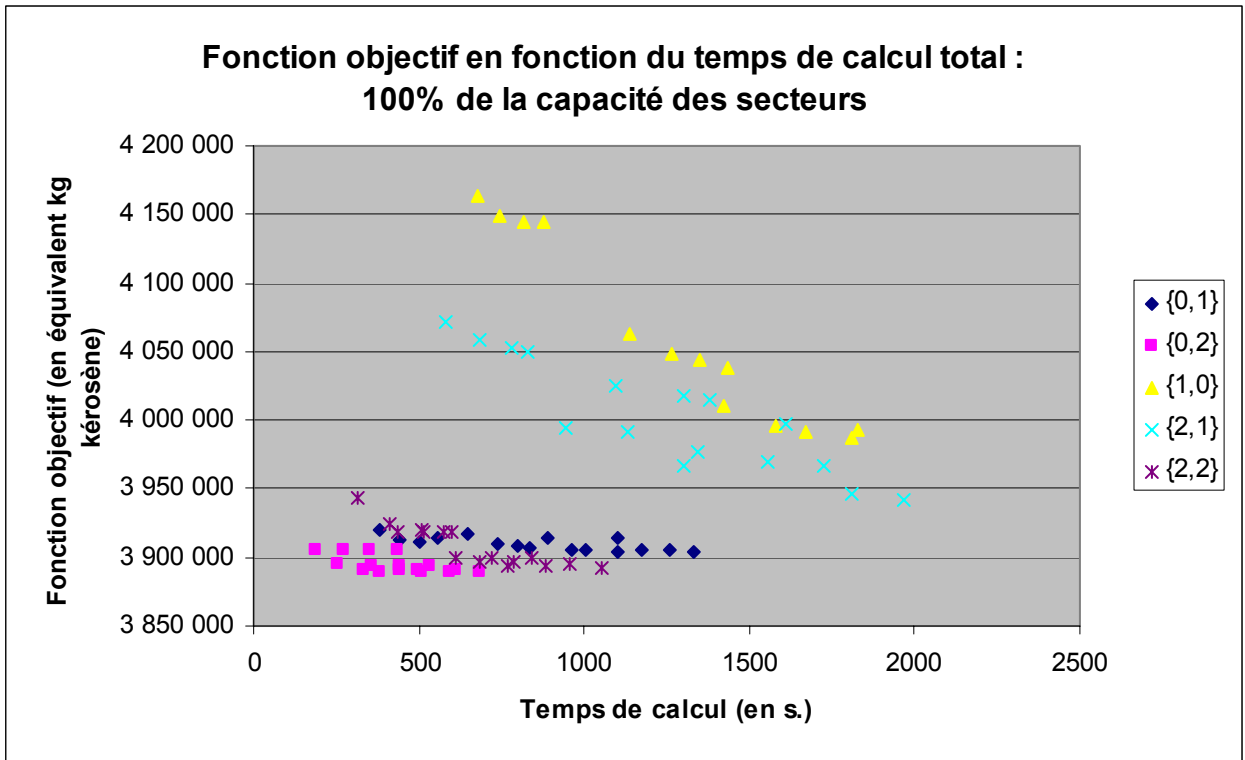


Figure 35) Valeur de la fonction objectif en fonction du temps de calcul total (capacité à 100%). Chaque point d'une même série partage les paramètres d'exploration et diffère des autres points de sa série par les paramètres d'arrêt utilisés.

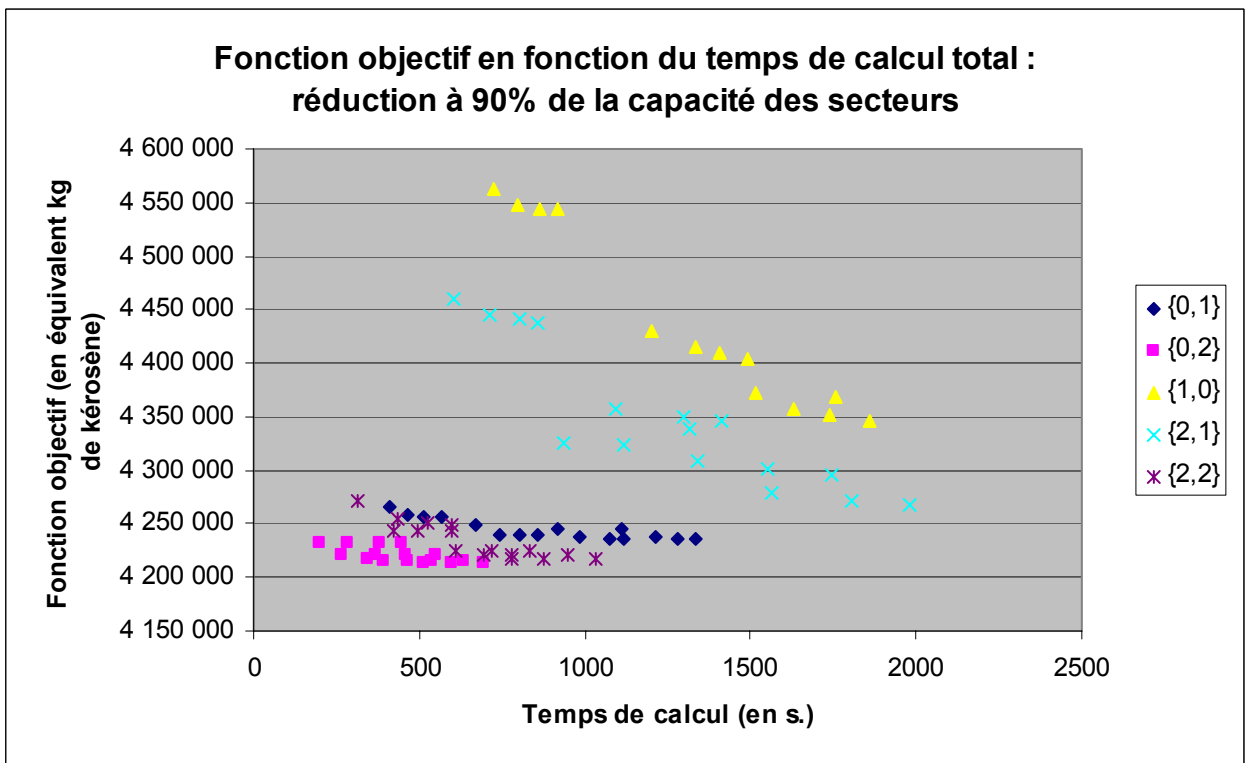


Figure 36) Valeur de la fonction objectif en fonction du temps de calcul (capacité à 90%). Chaque point d'une même série partage les paramètres d'exploration et diffère des autres points de sa série par les paramètres d'arrêt utilisés.

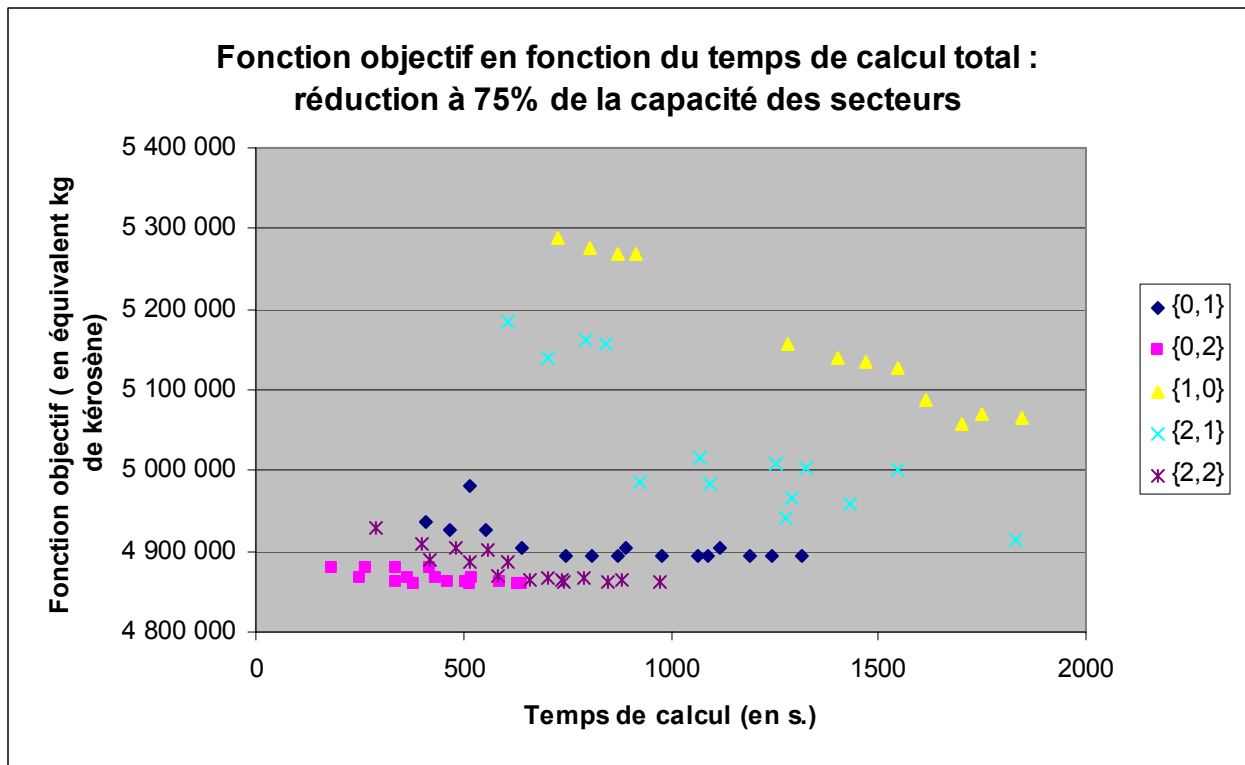


Figure 37) Valeur de la fonction objectif en fonction du temps de calcul (capacité à 75%). Chaque point d'une même série partage les paramètres d'exploration et diffère des autres points de sa série par les paramètres d'arrêt utilisés.

### iii) Détermination des paramètres d'arrêt pour la première boucle

On a représenté sur un graphique (Figure 38) l'évolution de la valeur de la fonction objectif en fonction du temps de calcul pour la combinaison de paramètres  $\{0,2\}$  et pour l'instance de test correspondant à 90% de la capacité (les résultats sont similaires pour les instances 75% et 100%). Sur ce graphe chaque courbe correspond au temps de calcul total et à la valeur de la fonction objectif obtenus en prenant comme limite de trajectoires le nombre indiqué dans la légende (10, 30, 50 ou 70) puis en faisant varier la limite temps de calcul par vol dans l'ensemble (10 s., 30 s., 50 s. et 70 s.). Chaque courbe possède donc 4 points, un point par valeur de limite de temps possible.

En augmentant le nombre de trajectoires recherchées par vol, on améliore la valeur de la fonction objectif. On décide de garder une limite à 50 trajectoires, meilleur compromis entre temps de calcul supplémentaire et amélioration de la fonction objectif. Pour cette limite le temps de calcul moyen sur les trois instances est de 337 secondes pour une limite de temps à 10 secondes par vol avec un gain moyen de 3 394 777 (kg de kérosène) pour la fonction objectif, 457 secondes pour une limite de temps à 30 secondes et 3 395 167 gagnés, 513 secondes pour 50 (gain de 3 395 233) et 612 pour 70 (même gain). On voit clairement que l'augmentation de la limite de temps de calcul par vol ne permet pas d'améliorer significativement la fonction objectif mais qu'elle a des conséquences notables sur le temps de calcul global. On décide de fixer la limite de temps de calcul par vol à 20 secondes pour la première phase.

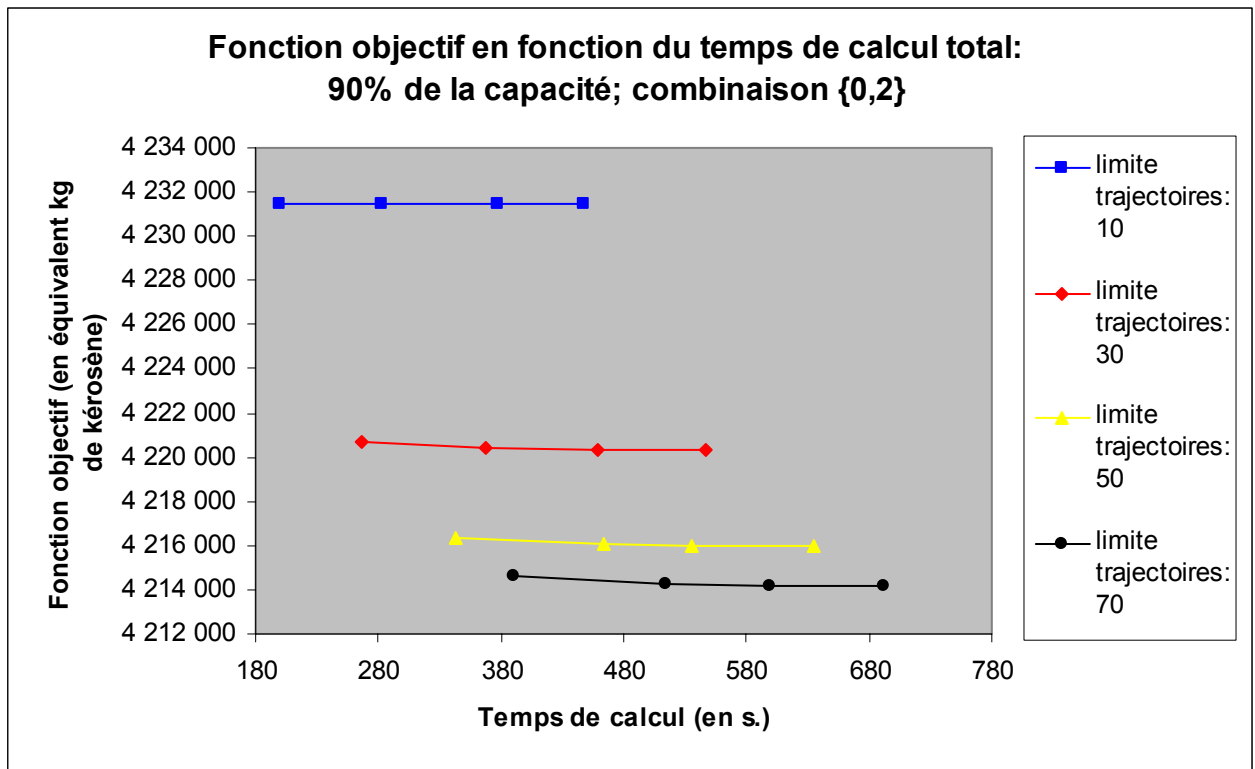


Figure 38) Valeur de la fonction objectif en fonction du temps de calcul total pour la combinaison de paramètres d'exploration {0,2} et différentes combinaisons de paramètres d'arrêt.

iv) *Récapitulatif des choix effectués pour la première boucle de génération de colonnes*

En résumé, pour la première boucle de génération de colonnes on effectue la recherche de trajectoires alternatives en choisissant comme feuille à chaque étape de l'algorithme de résolution du sous problème de tarification celle minimisant la distance à l'aéroport de destination (en suivant les routes aériennes) en cherchant au minimum 50 trajectoires par vol et en arrêtant le calcul pour chaque vol après 20 secondes de temps de calcul.

II.2.d) Résultats de l'algorithme de résolution du sous problème de tarification pour la deuxième boucle de la méthode de génération de colonnes

La démarche suivie pour déterminer les paramètres d'exploration de l'algorithme de résolution du sous problème de tarification pour la deuxième boucle est la même que celle utilisée pour la première boucle : les résultats seront moins détaillés pour ne pas alourdir inutilement ce document.

i) *Détermination des paramètres d'exploration : deuxième boucle et première évaluation des combinaisons de paramètres*

On a retenu les cinq meilleures combinaisons de paramètres d'exploration en prenant en compte les indicateurs principaux : amélioration absolue de la fonction objectif et rapport entre l'amélioration et le temps de calcul. Ces cinq combinaisons qui sont dans l'ordre {2,2}, {0,1}, {0,2}, {1,0} et {2,5} ont alors été étudiées plus en détail en faisant varier les critères d'arrêt entre 10 et 70 de la même manière que pour la première boucle.

ii) *Détermination des paramètres d'exploration : deuxième boucle et seconde évaluation des combinaisons de paramètres*

Les résultats obtenus ne permettent pas de trancher entre les deux meilleures combinaisons,  $\{0,1\}$  et  $\{2,2\}$  comme le montre la Figure 39 (obtenue dans le cas de 75% de la capacité, les résultats de 90 et 100% allant dans le même sens). On décide donc de faire les tests pour la boucle suivante en envisageant les deux combinaisons mais avec les mêmes critères d'arrêt.

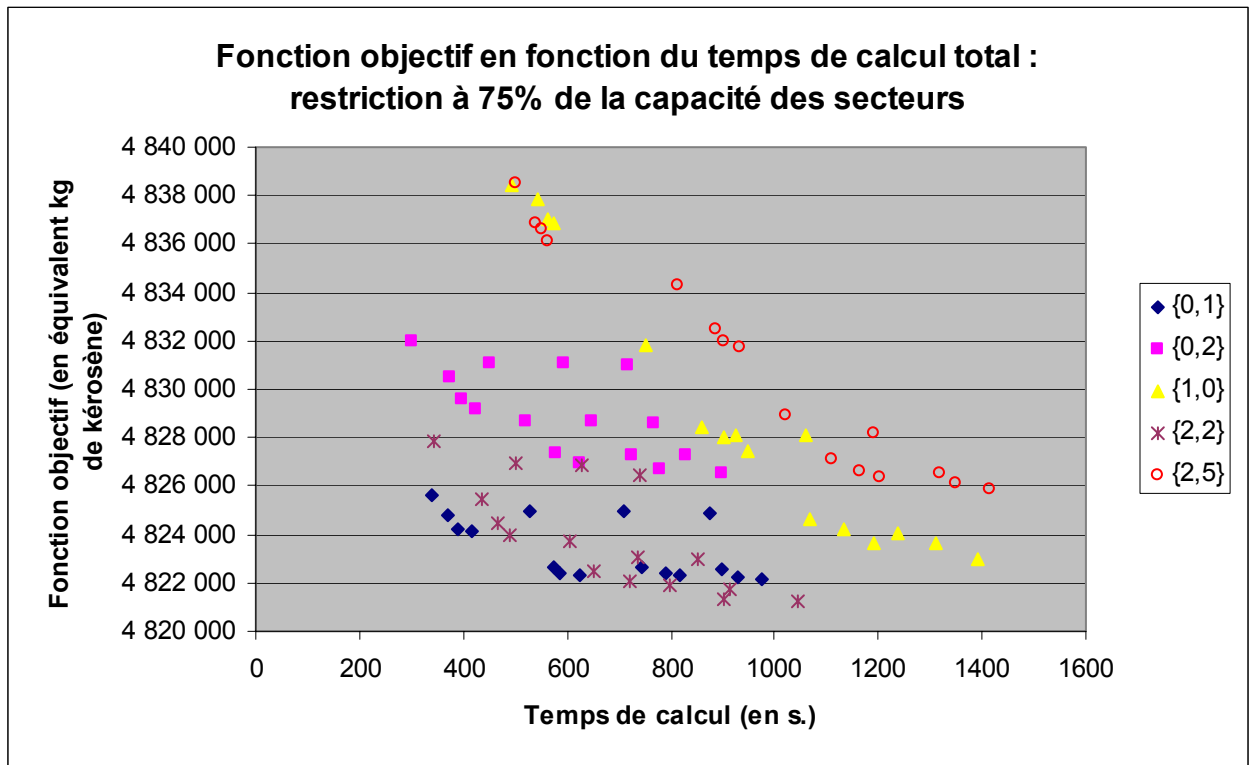


Figure 39 Valeur de la fonction objectif en fonction du temps de calcul (capacité à 75%)

iii) *Détermination des paramètres d'arrêt pour la deuxième boucle*

De la même manière que pour la première boucle, on étudie plus en détail les conséquences des paramètres d'arrêt sur les performances des deux combinaisons retenues. Les résultats obtenus (dont on peut voir un exemple dans la Figure 40) nous conduisent à choisir un temps d'arrêt de 30 secondes pour un nombre de trajectoires de 50.

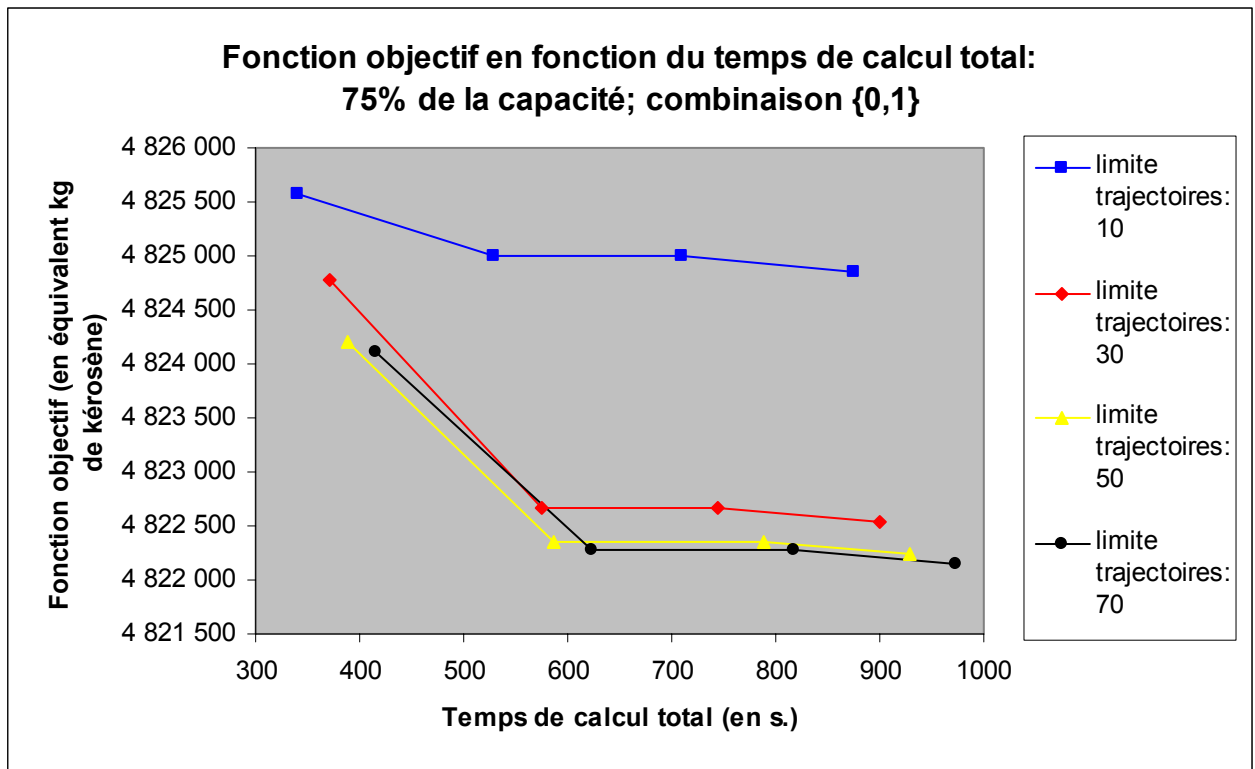


Figure 40) Valeur de la fonction objectif en fonction du temps de calcul pour la combinaison de paramètres d'exploration {0,1} et différentes combinaisons de paramètres d'arrêt.

iv) *Récapitulatif des choix effectués pour la deuxième boucle de génération de colonnes*

En résumé, pour la deuxième boucle de génération de colonnes on ne tranche pas tout de suite pour les paramètres d'exploration en conservant deux combinaisons {0,1} et {2,2}, mais on étudie les conséquences sur la boucle suivante. {0,1} correspond au choix de la feuille contenant la trajectoire partielle minimisant l'estimation de consommation de carburant. {2,2} correspond au choix de la feuille selon la règle d'Esopo-Pape (les feuilles encore jamais sélectionnées sont placées dans une pile et les autres dans une seconde pile), en prenant de plus en compte la distance à la destination pour choisir de mettre les nouvelles feuilles à examiner en début ou en fin de pile. Le calcul s'arrête lorsque l'algorithme a déterminé au moins 50 nouvelles trajectoires par vol ou si le temps de calcul dépasse 30 secondes.

II.2.e) Remarque sur l'étude paramétrique pour les boucles suivantes de génération de colonnes

On a exposé en détail dans les paragraphes précédents la méthode suivie et les résultats obtenus pour la détermination des paramètres d'exploration et d'arrêt de l'algorithme de résolution du sous problème de tarification pour la première boucle de la méthode de génération de colonnes. Les résultats ont été plus rapidement donnés pour le cas de la deuxième boucle. La méthodologie suivie pour la détermination des paramètres aux boucles suivantes a été la même que pour les premières boucles, aussi afin de ne pas lasser le lecteur, le détail des tests supplémentaires effectués dans ce cadre ne sera pas décrit plus en avant. Chaque phase de test a nécessité de lancer l'algorithme plus de 264 fois pour un temps de calcul global de plus de 60 heures et a généré de nombreuses données qu'il serait fastidieux de présenter ici. Après analyse, la stratégie finalement retenue est exposée dans le paragraphe suivant.



## II.3. Conclusions et stratégie adoptée pour la résolution du sous problème de tarification

### II.3.a) Stratégie suivie

#### i) *Première boucle*

Lors du premier appel à l'algorithme de résolution du sous problème, on va choisir comme feuille d'espace à explorer la feuille la plus proche de la destination parmi celles déjà atteintes par une trajectoire partielle (stratégie  $\{0,2\}$ ). On cherche pour chaque vol au minimum 50 trajectoires et on arrête l'exploration après 20 secondes de calcul par vol, l'expérience ayant montré que la grande majorité des trajectoires trouvables avec cette stratégie d'exploration lors de cette phase de la méthode sont déjà obtenues après 20 secondes de calcul.

#### ii) *Deuxième boucle*

Lors de la seconde phase la stratégie d'exploration est différente : lorsque l'on atteint une feuille lors de l'extension de trajectoires partielles, on regarde si on a déjà construit des trajectoires partielles traversant cette feuille. Si ce n'est pas le cas on ajoute cette feuille dans une première pile (1) et dans le cas contraire dans une seconde pile (2). A l'intérieur de la pile à laquelle on veut ajouter notre feuille, on examine la feuille située en tête : si elle est plus proche de l'aéroport de destination que la nouvelle feuille, on place celle-ci en queue de la pile et dans le cas contraire en début de la pile. Lors du choix d'une nouvelle feuille, on prend celle située en tête de la pile 1 et si celle-ci est vide, celle qui est en tête de la pile 2 (stratégie  $\{2,2\}$ ). Cette stratégie basée sur une modification de l'algorithme d'Esopo-Pape permet une exploration plus large du réseau aérien afin de trouver des trajectoires moins directes mais offrant plus de latitude au problème d'allocation des trajectoires. Cette stratégie s'est révélée étonnamment performante dans les tests effectués. On arrête l'exploration lorsque 50 trajectoires ont été construites ou après 30 secondes de temps de calcul, cette stratégie nécessitant effectivement plus de temps de calcul pour être pleinement efficace.

#### iii) *Troisième boucle*

Pour la phase suivante on revient à la stratégie utilisée pour la première phase (stratégie  $\{0,2\}$ ) en choisissant la feuille la plus proche de l'aéroport de destination et en recherchant au moins 50 trajectoires en moins de 50 secondes par vol.

#### iv) *Quatrième boucle*

Pour la quatrième phase qui est la dernière dans les tests effectués, on utilise la stratégie basée sur l'algorithme A\* : on choisit la feuille associée à la trajectoire partielle maximisant la réserve de carburant résiduelle à laquelle on soustrait la quantité minimale de carburant nécessaire pour rallier l'aéroport de destination (stratégie  $\{0,1\}$ ). Cette dernière phase a surtout pour objectif d'améliorer la borne inférieure à la solution. Etant donné que moins de vols sont concernés, on peut s'autoriser plus de temps de calcul par vol. On cherche 50 trajectoires par vol et on autorise 200 secondes de temps de calcul par vol.

### II.3.b) Conclusions sur la résolution du sous problème de tarification

Le sous problème de tarification est le cœur de la méthode de génération de colonnes : de son efficacité va dépendre la réalisabilité de la résolution par cette méthode. Les tests effectués sur le programme implémentant l'algorithme de résolution du sous problème de tarification ont constitué une grosse partie de la phase globale d'expérimentation. Les enseignements que l'on peut tirer de ces tests sont les suivants :

#### *i) Fiabilité de l'implémentation informatique*

Le programme écrit pour mettre en œuvre cet algorithme de résolution représente plus de 8000 lignes de code et a été écrit sur une période d'environ deux ans. Le lourd processus de tests et d'études paramétriques de l'algorithme a permis de mettre en évidence la fiabilité de l'implémentation informatique, avec cependant une grande consommation de mémoire vive dans le cas de certains paramètres.

#### *ii) Respect du cahier des charges du sous problème de tarification*

L'algorithme développé répond au cahier des charges du sous problème de tarification énoncé dans le chapitre IV paragraphe I.1. Il fournit un ensemble de trajectoires réalisables et vérifiant les contraintes de performance et de navigation aérienne en prenant en compte toute la complexité du système aérien (routes aériennes, niveau, ouverture/fermeture de certains secteurs). La modélisation originale développée à l'occasion de ce travail permet une description fine des trajectoires et notamment de la consommation associée et des heures d'arrivée dans les secteurs ou à l'aéroport d'arrivée. Cette description fine est indispensable pour effectuer un travail d'optimisation qui ait un sens.

#### *iii) Temps de calcul compatible avec le contexte de régulation dynamique*

Le temps nécessaire au calcul de trajectoires pour l'ensemble des vols commandables varie entre 190 et 2000 secondes en fonction des paramètres et du contexte d'appel de l'algorithme. Lors de la première phase de la génération de colonnes, l'algorithme peut trouver jusqu'à 33 000 trajectoires pour environ 850 vols, avec un temps de calcul moyen par trajectoire variant de 1 à 15 centièmes de secondes. Pour les phases suivantes, le temps de calcul global reste stable, mais on trouve moins de trajectoires et moins de vols sont concernés, ce qui est logique puisqu'on a déjà calculé un grand nombre de trajectoires durant les phases précédentes. Le temps de calcul par trajectoire augmente alors et varie entre 2 dixièmes de seconde et une seconde. Ces temps de calcul obtenus sur un PC de bureau (cadencé à 3,20 GHz) diminueraient en utilisant un ordinateur plus puissant. De plus, ils comprennent des opérations de gestion de la mémoire liées au fait qu'il n'est pas possible sur un PC de bureau d'effectuer 900 fois l'algorithme de résolution du sous problème de tarification sans libérer à chaque fois la mémoire utilisée au cours d'une recherche. Ainsi, on pourrait facilement diminuer ces temps globaux de calcul et leur ordre de grandeur général est tout à fait compatible avec le contexte de la régulation dynamique.

#### *iv) Efficacité de la programmation dynamique adaptée au problème*

La prise en compte de la structure particulière du coût réduit lors de la recherche de la politique de vitesse optimale pour une trajectoire 3D donnée permet une résolution très efficace du problème de programmation dynamique posé. La résolution de l'équation de

Hamilton-Jacobi-Bellman a permis de traiter le programme dynamique sans avoir besoin de discrétiser pas à pas la dimension temps, avec pour conséquence une résolution très rapide du problème. Sur tous les tests effectués, le temps de calcul consacré à la programmation dynamique lors de la recherche de trajectoires pour tous les vols n'a jamais dépassé 5 secondes alors que la première version de résolution à base de discrétisation du temps représentait 80% du temps de calcul.

v) *Souplesse de l'algorithme*

La présence de paramètres déterminant les règles d'exploration et d'arrêts de l'algorithme lui confère une grande souplesse d'utilisation, notamment pour rechercher des trajectoires dans des contextes différents : charge de l'espace aérien, réserve de carburant et consommation par exemple.

vi) *Importance et limites de l'étude paramétrique*

Cette souplesse apportée par les paramètres a cependant un revers : tout d'abord la détermination de ces paramètres peut se révéler lourde alors qu'elle est indispensable : les différences de temps de calcul et de performances entre les différents paramètres sont tout à fait significatives. La présence de ces paramètres nécessite donc un lourd travail préliminaire à l'utilisation de la méthode de génération de colonnes. Ce travail de détermination qui a été effectué lors de cette étude présente d'ailleurs quelques limites. Il n'est pas nécessaire de s'attarder sur le caractère particulier du paramètre d'arrêt après un temps de calcul donné, qui dépend évidemment des moyens de calcul mis en œuvre, l'allure des courbes étudiant ce paramètre ne changeant d'ailleurs pas pour autant. L'influence de la charge du système aérien a été étudiée au travers des 3 scénarios correspondant à une capacité normale (100%), réduite (90%) et fortement réduite (75%). Il est intéressant de noter que si cette charge influence les stratégies d'exploration et d'arrêt, certaines stratégies étant plus efficaces dans un contexte très saturé et le temps supplémentaire de calcul alloué à l'algorithme étant également plus efficace dans un tel contexte, il est cependant toujours possible de trouver une stratégie globalement efficace dans les trois scénarios. Les tests permettant d'étudier les conséquences de la variation de la réserve de carburant sur le comportement de l'algorithme n'ont par contre pas été réalisés et il serait intéressant de les mettre en œuvre. Une trop grande variabilité de l'algorithme face à la variation de certains paramètres (réserve, consommation de carburant, structure de la sectorisation, charge des secteurs) pourrait constituer une limite à cette méthode. Les tests ont montré que ce n'était pas le cas pour la charge, il faudrait le vérifier pour les autres paramètres.

Le bilan global de l'algorithme proposé et développé au cours de ce travail est néanmoins tout à fait positif. Il reste d'une manière plus générale à étudier le processus global de résolution qui est traité dans le paragraphe suivant.

### **III. Résolution globale du problème maître**

#### **III.1. Boucle de génération de colonnes**

##### **III.1.a) Définition de l'instance de tests principale**

On a utilisé une instance principale du problème d'optimisation pour effectuer la plus grande partie des tests. C'est évidemment la même que pour les tests sur le sous problème de

tarification puisque leur résolution est liée. L'instance représente donc le système aérien géré par la CFMU le jeudi 10 septembre 2003 à midi.

Les paramètres de modélisation retenus sont les suivants :

- Capacité

On utilise des contraintes de capacité sur des périodes fixes d'une durée d'une heure en utilisant les capacités fournies par la CFMU. Pour assurer un lissage de la charge, on utilise des contraintes de capacité supplémentaires sur des sous périodes d'une durée de 15 minutes. La capacité d'une sous période est celle de la période d'une heure divisée par 4 et multipliée par un facteur de 1,5.

- Correspondances

Comme expliqué précédemment, on ne dispose pas de données réelles sur les correspondances. Pour un vol *f* arrivant à un aéroport, on a considéré comme correspondance tout vol partant du même aéroport et dont le départ est planifié pendant une période de deux heures après la date d'arrivée prévue du vol *f* (plus le turn-over). Le turn-over qui est le temps minimum nécessaire entre l'arrivée d'un vol et le départ d'une de ses correspondances est de 5 minutes pour tous les couples vol-correspondance.

- Horizons de temps

L'horizon de protection pendant lequel on protège les secteurs des surcharges est de 3 heures. L'horizon de prédiction pendant lequel on prend en compte les conséquences des actions choisies court jusqu'à la date d'arrivée du dernier vol commandable.

- Trafic

30 203 plans de vols ont été déposés à la CFMU pour la journée du 10 septembre. Sur ceux-ci 2305 sont en l'air à midi. On peut agir sur 1112 d'entre eux. La différence vient principalement des vols longue distance qui viennent d'une zone hors CFMU ou qui y partent. Pour évaluer l'état du trafic à un instant donné, on effectue une simulation du trafic aérien à partir des plans de vol déposés (et non régulés).

- Réseau aérien

Le réseau aérien considéré représente 7837 balises pour 911 aéroports utilisés par les vols considérés. Le calcul de distances minimales entre les balises et les aéroports importants génère donc une matrice (balises,aéroports) de 7 139 507 éléments (7837\*911). Le réseau aérien est modélisé par 21 520 feuilles d'espace. Les contraintes de capacité sont assurées sur 1034 secteurs.

- Coûts des retards et de la consommation

Le coût utilisé pour les retards a été tiré du rapport de l'université de Westminster consacré à ce sujet ([6]). On a ainsi compté un coût de 0.30 Euros par passager et par minute. Pour effectuer le lien avec la consommation, on a utilisé le prix du gallon de kérosène à Rotterdam le 23/01/2006, soit 1.80 \$ converti en Euros avec un taux de un euro pour 1.24234 \$. Enfin un gallon représente 3.7854 litres et un litre 0.80 kg de kérosène. La consommation fournie en kg de kérosène a été calculée à partir des données BADA<sup>13</sup>.

---

<sup>13</sup> BADA : Base of Aircraft Data, Base de données de performances avion éditée par Eurocontrol (voir aussi p. 34)

- Initialisation

La base de trajectoires utilisée pour la première résolution du problème maître restreint comprend une trajectoire artificielle par vol, la trajectoire initiale et un ensemble de trajectoires déterminées à partir de la trajectoire initiale. La trajectoire initiale est déterminée par une simulation effectuée à partir des données des plans de vol déposés. On détermine ainsi une consommation, une date d'arrivée et une liste de secteurs-périodes traversés. La trajectoire artificielle a une consommation de deux fois celle de la trajectoire initiale et arrive avec 3 heures de retard sur l'heure d'arrivée prévue. On construit enfin des trajectoires à partir de la trajectoire initiale en autorisant un certain nombre de boucles d'attente en l'air, le nombre maximal de boucles étant déterminé par la réserve de carburant.

La méthode a ensuite été appliquée avec les mêmes paramètres algorithmiques à la même date mais à 7h00 et à 18h00. Enfin, pour chaque instance, des tests ont été effectués pour une réduction uniforme de la capacité de 10% (il reste 90% de la capacité initiale) et de 25% (75% restante).

### III.1.b) Résolution du programme linéaire relaxé

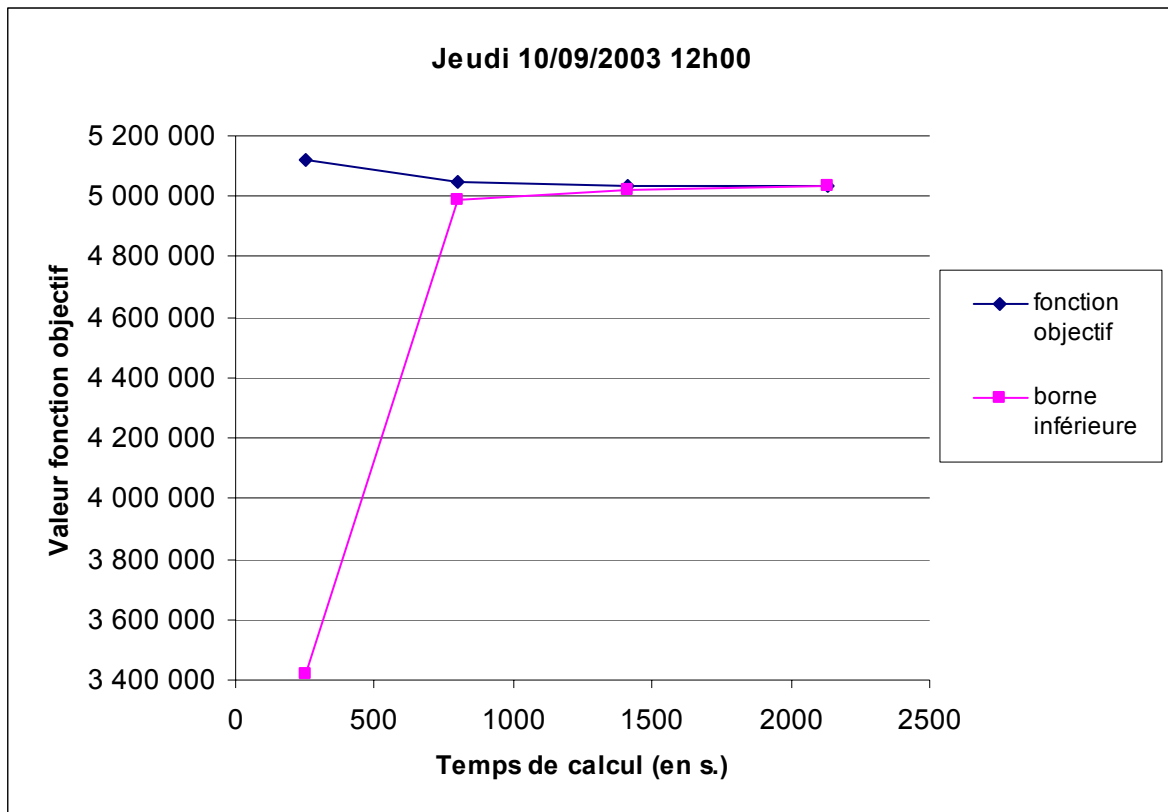
#### i) Résultats généraux de la résolution

Le nombre de boucles nécessaires pour résoudre le problème a été déterminé empiriquement en fixant un seuil sur l'écart recherché : 0.50%. Dans tous les cas, ce seuil est atteint en 4 boucles. Afin de mieux observer le comportement de l'algorithme, on a effectué toujours effectué 4 boucles même si le seuil était atteint plus rapidement.

4 boucles correspondent à 5 résolutions du problème maître restreint et 4 calculs de nouvelles trajectoires pour enrichir notre ensemble de trajectoires. Le calcul des nouvelles trajectoires a été effectué à partir des paramètres exposés dans le paragraphe II.3.a) de ce chapitre. Les résultats sont présentés dans le Tableau 6 et dans la Figure 41.

Paramètres d'exploration et d'arrêt	Boucle	Temps de calcul (secondes)	Fonction objectif	Borne inférieure	Ecart	Nombre de trajectoires nouvelles
	Boucle 0	<1	13 472 900			
{0,2} 50,20	Boucle 1	252	5 121 020	3 417 623	33.26 %	21 496
{2,2} 50,30	Boucle 2	798	5 045 200	4 986 732	1.16 %	3 365
{0,2} 50,50	Boucle 3	1 415	5 037 360	5 021 032	0.32 %	817
{0,1} 50,200	Boucle 4	2 131	5 034 360	5 032 246	0.04 %	297

**Tableau 6) Résultats de la résolution pour l'instance de tests principale. Les paramètres d'arrêt sont donnés dans l'ordre trajectoires, temps de calcul**



*Figure 41) Evolution de la fonction objectif et de la borne inférieure au cours des différentes phases de la génération de colonnes, instance de midi*

La méthode a été appliquée pour deux instances différentes la même journée à 07h00 et 18h00. Les résultats pour 07h00 sont présentés dans la Figure 42. Les résultats pour 18h00 sont similaires. L'écart entre la fonction objectif et la borne inférieure est de 0.07 % après 30 minutes de calcul pour l'instance 07h00 et de 0.01 % après 25 minutes de temps de calcul pour l'instance 18h00. L'écart est de 0.04% après 35 minutes de temps de calcul dans le cas de l'instance principale.

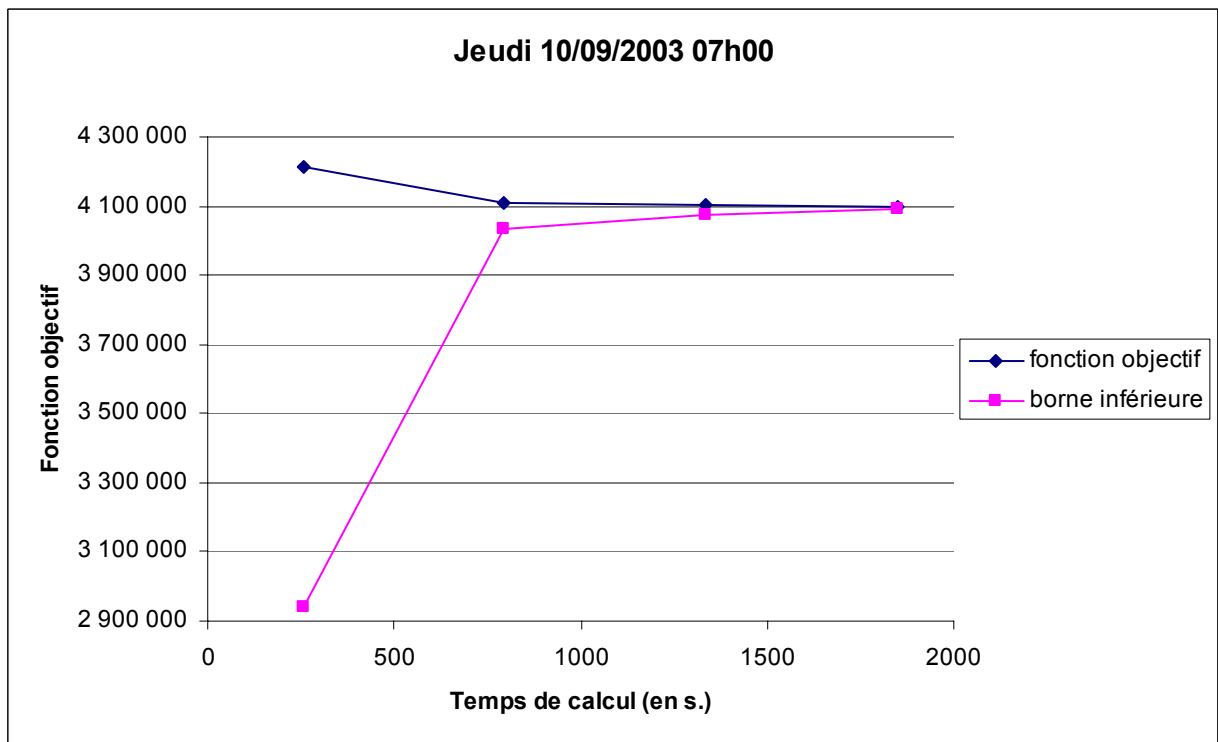


Figure 42) Evolution de la fonction objectif et de la borne inférieure au cours des différentes phases de la génération de colonnes, instance de 07h00

ii) Remarques générales sur les résultats

Sur nos 3 instances de tests, après 4 boucles de génération de colonnes, l'écart à la borne inférieure est compris entre 0.01 % et 0.07% de la fonction objectif. Le temps de calcul total est compris entre 25 et 35 minutes. On peut également remarquer qu'après 3 boucles, on n'est dans le pire des cas qu'à 0.68 % de la borne inférieure pour un temps de calcul inférieur de 10 minutes en moyenne. On a donc une convergence assez rapide de l'algorithme et surtout une estimation assez fine de la qualité de la solution réalisable obtenue. C'est un argument fort en faveur de l'algorithme dans le contexte de l'application dans le cadre de la régulation court terme lorsque la contrainte de temps de calcul est importante.

iii) Bornes inférieures

Les bornes inférieures présentées dans les figures et les tableaux de résultats ont été calculées à l'aide de la formule :  $\bar{z} + \sum_{f \in F} \bar{c}_f^*$  (borne 1). Une deuxième borne (2) avait été proposée à partir d'une proposition de Farley ([42]). Afin de pouvoir l'utiliser avec les données disponibles dans notre cas, l'expression de cette borne a du être simplifiée, perdant ainsi de sa pertinence (voir p.73 chapitre III paragraphe V.2.b). Les résultats expérimentaux montrent que la deuxième borne proposée ne présente plus aucun intérêt : la borne obtenue ainsi est systématiquement moins bonne que celle fournie par la borne 1 alors que l'effort de calcul est supérieur. D'autre part, du fait de son expression simplifiée cette borne ne s'améliore quasiment pas lors des différentes boucles de génération de colonnes. En conclusion, dans son expression actuelle cette borne est inutile. Pour pouvoir utiliser une borne basée sur celle de Farley, il faudrait faire au cours de l'algorithme des calculs supplémentaires pour obtenir toutes les données nécessaires à son utilisation sans

simplifications. Le gain attendu n'en vaut peut être pas la peine étant donné que la première borne donne des résultats très satisfaisants et s'améliore grandement au fur et à mesure des itérations de la génération de colonnes.

#### *iv) Perspectives sur les temps de calcul*

Les temps de calcul globaux obtenus pour résoudre le problème relaxé ont un ordre de grandeur tout à fait satisfaisant. En effet trouver une solution en 30 minutes sur un PC de bureau rend tout à fait envisageable la même résolution en 5 minutes dans un contexte plus professionnel. Ils sont quasiment exclusivement consacrés à la recherche de trajectoires alternatives, la résolution des problèmes maîtres restreints successifs par CPLEX étant quasiment instantanée.

En ce qui concerne la résolution du sous problème de tarification, on peut par ailleurs mettre en lumière quelques pistes d'amélioration. Tout d'abord certaines données sont calculées au cours de la recherche de trajectoires alternatives afin de pouvoir calculer la deuxième borne inférieure (dérivée de celle de Farley). On vient de voir que cette borne n'apportait aucune information supplémentaire, on pourrait donc supprimer ces calculs et accélérer le processus.

Par ailleurs en étudiant plus en détail la répartition du temps de calcul, on s'aperçoit qu'un petit nombre de vols « difficiles » représentent une part non négligeable du temps de calcul total. Ainsi deux vols nécessitent 27 % du temps de calcul complet pour l'instance principale (12h00), 15% pour un vol dans l'instance 2 (07h00) et 30% pour deux vols dans l'instance 3 (18h00). Une stratégie envisageable pour faire diminuer les temps de calculs globaux serait alors d'individualiser les paramètres utilisés dans l'algorithme de résolution du sous problème de tarification. Actuellement les mêmes paramètres sont utilisés pour tous les vols et ne changent qu'en fonction de la boucle de génération de colonnes. On pourrait envisager de repérer dans la première ou deuxième boucle les vols difficiles (pour lesquels on utilise tout le temps de calcul autorisé) afin d'appliquer des paramètres spécifiques pour ces vols dans les phases suivantes. La détermination des paramètres à appliquer demande un travail de tests supplémentaire mais les résultats pourraient être intéressants, étant donnée le poids des vols difficiles dans le temps de calcul global.

Enfin, on peut noter que la détermination de trajectoires de coût réduit négatif est complètement indépendante pour chaque vol. Les calculs pourraient alors être parallélisés, c'est-à-dire effectués sur plusieurs processeurs en même temps, diminuant ainsi le temps de calcul global.

#### III.1.c) Résolution du programme linéaire relaxé avec des capacités réduites

Pour les trois instances de tests (07h00, 12h00 et 18h00), la même méthode de résolution a été appliquée en diminuant artificiellement la capacité afin d'étudier les conséquences sur le processus de résolution. D'une manière générale, l'évolution de la fonction objectif et de la fonction représentant la borne inférieure est la même que dans le cas où l'on utilise la capacité initiale. La qualité finale de la solution n'est pas affectée par la diminution de capacité puisqu'à l'issue du processus de résolution, l'écart maximum entre la fonction objectif et la borne inférieure ne représente que 0.39 % de la fonction objectif. En revanche on peut noter une convergence un peu moins rapide comme on peut le voir sur la Figure 43. De la même manière, le temps de calcul global augmente mais n'explose pas : 40 minutes au maximum pour l'instance à 07h00 du matin et pour une capacité réduite à 75% de la capacité initiale. La résolution du problème maître restreint par CPLEX n'est pas affectée même si le nombre de



contraintes et de variables augmente. Il faut toujours moins de une seconde pour résoudre le RMP.

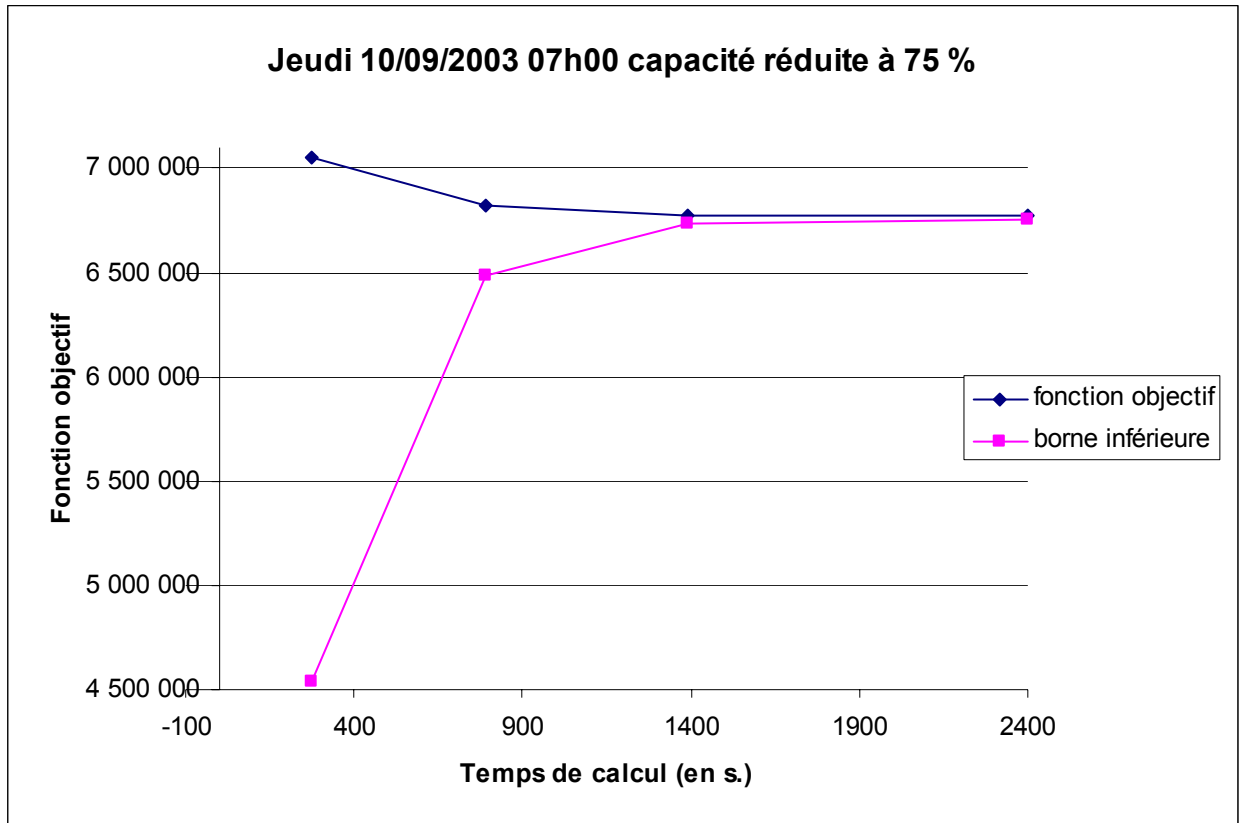


Figure 43) Evolution de la fonction objectif et de la borne inférieure au cours des différentes phases de la génération de colonnes, instance de 07h00 capacité réduite à 75% de la capacité initiale

### III.1.d) Résolution du programme linéaire en nombres entiers

#### i) Caractère peu fractionnaire de toutes les solutions obtenues

Une attention conséquente avait été portée au cours de l'élaboration du processus global de résolution à la phase permettant d'obtenir une solution entière par le processus de branch-and-price. Or les solutions réalisables du problème relaxé obtenues à l'issue de la première phase de résolution ont toutes un caractère peu fractionnaire, c'est-à-dire que peu de variables de décision prennent une valeur non entière. Ce constat est valable pour toutes les instances de tests (9 différentes) et également à chaque résolution du RMP lors des différentes boucles de génération de colonnes. Ceci nous a conduit à tenter de résoudre le dernier problème maître restreint directement en nombres entiers à l'aide de CPLEX, sans passer par des heuristiques d'arrondi.

#### ii) Résolution directe en nombres entiers par CPLEX

Après la dernière boucle de génération de colonnes, on résout directement le dernier problème maître restreint obtenu en ajoutant les contraintes d'intégrité pour les variables binaires de décision. Toutes les trajectoires alternatives calculées au cours des phases précédentes ont déjà été ajoutées au problème maître restreint. Tout d'abord, la résolution en nombres entiers par CPLEX ne pose pas de problèmes de temps de calcul, cette résolution étant très rapide du fait du caractère peu fractionnaire de la solution du RMP relaxé. Par ailleurs, et c'est le plus intéressant, la valeur de la solution entière obtenue n'est guère plus

élevée que la solution du problème relaxé. Comme la borne inférieure calculée pour le programme relaxé est toujours valable pour le problème en nombres entiers, on peut facilement évaluer la qualité des solutions entières obtenues. Ainsi dans le cas de la capacité initiale, on obtient un écart de respectivement 0.46%, 0.05% et 0.28% pour les instances de 07h00, 12h00 et 18h00. Pour les instances pour lesquelles on a réduit la capacité, les résultats sont un peu moins bons mais restent tout à fait acceptables puisque dans le pire des cas on obtient un écart de 1.18%. Devant ces résultats, nous avons décidé de ne pas pousser plus loin les tests concernant le processus d'arrondi de la solution.

Nous nous sommes aperçus en revanche que les solutions optimales comportaient toujours un certain nombre de trajectoires artificielles. Ces trajectoires artificielles sont ajoutées au début du processus afin de garantir l'existence d'une solution réalisable pour le RMP. C'est la technique dite du « grand M ». Nous avons donc décidé d'étudier plus en détail le rôle de ces trajectoires artificielles et les techniques possibles de gestion de ces trajectoires, de manière à allouer une trajectoire réelle à chaque vol à la fin du processus. C'est l'objet du prochain paragraphe consacré à la gestion des trajectoires artificielles.

### III.2. Gestion des trajectoires artificielles

Dans la solution entière obtenue au terme du processus de résolution de l'instance principale, on alloue à 48 vols (sur 1112) une trajectoire artificielle. Ce sont respectivement 51 vols (sur 1083) et 42 vols (sur 954) qui sont dans le même cas pour les instances de 07h00 et de 18h00. Deux possibilités peuvent expliquer la présence de trajectoires alternatives : soit le problème initial n'est pas réalisable et la présence de trajectoires artificielles dans une solution est obligatoire, soit le coût choisi pour les trajectoires artificielles est trop faible et il existe alors des solutions réalisables avec un nombre moins important de trajectoires artificielles. Des tests supplémentaires sont nécessaires pour déterminer si la présence de trajectoires artificielles est obligatoire ou si elles peuvent être éliminées en augmentant leur coût.

#### III.2.a) Tests additionnels concernant les trajectoires artificielles

##### *i) Démarche des tests additionnels*

Le coût des trajectoires artificielles est calculé initialement à partir de celui des trajectoires initiales. La trajectoire artificielle du vol  $f$  a ainsi une consommation égale à deux fois la consommation de la trajectoire initiale du vol  $f$ . Elle arrive avec trois heures de retard sur l'horaire prévu d'arrivée mais elle ne consomme aucune capacité de contrôle. On va modifier les caractéristiques des trajectoires artificielles, puis effectuer une nouvelle boucle de génération de colonnes en considérant toutes les trajectoires calculées lors de la résolution du problème maître relaxé. Afin d'analyser le rôle de la structure de coût des trajectoires artificielles sur la solution, deux possibilités sont utilisées.

- Première possibilité pour le coût des trajectoires artificielles

La consommation des trajectoires artificielles est égale à six fois celle de la trajectoire initiale. La trajectoire artificielle arrive avec un retard de trois heures.

- Deuxième possibilité pour le coût des trajectoires artificielles

Toutes les trajectoires artificielles ont une consommation de 100 000 kilos de kérosène et arrivent avec trois heures de retard.

## ii) Résultats des tests additionnels

Aucune tentative effectuée en augmentant le coût des trajectoires artificielles n'a modifié la solution obtenue en fin de processus de résolution. Pourtant le temps consacré à la recherche de nouvelles trajectoires a été conséquent, avec 250 secondes par vol. La fonction objectif a une valeur plus élevée lorsque l'on a augmenté le coût des trajectoires artificielles mais la part du coût due aux trajectoires réelles est inchangée. On a représenté sur la Figure 44 l'évolution de la part de chaque composante dans la fonction objectif. On a également représenté sur cette même figure le coût des trajectoires initiales des vols auxquels a été allouée une trajectoire artificielle afin de mesurer leur poids. La première composante de la fonction objectif représente la part des trajectoires artificielles. La seconde composante représente la part des trajectoires réelles. On peut remarquer tout d'abord la faiblesse du coût des vols auxquels on a alloué une trajectoire artificielle par rapport au coût total. Par ailleurs la part réelle est quasiment inchangée. La figure a été obtenue en allouant une consommation arbitraire de 100 000 kilos de kérosène à chaque trajectoire artificielle, mais les résultats sont identiques en utilisant une consommation proportionnelle à la consommation de la trajectoire initiale. Il n'y a alors probablement pas de solution réalisable au problème.

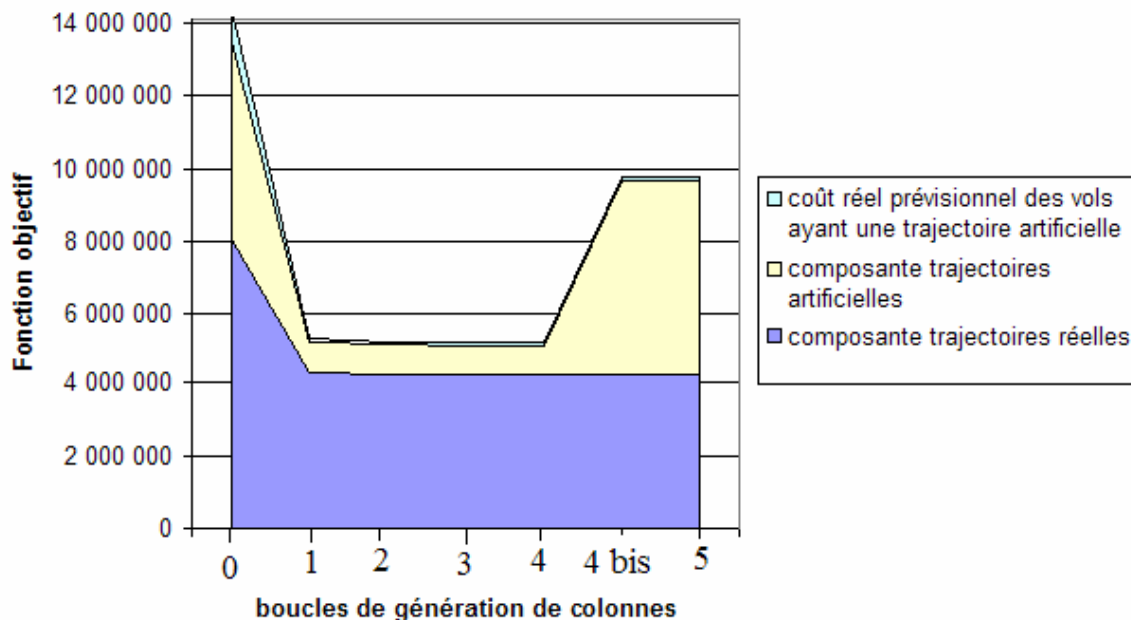


Figure 44) Evolution de la composition de la fonction objectif au cours du processus de générations de colonnes. Entre les boucles 4 et 4 bis, on a augmenté la consommation des trajectoires artificielles (à 100 000 kilos par trajectoire) puis on a recalculé des trajectoires pour la dernière boucle

### III.2.b) Analyse de la place des trajectoires artificielles dans la solution obtenue

#### i) Place des trajectoires artificielles

Si la présence de trajectoires artificielles est obligatoire, la nature des coûts de ces trajectoires joue un rôle dans la manière dont ces trajectoires artificielles sont allouées. Dans les tests effectués, on a fait varier la consommation des trajectoires artificielles mais on n'a pas modifié l'heure d'arrivée : trois heures de retard systématiques. Ceci a pour conséquence l'allocation des trajectoires artificielles à des vols n'ayant peu ou pas de correspondances. Le fait que la solution reste inchangée en augmentant sensiblement la consommation des trajectoires artificielles laisse supposer l'existence de trajectoires artificielles obligatoires pour

certaines vols. Nous allons donc maintenant nous intéresser aux causes de la persistance de la présence de trajectoires artificielles.

### *ii) Causes de la persistance de trajectoires artificielles*

Les résultats des tests additionnels montrent que le problème posé initialement n'admet pas de solution réalisable et qu'en conséquence des trajectoires artificielles sont allouées à certains vols. Plusieurs facteurs liés à l'instance de tests choisie expliquent cette situation. Certains de ces facteurs sont d'ordre structurel (liés au problème réel) et d'autres sont liés à l'implémentation expérimentale du problème.

#### ▪ Causes structurelles

La simulation de la journée de trafic nécessaire pour déterminer l'état du système aérien à un instant donné a été effectuée à partir des données de plans de vols déposés. Cela signifie que les régulations imposées par la CFMU (retards au sol notamment) ne sont pas prises en compte. Ainsi, lorsque les seules trajectoires possibles pour les vols sont les trajectoires initiales ou des variantes de celles-ci à partir de la même trajectoire 3D, la première résolution du RMP conduit à 271 trajectoires artificielles. La situation initiale choisie est donc particulièrement difficile. De plus, on n'agit ni sur les vols encore au sol du fait de la trop grande incertitude sur leur heure de décollage, ni sur les vols long-courriers partant ou arrivant d'un aéroport en dehors de la zone de responsabilité de la CFMU. Tous ces vols utilisent de la ressource de capacité mais ne peuvent être modifiés. Ainsi certains secteurs-périodes sont déjà saturés uniquement avec les vols non commandables. Certains vols utilisant obligatoirement un secteur-période déjà saturé auront alors automatiquement une trajectoire artificielle.

#### ▪ Causes liées à l'implémentation

Comme toute mise en œuvre expérimentale, notre approche est dépendante de la qualité des données utilisées. Or, si les données disponibles sont fiables et complètes, elles sont peu documentées, provoquant ainsi certaines difficultés. Ainsi certains secteurs-périodes ont des capacités officielles particulièrement faibles. De même, certains vols suivent des trajectoires initiales non compatibles avec les données sur les routes aériennes. Etant donné que nous n'agissons pas sur les vols qui ne survolent pas une feuille d'espace valide, ces incertitudes sur les données se traduisent par une liberté d'action diminuée.

Dans une application opérationnelle, le filtre tactique de la CFMU serait au moins partiellement appliqué et la situation serait ainsi plus facile à résoudre. De plus les problèmes non structurels seraient résolus et la liberté d'action ne serait plus artificiellement réduite. On peut donc faire l'hypothèse que le nombre de trajectoires artificielles résiduelles serait notablement réduit. L'objectif de la résolution du problème d'optimisation étant d'allouer une trajectoire réelle à chaque vol, il faut cependant étudier la manière de gérer les trajectoires artificielles résiduelles. C'est l'objet du paragraphe suivant.

### III.2.c) Gestion des trajectoires artificielles

Plusieurs démarches sont possibles afin d'allouer à chaque vol une trajectoire réelle à partir de la solution proposée par la résolution globale. Toutes auront pour conséquence de violer certaines contraintes de capacité puisque le problème n'admet pas de solution réalisable.

*i) Première possibilité de suppression des trajectoires artificielles*

Une première possibilité consiste à conserver la trajectoire allouée pour tous les vols pour lesquels elle est réelle. On choisit ensuite une trajectoire réelle pour les autres vols. On peut par exemple choisir la trajectoire initiale. On peut également choisir la trajectoire réelle ayant le coût réduit le plus petit dans la base de la solution retenue. Cette méthode est probablement la plus efficace pour limiter les violations de contraintes de capacité puisque le coût réduit prend en compte le coût de la traversée de secteurs-périodes saturés. Une fois une trajectoire réelle allouée à chaque vol, il faut ensuite faire une analyse des contraintes de capacité violées afin de déterminer si la solution est acceptable ou non. L'ensemble des trajectoires calculées pour chaque vol est alors un outil précieux pour affiner la solution retenue. Cette méthode permet d'identifier les vols particulièrement difficiles et les secteurs-périodes particulièrement surchargés.

*ii) Deuxième possibilité de suppression des trajectoires artificielles*

Une deuxième possibilité consiste à profiter de la rapidité de la résolution des différents problèmes maîtres restreints afin de réoptimiser le problème en relaxant petit à petit les contraintes de capacité jusqu'à ce qu'il n'y ait plus de trajectoires artificielles. On se place à la fin du processus global de résolution. On augmente par exemple d'une unité toutes les capacités des secteurs-périodes et on procède à la résolution du nouveau programme linéaire obtenu, sans changer l'ensemble des trajectoires artificielles. On continue ainsi d'augmenter les capacités des secteurs-périodes jusqu'à ce qu'il n'y ait plus de trajectoires artificielles. Cette méthode présente l'avantage et l'inconvénient d'être fondée sur un processus d'optimisation remettant en cause toutes les décisions prises jusqu'à maintenant en fonction des nouvelles données disponibles. La solution trouvée aura ainsi un coût faible mais le respect des contraintes de capacité sera en contrepartie moins bien assuré. La méthode peut être affinée en augmentant de manière plus discriminée les capacités des secteurs-périodes : on peut par exemple n'augmenter tout d'abord que les capacités sur les sous périodes avant de toucher aux capacités sur les périodes horaires.

*iii) Résultats de la suppression des trajectoires artificielles*

On a implémenté à titre d'exemple la première méthode sur l'instance principale en choisissant la trajectoire initiale pour les vols ayant une trajectoire artificielle. 38 secteurs subissent alors des dépassements de capacité. Parmi ceux-ci, 31 sont des secteurs dont la capacité résiduelle (ce qui reste lorsque l'on a enlevé l'utilisation des vols non commandables) est nulle. De plus, 50% des surcharges concernent la dernière période pendant laquelle on surveille les capacités, lorsqu'une grande partie de la charge vient des vols encore au sol sur lesquels on n'agit pas. Ceci confirme le fait que la présence de trajectoires artificielles est principalement due au manque de liberté d'action sur le système dans notre instance de test, liberté qui sera plus grande dans l'application opérationnelle de la régulation court terme du trafic aérien.

Le bilan global de l'application de la méthode de génération de colonnes au problème d'optimisation de la régulation court terme est présenté dans le prochain chapitre qui constitue la conclusion de ce document. On y détaille également les perspectives d'approfondissement de ce travail.

# chapitre VI Conclusion générale et perspectives

## I. Perspectives d'approfondissement

### I.1. Approfondissement sur la résolution d'une instance

#### I.1.a) Extensions algorithmiques possibles

##### i) *Résolution du sous problème de tarification*

La résolution du sous problème de tarification a été la phase du processus de résolution la plus étudiée au cours de ce travail. On peut cependant envisager quelques approfondissements.

- Programmation dynamique sous contrainte de carburant

Tout d'abord, il reste un point en suspens dans la résolution proposée. L'algorithme de détermination des dates d'entrée secteur minimisant le coût réduit d'une trajectoire 3D peut, dans certains cas très particuliers, « manquer » une trajectoire réalisable. En effet, cet algorithme calcule pour toutes les dates d'arrivée possibles, la trajectoire 4D qui minimise le coût réduit, mais sans prendre en compte la contrainte de carburant. Dans le cas où l'on n'a pas trouvé de trajectoire réalisable sur l'ensemble des dates d'arrivée possibles, une trajectoire, de coût réduit supérieur mais néanmoins négatif, mais avec une consommation réalisable peut alors avoir été oubliée. Ce cas de figure se produit rarement dans les instances testées mais il faut cependant vérifier l'occurrence d'un tel cas de figure sur d'autres instances. Par ailleurs, il serait très intéressant d'élaborer un processus complémentaire n'agissant que lorsque qu'aucune trajectoire satisfaisant les contraintes de carburant n'a été trouvée. Ce problème de programmation dynamique sous contrainte supplémentaire est apparenté aux recherches de plus courts chemins sous contraintes supplémentaires.

- Nouveaux paramètres d'exploration

Au cours de la phase I de l'algorithme de résolution du sous problème de tarification, on gère une liste dynamique de feuilles restant à explorer. Le choix de la prochaine feuille que l'algorithme va étudier est déterminé par ce que l'on a nommé le paramètre d'exploration. On a présenté et testé dans ce travail 13 possibilités différentes pour ce paramètre. Ces possibilités sont décrites dans le paragraphe III.4.d) du chapitre IV (p. 96). Parmi celles-ci une stratégie consiste à stocker les feuilles restant à explorer dans deux piles distinctes, puis à vider la première pile avant de prendre une feuille dans la seconde. Les feuilles qui apparaissent pour la première fois sont stockées dans la pile 1 et les autres dans la seconde. Une nouvelle possibilité consiste alors à échanger ces deux piles, c'est-à-dire traiter en priorité la pile 2 au lieu de la pile 1. Dans la version proposée par Bertsekas ([53]) on compare de plus le label d'une nouvelle feuille à stocker avec le label de la feuille en tête de la pile afin de déterminer si on place la nouvelle feuille en tête ou en queue de pile. Une autre nouvelle possibilité consiste alors à trier complètement les deux piles en fonction du label. Etant donné que l'on peut combiner le choix d'une structure de données à celui du label de la feuille, cela donnerait 6 nouveaux paramètres d'exploration à tester. Cela représente un travail conséquent sans garantie d'améliorer les résultats de l'algorithme, mais on vu lors des tests que les différences de performance entre les différentes stratégies d'exploration sont significatives. L'algorithme pourrait alors être potentiellement amélioré.

## ii) *Résolution du problème maître restreint*

Le processus itératif de l'algorithme de génération de colonnes peut lui aussi faire l'objet de tests d'approfondissement.

- Nombre de boucles de génération de colonnes

On a vu lors des tests que la convergence de l'algorithme est très rapide. Les deux premières boucles améliorent significativement la valeur de la fonction objectif et celle de la borne inférieure. Les deux dernières affinent principalement la connaissance de la borne inférieure. En résumé on consacre environ un tiers du temps de calcul global à déterminer plus précisément la borne inférieure sans amélioration de la fonction objectif. Cette connaissance de la borne inférieure est précieuse mais dans un contexte où le temps de calcul est précieux, peut être pourrait-on en économiser lors de cette phase. Il faudrait effectuer pour cela des tests beaucoup nombreux sur des instances différentes du problème d'optimisation afin de pouvoir tirer des lois générales sur la convergence de l'algorithme. On pourrait alors fixer un seuil d'arrêt pour l'écart connu en sachant que l'écart réel est probablement plus faible.

- Détermination plus fine des paramètres

Nous avons effectué durant ce travail une étude paramétrique approfondie pour les paramètres d'exploration et d'arrêt de l'algorithme de résolution du sous problème de tarification. Cependant, les conséquences du choix d'un couple de paramètres ont été généralement étudiées jusqu'à la prochaine boucle et dans certains cas celle d'après. Pour 13 paramètres d'exploration possibles et au moins 16 paramètres d'arrêt, il n'est pas possible d'étudier systématiquement les conséquences pour toutes les boucles suivantes. En effet s'il y a 208 possibilités de paramètres ( $13 \times 16$ ) pour la première boucle, étudier toutes les combinaisons possibles à la seconde boucle nécessite 43 264 tests, 8 998 912 à la troisième... Aussi la prise en compte des conséquences du choix d'une combinaison de paramètres pour des autres boucles que la suivante nécessite d'effectuer un tri sur les combinaisons considérées. Ce travail a partiellement été fait mais pourrait encore être prolongé.

- Gestion vol par vol

Une autre piste pour améliorer le temps de calcul consiste enfin à individualiser les paramètres d'exploration et d'arrêt. Comme il est sans intérêt de faire des tests vol par vol qui ne seraient pas utilisables pour d'autres instances, une possibilité consisterait à déterminer quelques grandes catégories de vols et à chercher les meilleurs paramètres pour chaque catégorie. Ces catégories pourraient être définies a priori en fonction des caractéristiques du vol (réserve de carburant, distance à l'aéroport de destination par exemple) ou en cours de résolution (vols pour lesquels aucune trajectoire n'a été trouvée ou ayant nécessité l'utilisation complète du temps de calcul alloué).

### I.1.b) L'obtention d'une solution entière

#### i) *Expérimentation du processus de branch-and-price*

Une grande attention a été consacrée au processus théorique pour obtenir une solution entière. Une règle de branchement a été proposée, les modifications nécessaires de l'algorithme de résolution du sous problème de tarification ont été identifiées et l'algorithme global de branch-and-price a été programmé. Or, du fait du caractère très peu fractionnaire des solutions relaxées obtenues, l'expérimentation pratique de la méthode d'obtention d'une solution entière n'a pas été mise en œuvre. Une extension logique de ce travail consisterait

alors à utiliser l'outil qui a été préparé pour tester le processus de branch-and-price sur les instances où la solution entière obtenue est la moins bonne. Une telle expérimentation permettrait de déterminer la structure de l'arbre de branch-and-price et d'évaluer le temps de calcul qui lui est nécessaire pour obtenir la solution optimale. Un grand nombre de trajectoires sont calculées au cours de la résolution du problème relaxé : de l'ordre de 26 000 dans l'instance de tests principale. Les tests supplémentaires effectués pour tenter de faire disparaître les trajectoires artificielles n'ont pas permis de déterminer un nombre significatif de nouvelles trajectoires malgré un temps de calcul autorisé conséquent (250 secondes par vol pour 4 nouvelles trajectoires en tout). Aussi, on peut douter du fait que les contraintes supplémentaires ajoutées lors du processus de branch-and-price auraient permis de trouver des nouvelles trajectoires changeant radicalement la solution. Une expérimentation permettrait ainsi de déterminer le nombre de trajectoires nouvelles calculées et d'en déduire si les gains obtenus valent le temps de calcul consacré au processus de branch-and-price.

#### *ii) Etude théorique et pratique du caractère fractionnaire de la solution relaxée*

Les solutions relaxées obtenues pour toutes les instances testées étaient remarquablement peu fractionnaires, ce qui a permis leur résolution en nombres entiers sans utiliser le processus de branch-and-price avec un résultat très proche de la borne inférieure. S'il est assuré que ce processus est valide dans le cas testé, on ne peut tout à fait exclure la possibilité que d'autres instances auraient une solution relaxée très fractionnaire. Deux approches complémentaires sont alors possibles pour lever cette incertitude. La première, expérimentale, consiste à multiplier les instances de tests. Les instances testées étaient très variables d'un point de vue trafic et capacité des secteurs. En revanche les routes aériennes étaient similaires et peut être faudrait-il tester des instances faisant également varier ce facteur.

La deuxième approche est plus théorique : il est possible dans certains cas de montrer que les contraintes d'un programme linéaire représentent des facettes du polyèdre défini par l'enveloppe convexe des solutions en nombres entiers. Dans ce cas, la résolution du problème relaxé sera peu fractionnaire. Une telle étude en complément des tests supplémentaires permettrait d'améliorer la compréhension des résultats obtenus. Elle permettrait également de distinguer les contraintes qui jouent un rôle important dans le caractère fractionnaire ou entier des solutions relaxées. Une autre formulation des contraintes de capacité (capacité instantanée par exemple) conduirait à des solutions différentes, peut être plus fractionnaires.

#### *I.1.c) La gestion des trajectoires artificielles*

La persistance des trajectoires artificielles dans la solution optimale a constitué une surprise de la mise en œuvre expérimentale. En effet, dans les problèmes classiques, les variables artificielles introduites par la méthode du grand M ont vocation à disparaître. La présence de trajectoires artificielles dans la solution optimale appelle donc un approfondissement du travail les concernant.

#### *i) Tests sur la présence des trajectoires artificielles*

Tout d'abord, on a vu que la persistance des trajectoires artificielles dans nos instances de tests provenait à la fois de la structure du problème (surcharge, liberté d'action restreinte à certains vols en l'air) et de l'implémentation pratique (capacité officielle de certains secteurs extrêmement différente de la charge prévue, données manquantes limitant le nombre de vols commandables...). Des tests supplémentaires sur des instances différentes en corrigeant éventuellement les données permettraient de déterminer la part résiduelle de trajectoires artificielles qui sera plus faible que celle observée dans nos instances.



## *ii) Tests sur la gestion des trajectoires artificielles*

Même si la part de trajectoires artificielles diminue, le but final poursuivi est quand même d'affecter une trajectoire réelle à chaque vol. Nous avons proposé dans le chapitre V (p. 137 paragraphe III.2.a) plusieurs stratégies possibles pour effectuer cette allocation une fois la résolution théorique terminée. Une expérimentation plus complète que celle ébauchée ici est alors nécessaire pour choisir la meilleure stratégie. Il faut évaluer les stratégies en termes de coût global de la solution finale et en termes de dépassement de la capacité de certains secteurs-périodes. En l'état actuel des tests, la méthode la plus prometteuse consiste à garder les trajectoires réelles allouées et à choisir la trajectoire de plus petit coût réduit pour les vols ayant une trajectoire artificielle.

## **1.2. Intégration de la résolution d'une instance dans le cadre du régulateur court terme**

Le travail de thèse présenté ici est centré sur la résolution d'une instance du problème d'optimisation posé par la régulation court terme du trafic aérien. Une extension logique et naturelle de ce travail réside dans l'étude de l'articulation entre les résolutions successives du problème d'optimisation.

### **1.2.a) Articulation entre les résolutions successives du problème d'optimisation**

#### *i) Liens entre deux résolutions successives*

La résolution d'une instance du problème d'optimisation n'est qu'une phase du processus de régulation court terme. Celui-ci est basé sur la commande optimale à horizon glissant. Cela implique que les actions de régulation seront réévaluées et éventuellement modifiées à chaque pas de régulation. Ainsi à chaque pas, les données sont réactualisées et la nouvelle instance obtenue est résolue de nouveau. Il est évident que deux instances successives du problème d'optimisation partageront un grand nombre de caractéristiques communes et il y a probablement possibilité de tirer parti des résolutions précédentes.

#### *ii) Définition et gestion de la banque de route 3D*

La banque de route 3D a justement été imaginée à cet effet. Elle a pour principe de stocker certaines trajectoires 3D intéressantes afin de les utiliser pour initialiser une nouvelle instance. Sa mise en œuvre effective demande cependant la résolution de plusieurs problèmes spécifiques. Tout d'abord, la manière dont sont codées les routes 3D dans cette banque jouera un rôle important dans la facilité d'associer une route 3D potentielle à un vol. En effet, étant donné que la position de chaque vol sera différente à chaque pas de régulation, il est crucial de disposer d'un stockage permettant de repérer facilement quelle route 3D pourrait convenir à un vol donné.

Une fois un format de stockage défini, il faut déterminer des règles d'entrée et de sortie des nouvelles routes 3D dans la banque. Enfin, ces règles doivent être évaluées en fonction de leurs conséquences sur la taille de la base de routes et sur le déroulement du processus de résolution. En effet la présence d'un ensemble de trajectoires initiales plus conséquent devrait permettre de diminuer encore le temps de calcul consacré à la résolution du problème d'optimisation par génération de colonnes.

## I.2.b) Evaluation du processus de régulation court terme

### i) *De programme testé en outil testeur*

L'outil de résolution élaboré au cours de ce travail pourra être combiné à un simulateur du trafic aérien afin de pouvoir étudier plus en détail le mécanisme même de la régulation court terme. Le programme permettant de résoudre une instance du problème d'optimisation ne sera alors plus l'objet évalué, mais l'outil permettant l'évaluation d'une méthode plus globale.

### ii) *Paramètres de la régulation court terme à étudier*

L'étude de la régulation court terme est un travail de recherche à part entière sur lequel le LICIT travaille déjà depuis quelques années. La présence d'un logiciel d'optimisation capable de résoudre le problème d'optimisation posé à chaque pas de régulation permettra alors d'étudier de nombreux facteurs et paramètres concernant ce nouveau mode de régulation :

- Influence de la réserve de carburant sur l'efficacité de la résolution
- Choix des différents horizons de temps, intervalle de temps entre deux pas
- Liens entre la régulation court terme et la régulation existante tactique
- Potentialités de la gestion dynamique de la capacité.

Toutes ces questions ne sont pas à proprement parler une prolongation de ce travail mais constituent tout un nouveau pan de l'étude de la régulation court terme.

La deuxième et dernière partie de ce chapitre est consacrée au bilan global de l'application de la méthode de génération de colonnes au problème d'optimisation.

## **II. Bilan global de l'application de la méthode de génération de colonnes**

### II.1. Objectifs initiaux et résultats obtenus

#### II.1.a) Objectifs initiaux

L'objectif initial fixé à ce travail était de modéliser et de résoudre une instance donnée du problème d'optimisation provenant de la régulation court terme du trafic aérien. Cette résolution devait s'effectuer en un temps de calcul compatible avec le cadre général de la régulation par commande à horizon glissant. La modélisation du système aérien et du problème d'optimisation devait être suffisamment fine pour pouvoir remplir efficacement les objectifs de la régulation court terme : protection des contrôleurs contre les surcharges, minimisation des conséquences de la régulation et lissage de la charge. La modélisation des actions de régulation devait permettre de respecter toutes les règles de navigation aérienne. Enfin le calcul des conséquences des actions de régulation devait être aussi précis que possible afin de diminuer l'impact de l'élément le plus limitant de la régulation aérienne : l'incertitude.

## II.1.b) Résultats principaux

### *i) Modélisation du système aérien et des trajectoires*

Le réseau aérien a été modélisé à partir du concept des feuilles d'espace, englobant routes aériennes, règles d'utilisation de ces routes et secteurs aériens. Cette modélisation a servi de base à la détermination des trajectoires avec une précision tout à fait inédite dans tous les travaux consacrés à cette famille de problèmes : prise en compte de la trajectoire balise par balise et des niveaux aériens, modélisation fine du changement de niveau de vol à partir de données fournies par un simulateur aérien, prise en compte des performances de l'avion en fonction du type d'appareil. Les correspondances des vols sont prises en compte dans le modèle pour estimer plus précisément le coût d'un retard.

### *ii) Instance principale*

L'instance principale représente la journée du 10 septembre 2003 à midi et a été obtenue à partir de données réelles. La protection des secteurs contre la surcharge est assurée sur l'espace aérien européen et limitrophe dont la CFMU est en charge. Elle représente 1112 vols commandables pour lesquels on cherche une trajectoire réalisable et 1034 secteurs protégés pendant une durée de 3 heures. Le problème d'optimisation a alors été résolu en 2131 secondes soit 35 minutes et 31 secondes sur un PC de bureau cadencé à 3,20 GHz. L'écart entre la valeur de la solution obtenue et la borne inférieure représente 0.05% de la valeur de la fonction objectif. La méthode a ensuite été appliquée avec le même succès à une dizaine d'instances différentes mais concernant toutes la journée du 10 septembre 2003.

### *iii) Extensions possibles*

La structure de résolution associée à la modélisation du système aérien permet de prendre en compte des données supplémentaires sans modification de la complexité des algorithmes utilisés. Ainsi, la modélisation d'une trajectoire qui est pour le moment effectuée à partir de segments pourrait être affinée en prenant en compte des parcours non rectilignes. De nombreuses données supplémentaires peuvent être intégrées au modèle, telles la météo ou la masse précise de l'avion.

## II.2. Bilan global

### II.2.a) Limites détectées et réponses apportées

Deux questions principales restent posées à l'issue des tests effectués : le caractère peu fractionnaire de la solution relaxée et la persistance de trajectoires artificielles dans la solution finale. Nous avons déjà évoqué ces deux phénomènes et expliqué quels tests et études permettraient de répondre le plus précisément possible à ces interrogations. On peut cependant remarquer sans attendre qu'elles ne remettent pas du tout en cause l'utilisation de la méthode de génération de colonnes.

### *i) Processus d'obtention d'une solution entière*

Au cours du processus de génération de colonnes, un grand nombre de trajectoires sont calculées pour la résolution du problème relaxé. Ainsi même si on obtenait une solution relaxée très fractionnaire nécessitant la création d'un arbre de branch-and-price, le nombre de

trajectoires nouvelles à construire pour résoudre chaque problème fils serait très restreint. Etant donné que la résolution des sous programmes linéaires par CPLEX est quasiment instantanée, la résolution de chaque sous problème par génération de colonnes ou l'obtention d'une bonne borne inférieure serait très rapide. Par ailleurs, la taille des instances traitées est inférieure à la taille des problèmes de référence que CPLEX est capable de résoudre en nombres entiers. Le processus d'obtention d'une solution entière n'est donc pas limitatif de la méthode.

## ii) *Trajectoires artificielles*

La persistance de trajectoires artificielles dans la solution optimale est plus une question à traiter au niveau du régulateur court terme qu'au niveau de la résolution du problème d'optimisation. En effet, à partir du moment où le problème n'admet pas de solution réalisable, n'importe quelle méthode donnerait une solution avec des trajectoires artificielles, voire pas de solution du tout. Ainsi la présence de trajectoires artificielles n'est pas un argument contre la méthode employée, mais plutôt une démonstration de sa capacité à fournir des éléments précieux même quand le problème n'est pas résoluble : vols critiques, secteurs saturés, ensemble de trajectoires intéressantes pour les vols et choix d'une trajectoire réelle pour la grande majorité des vols. Par ailleurs, comme écrit plus tôt le nombre de trajectoires artificielles devrait diminuer avec une meilleure connaissance des données représentant le système aérien et avec la présence d'un filtre tactique de régulation.

### II.2.b) Les atouts de la méthode

Nous espérons avoir montré au cours de ce travail que la méthode de génération de colonnes est particulièrement adaptée au problème d'optimisation posé. Très peu d'autres méthodes permettent en effet une modélisation aussi fine des trajectoires des avions, finesse indispensable à une régulation efficace. Par ailleurs le temps de calcul global obtenu sur un ordinateur de bureau, de l'ordre de 35 minutes, est complètement compatible avec une application dans le cadre de la commande à horizon glissant. De plus, une optimisation du code informatique et l'utilisation de moyens de calcul plus puissants permettrait de diminuer drastiquement le temps de calcul global. En outre une grande partie des calculs peuvent être effectués sur des machines en parallèle diminuant encore le temps de calcul nécessaire.

Enfin, cette méthode peut même être utilisée partiellement dans le cadre d'un système d'aide à la décision : elle permet d'identifier les vols problématiques et les secteurs saturés tout en fournissant une banque de trajectoires pour chaque vol. On peut donc apporter une réponse positive à la question concernant l'applicabilité de la méthode de génération de colonnes à la résolution du problème d'optimisation provenant de la régulation court terme du trafic aérien.



# Appendices

## I. Bibliographie

### I.1. Fonctionnement de l'ATFM et de la CFMU, recherches globales

- [1] H.D. Sherali, R.W. Staats, A.A. Trani, *An airspace-planning and collaborative decision-making model: Part II Cost model, Data consideration, and computations*, Transportation Science, Vol. 40 N°2 p.147-164, Mai 2006
- [2] Performance Review Commission, *An Assessment of Air Traffic Management in Europe during the Calendar Year 2005*, Bruxelles, Avril 2006
- [3] R. Fondacci, B. Fontaine, O. Richard, *Rapport final de recherche, convention EUROCONTROL-INRETS C/1.195/CE/NB/03 Document 1 Régulation court terme du trafic aérien*, p.9-127 Bron, Décembre 2005
- [4] Commission d'examen des performances, *Rapport d'examen des performances, Evaluation de la gestion de la circulation aérienne en Europe au cours de l'année civile 2004*, Bruxelles, Avril 2005
- [5] Eurocontrol, *Air Traffic Flow and Capacity Management Strategy, Edition 1.2*, Bruxelles, Avril 2004
- [6] Université de Westminster, Commission d'examen des performances, *Evaluating the true cost to airlines of one minute of airborne or ground delay, Edition 4*, Bruxelles, Février 2004
- [7] Eurocontrol, *Route availability document Basic CFMU Handbooks Appendix 1*, Bruxelles, Mai 2003
- [8] Eurocontrol, *ATFM Users Manual, Edition 9.0*, Basic CFMU Handbooks, Bruxelles, Avril 2003
- [9] Eurocontrol, *General & CFMU Systems, Edition 8.0*, Basic CFMU Handbook, Bruxelles, Mars 2002
- [10] C. Pusch, *Air Traffic Management*, Air & Space Europe, vol.3 No 374, 2001
- [11] P. Matos, R. Ormerod, *The application of operational research to European air traffic management – understanding the context*, European Journal of Operational Research, 123 p.125-144, 2000.
- [12] R. Beatty, R. Hsu, L. Berry, J. Rome, *Preliminary evaluation of flight delay propagation through an airline schedule*, Proceedings 2<sup>nd</sup> USA/Europe ATM R&D Seminar, Orlando 1-4 Décembre 1998
- [13] R. Fondacci, A. Vallée, *Cost function in ATFM optimization problems*, UE project NOAA final report, Workpackage 3.3.a, 1996
- [14] A. Odoni, *Issues in Air Traffic Flow Management*, in Proceedings of Conference on Advanced Technologies for Air Traffic flow Management : Chapter in “Advanced technologies for Air Traffic Flow Management” H. Winter and H.G. Nusser, editors, Springer-Verlag, Berlin, p. 43-63, 1994

- [15] W. Philipp, F. Gainche, *Air Traffic Flow Management in Europe*, in Proceedings of Conference on Advanced Technologies for Air Traffic flow Management, Deutsche Forschungsanstalt für Luft und Raumfahrt e.V. (DLR), Bonn, 1994

## 1.2. Le Ground Holding Problem et ses extensions

- [16] B. Kotnyek, O. Richetta, *Equitable models for the stochastic ground-holding problem under collaborative decision making*, Transportation Science, Vol. 40 N°2 p.133-146, Mai 2006
- [17] J.M. van den AKKER, K. Nachtigall, *Slot Allocation by column generation*. NLR Technical Publication 97286, Submitted to European Journal of Operational Research, 1997
- [18] R. Hoffman, M.O. Ball, *A comparison of formulations for the single-airport ground holding problem with banking constraints*, Technical Research Report ISR, Octobre 1997
- [19] G. Andreatta, A.R. Odoni, and O. Richetta, *Models for the ground holding problem*, In L. Bianco and A.R. Odoni, editors, Large Scale Computation and Information Processing in Air Traffic Control, Transportation Analysis, p.125-168. Springer-Verlag, 1993
- [20] A.R. Odoni, *The flow management problem in air traffic control*, In A.R. Odoni et al, editor, Flow Control of Congested Networks, volume F38 of ASI Series, p.269-288, NATO, 1987.
- [21] G. Andreatta, G. Romanin-Jacur, *Aircraft Flow Management under Congestion*, Transportation Science, Vol. 21 N°4, Novembre 1987

## 1.3. Le TFMP et ses extensions

- [22] O. Richard, W. Bienia, M. Queyranne, R. Fondacci, *Column generation for dynamic ATFM*, In Proceedings of the 3<sup>rd</sup> EUROCONTROL Innovative Research Workshop, Brétigny sur Orge, 9,10 Décembre 2004
- [23] N. Barnier, P. Brisset, T. Rivière, *Slot allocation with constraint programming : Models and result*, 4th USA/EUROPE ATM R&D Santa Fé, Décembre 2001
- [24] D. Bertsimas, S. Stock Patterson, *The traffic flow management rerouting Problem in air traffic control: a dynamic network Flow Approach*, Transportation Science, vol. 34, N°3 p.239-255, Août 2000
- [25] A. Alonso, L.F. Escudero, M.T. Ortuno, *A stochastic 0-1 program based approach for the air traffic flow management problem*, European Journal of Operational Research, 120 p.47-62, 2000
- [26] D. Bertsimas, S. Stock, *The air traffic flow Management problem with enroute capacities*, Operations Research, 46: p.406-422, 1998
- [27] S. Oussedik, D. Delahaye, M. Schoenauer, *Air Traffic Management by Stochastic Optimization* International Air Traffic Management R&D seminar, Orlando, Décembre 1998
- [28] V.Tosic, O. Babic, M. Cangalovic and D. Hohlacov, *A Model to Solve En Route Air Traffic Flow Management Problem: A Temporal and Spatial Case*, rapports du séminaire ATFM 1997, Saclay France, 17/20 Juin 1997

- [29] P. Vranas, *Optimal Slot Allocation for European Air Traffic Flow Management*, Air Traffic control Quaterly, Vol 4(4) p.249-280, 1997
- [30] L. Maugis, *Mathematical programming for the air traffic flow management problem with en-route capacities*, IFOR, 1996.
- [31] V. Tomic, O. Babic, M. Cangalovic and D. Hohlacov, *Some models for en route air traffic flow Management*, Transportation Planning and Technology, 19 (2) p.147-164, 1995

#### I.4. Problèmes de routage et de reroutement dynamique

- [32] S. Constans, N. Gadenne and R. Fondacci, *Applying genetic techniques to the tactical flight level assignment*, Advanced OR and AI Methods in Transportation, Proc. of the 10 Jubilee Meeting of the EURO Working Group on Transportation and 16 Mini-Euro Conference, Andrzej Jaskiewicz Ed., p. 69-75, 2005
- [33] A. Nilim, L. El Ghaoui and V. Duong, *Multi-Aircraft Routing and Traffic Flow Management under Uncertainty*, 5th USA/EUROPE ATM R&D Budapest, Juin 2003
- [34] N. Barnier, P. Brisset, *Graph coloring for Air Traffic Flow Management*, Proceedings CPAIOR'02 p.1-15, 2002
- [35] R. Fondacci, O. Goldschmidt, V. Letrouit, *Combinatorial Issues in Air Traffic Optimization*, Transportation Science, Vol.32 N° 3, Août 1998

#### I.5. Programmation linéaire et applications, génération de colonnes

- [36] J. Desrosiers, M.E. Lübbecke, *A primer in column generation*, Les cahiers du GERAD, G-2004-02, Janvier 2004
- [37] M.E. Lübbecke, J. Desrosiers, *Selected topics in column generation*, Les cahiers du GERAD, G-2002-64 2002, révisé Octobre 2004
- [38] R.E. Bixby, *Solving Real-World linear Programs: a decade and more of progress*, Operations Research, 50 p. 3-15, 2002
- [39] W.E. Wilhelm, *A technical review of column generation in integer programming*, Optimization and Engineering, 2 p.159-200, 2001
- [40] R.E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, R. Wunderling, *MIP Theory and practice, closing the gap*, In M. J. D. Powell and S. Scholtes, editors, System Modelling and Optimization: Methods, Theory, and Applications, Kluwer Academic Publishers, p. 19-49, 2000
- [41] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, P.H. Vance, *Branch and Price Column generation for solving huge integer programs*, Operations Research, Vol 46 N° 3, Mai-Juin 1998
- [42] A.A. Farley, *A note on bounding a class of linear programming problems, including cutting stock problems*, Operational research, 38 (5) p.922-923, 1990
- [43] J. Desrosiers, F. Soumis, M. Desrochers, *Routing with time windows by column generation*, Networks, 14 p.545-565, 1984



- [44] D.M. Ryan, B.A. Foster, *An Integer Programming approach to Scheduling*, Computer scheduling of public Transport Urban Passenger Vehicle and Crew Scheduling. A Wren (ed.) North Holland Amsterdam p.269-280, 1981
- [45] P.C. Gilmore, R.E. Gomory, *A linear programming approach to the cutting stock problem Part II*, Operational Research, 11 p.863-888, 1963
- [46] P.C. Gilmore, R.E. Gomory, *A linear programming approach to the cutting stock problem*, Operational Research, 9 p.849-859, 1961
- [47] G.B. Dantzig, P. Wolf, *Decomposition principle for linear programs*, Operational Research, 8 p.100-111, 1960

## I.6. Problèmes de plus courts chemins

- [48] A.V. Goldberg, C. Harrelson, *Computing the shortest path: A\* Search meets Graph Theory*, Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms p.156-165, 2005
- [49] R. Gutman, *Reach-based routing: a new approach to shortest path algorithms optimized for road networks*, In Proc. Algorithm engineering and experimentation: sixth annual international workshop, 2004
- [50] R.K. Ahuja, J.B. Orlin, S. Pallottino, M. G. Scutellà, *Dynamic Shortest Paths Minimizing Travel Times and Costs*, Networks, vol. 41 (4) p.197-205, 2003
- [51] K. Melhorn, M. Ziegelmann, *Resource constrained shortest paths*, Proceedings 8<sup>th</sup> European Symp. Algorithms (ESA 2000) Lecture Notes in Computer Science 1879, Springer-Verlag, p.326–337, 2000
- [52] S. Pallottino, M.G. Scutellà, *Shortest path algorithms in transportation models: classical and innovative aspects*, In P. Marcotte and S. Nguyen, editors, Equilibrium and Advanced Transportation Modelling, Kluwer, p.245-281, 1998.
- [53] D.P. Bertsekas, *A simple and fast label correcting algorithm for shortest paths*, Networks, Vol 23 p.703-709, 1993
- [54] M. Desrochers, F. Soumis, *A generalized permanent labelling algorithm for the shortest path problem with time windows*, INFOR, 26 p.191-212, 1988
- [55] F. Glover, R. Glover, D. Klingman, *The threshold shortest path algorithm*, Networks, Vol. 14 n°1, 1986
- [56] R.E. Tarjan, *Data structures and network algorithms*, SIAM Philadelphia PA, 1983
- [57] G. Handler, I. Zang, *A dual algorithm for the constrained shortest path problem*, Networks, 10 p. 281-291, 1980
- [58] M. Gondran, M. Minoux, *Graphes et algorithmes*, Paris, Eyrolles, Coll. Etudes et recherches d'électricité de France, 1979
- [59] U. Pape, *Implementation and efficiency of Moore-algorithms for the shortest route problem*, Mathematical Programming, 7 p. 212-222, 1974
- [60] E. W. Dijkstra, *A note on two problems in connection with graphs*, Numerische Mathematik, 1 p. 269-271, 1959
- [61] R.E. Bellman, *On a routing problem*, Quarterly of Applied Mathematics, vol. 16, p.87-90, 1958

[62] L. Ford, *Network Flow Theory*, Technical Report P-932, The Rand Corporation, 1956

### I.7. Optimisation combinatoire: autres.

[63] J.P. Lebacque, *Programmation dynamique, principes et résultats fondamentaux*, Cours de méthodes de la recherche opérationnelle ENTPE, 2005

[64] P. Toth, D. Vigo, *The Vehicle Routing Problem*, SIAM, Philadelphie, 2001

[65] A. Aho, J. Hopcroft, J. Ullman, *Structures de données et algorithmes*, Interéditions, 1988

[66] M.L. Fisher, R. Jaikumar, L.N. Van Wassenhove, *A Multiplier Adjustment Method for the Generalized Assignment Problem*, *Management Science*, 32 p.1095-1103, 1986

[67] L.R. Ford, D.R. Fulkerson, *A suggested computation for maximal multicommodity network flows*, *Management Science*, 5 p. 97-101, 1958

[68] R. Bellman, *Dynamic Programming*, Princeton Univ. Press, 1957

### I.8. Sites Internet

[69] Eurocontrol, IATA, ACI Europe, *European CDM Portal* <http://www.euro-cdm.org/> accédé le 16/11/2006

[70] Eurocontrol STATFOR, *Air Traffic Statistics and Forecasts* [http://www.eurocontrol.int/statfor/public/subsite\\_homepage/homepage.html](http://www.eurocontrol.int/statfor/public/subsite_homepage/homepage.html) accédé le 16/11/2006

## II. Annexes

### II.1. Descriptions et démonstrations mathématiques

Les notations, sauf mention contraire, sont celles décrites dans le Tableau 2) Liste des notations utilisées p. 53

#### II.1.a) Description matricielle du problème maître

$$\text{Minimiser } z = \sum_{f \in \mathcal{F}} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} X_{f,r} \right) + \sum_{g \in \mathcal{C}} c_g r_g = c \cdot X$$

Sous

$$\begin{pmatrix} \delta_{f,r,s,p} \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} \leq \begin{pmatrix} C_{s,p} \end{pmatrix}$$

$$\begin{pmatrix} 1111100000000000 \\ 0000011110000000 \\ \\ \\ \\ \\ \\ \\ \\ 00000000000001111 \end{pmatrix} \begin{pmatrix} 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} a_{f,r} \text{---} a_{f,r} 0 \text{-----} 0 \\ a_{f,r} \text{---} a_{f,r} 0 \text{-----} 0 \\ 0 \text{-----} 0 a_{f,r} \text{---} a_{f,r} 0 \text{---} 0 \\ 0 \text{-----} 0 a_{f,r} a_{f,r} \end{pmatrix} \begin{pmatrix} -10 & 0 \\ 0-10 & \text{-----} 0 \\ 0 \text{-----} 0-10 \text{-----} \\ 0 \text{-----} 0-1 \end{pmatrix} \leq \begin{pmatrix} d_{g_1-k} \\ d_{g_2-k} \\ \\ \\ d_{g_n-k} \end{pmatrix}$$

avec  $X = \begin{pmatrix} X_{f_1,r} \\ | \\ | \\ X_{f_m,r} \\ r_{g_1} \\ | \\ r_{g_n} \end{pmatrix}$

## II.1.b) Bornes inférieures du programme linéaire

### i) Première borne inférieure

Soit le programme linéaire suivant résolu par la méthode de génération de colonnes :

$$\text{Minimiser } z = \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in C} c_g r_g$$

Sous  $A \cdot x = b$

Avec  $x \geq 0$

A une étape quelconque du processus de génération de colonnes, on a résolu le problème maître restreint et on a obtenu le vecteur des valeurs duales associées aux contraintes du problème primal noté  $\pi$ . On note  $B$  la base optimale et  $(\bar{x}, \bar{r})$  la solution optimale obtenue. La valeur du critère  $z$  à cette étape est notée  $\bar{z}$  et l'on note  $c'_{f,r}$  et  $c'_g$  la valeur des coûts réduits dans cette base. Par définition on a :

$$c'_{f,r} = c_{f,r} - \pi^T A_{f,r}$$

$c'_g = c_g - \pi^T A_g$  où  $A_{f,r}$  est la colonne de la matrice  $A$  correspondant à la variable  $x_{f,r}$  et  $A_g$  à la variable  $r_g$ .

Soit  $(x,r)$  une solution quelconque du problème maître :

$$\begin{aligned} z &= \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in C} c_g r_g \\ &= \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c_{f,r} x_{f,r} \right) + \sum_{g \in C} c_g r_g + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g - \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) - \sum_{g \in C} c'_g r_g \\ &= \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} (c_{f,r} - c'_{f,r}) x_{f,r} \right) + \sum_{g \in C} (c_g - c'_g) r_g + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g \end{aligned}$$

Or  $c_{f,r} - c'_{f,r} = \pi^T A_{f,r}$  et  $c_g - c'_g = \pi^T A_g$

$$\begin{aligned} z &= \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} \pi^T A_{f,r} x_{f,r} \right) + \sum_{g \in C} \pi^T A_g r_g + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g \\ &= \pi^T \left( \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} A_{f,r} x_{f,r} \right) + \sum_{g \in C} A_g r_g \right) + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g \end{aligned}$$

$(x,r)$  est solution du problème maître, d'où

$$z = \pi^T b + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g$$

Etant donné que  $\pi$  est le vecteur dual associé à la solution obtenue lors de la résolution du problème maître restreint on a  $\pi^T b = \bar{z}$

D'où

$$z = \bar{z} + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right) + \sum_{g \in C} c'_g r_g$$

Tout d'abord on peut remarquer que  $c'_g \geq 0$  car  $c'_g$  représentent les coûts réduits dans la base  $B$  qui est optimale pour le problème maître restreint et les variables  $r_g$  du problème maître sont toutes présentes dans le problème maître restreint. De plus les variables  $r_g$  sont positives :

$$z \geq \bar{z} + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} c'_{f,r} x_{f,r} \right)$$

On note  $\bar{c}_f^* = \min(c'_{f,r} \forall r \in \mathcal{R}_f)$

Alors  $z \geq \bar{z} + \sum_{f \in F} \left( \sum_{r \in \mathcal{R}_f} \bar{c}_f^* x_{f,r} \right)$  car  $x_{f,r} \geq 0$

$$z \geq \bar{z} + \sum_{f \in F} \left( \bar{c}_f^* \sum_{r \in \mathcal{R}_f} x_{f,r} \right) = \bar{z} + \sum_{f \in F} \bar{c}_f^*$$

On en déduit que  $\bar{z} + \sum_{f \in F} \bar{c}_f^*$  est une borne inférieure du problème maître.

ii) Deuxième borne inférieure

On se place dans le même cas que pour le calcul de la première borne inférieure : on a résolu un problème maître restreint et on dispose de la solution  $(\bar{x}, \bar{r})$ , des valeurs duales  $\pi = (\pi_{s,p}, \pi_g)$  et de la valeur du critère  $\bar{z}$ .

On pose

$$\lambda_{f,r} = \pi^T A_{f,r} = \sum_{s,p} (\delta_{f,r,s,p} \pi_{s,p}) + \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,g}$$

c'est-à-dire  $c'_{f,r} = c_{f,r} - \lambda_{f,r}$

Si  $\lambda_{f,r} \leq 0 \forall f \in F, \forall r \in \mathcal{R}_f$

On a  $c'_{f,r} \geq 0 \forall f \in F, \forall r \in \mathcal{R}_f$  car  $c_{f,r} \geq 0$

et  $(\bar{x}, \bar{r})$  est optimal pour le problème maître relaxé. On se place dans le cas où on n'est pas encore arrivé à l'optimalité du problème maître relaxé.

Soit  $K_1 = \text{Min}(\frac{c_{f,r}}{\lambda_{f,r}}; \forall f \in F, \forall r \in \mathcal{R}_f / \lambda_{f,r} > 0)$

On veut montrer que  $K_1 \pi$  est solution du problème dual du problème maître relaxé, problème qui s'écrit :

$$\text{Maximiser } \sum_{s \in S} \sum_{p \in \mathcal{P}} C(s,p) * \pi_{s,p} + \sum_{f \in F} \pi_f + \sum_{f \in F} \sum_{g \in C_f} (d_g - k) \pi_{f,g}$$

Sous

$$\sum_{s \in S} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \pi_{s,p} + \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,g} \leq c_{f,r} \quad \forall f \in F, r \in \mathcal{R}_f \quad (1)$$

$$\sum_{f/g \in C_f} -\pi_{f,g} \leq c_g \quad \forall g \in C \quad (2)$$

$$\pi_{s,p} \leq 0 \quad \forall s \in S, p \in \mathcal{P} \quad (3)$$

$$\pi_{f,g} \leq 0 \quad \forall f \in F \quad \forall g \in C_f \quad (4)$$

On peut remarquer que  $\pi$  est solution du problème dual du problème maître restreint :

On a déjà  $\pi_{s,p} \leq 0 \forall s \in S, p \in \mathcal{P}, \pi_{f,g} \leq 0 \forall f \in F \quad \forall g \in C_f$  et  $\lambda_{f,r} \sum_{f/g \in C_f} -\pi_{f,g} \leq c_g \quad \forall g \in C$  car

tous les vols et tous les vols en correspondance sont présents dans le problème maître relaxé.

Comme  $K_1 > 0$  par construction,  $K_1 \pi$  vérifie les équations (3) et (4).

On n'est pas encore arrivé à l'optimalité du problème maître relaxé :

$$\exists (f,r) / c'_{f,r} < 0 \text{ avec } c'_{f,r} = c_{f,r} - \lambda_{f,r}$$

$$\exists (f,r) / c_{f,r} - \lambda_{f,r} < 0$$

$$\text{Comme } c_{f,r} > 0 \quad \exists (f,r) / 0 < c_{f,r} < \lambda_{f,r}$$

$$\exists (f,r) / > 0 \text{ et } \frac{c_{f,r}}{\lambda_{f,r}} < 1$$

On en déduit que  $K_1 < 1$ .

On a  $\sum_{f/g \in C_f} -\pi_{f,g} \leq c_g \quad \forall g \in C, -\pi_{f,g} \geq 0$  et  $0 < K_1 < 1$ , d'où  $\sum_{f/g \in C_f} -K_1 \pi_{f,g} \leq c_g \quad \forall g \in C$

$K_1 \pi$  vérifie l'équation (2)

On étudie l'équation (1) pour  $K_1 \pi$

$$\begin{aligned}
\text{Soient } f \in \mathcal{F}, r \in \mathcal{R}_f & \left( \sum_{s \in S} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} K_1 \pi_{s,p} + K_1 \pi_f + \sum_{g \in C_f} a_{f,r} K_1 \pi_{f,r,g} \right) \\
& = K_1 \left( \sum_{s \in S} \sum_{p \in \mathcal{P}} \delta_{f,r,s,p} \pi_{s,p} + \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,r,g} \right) \\
& = K_1 \lambda_{f,r}
\end{aligned}$$

premier cas :  $\lambda_{f,r} \leq 0$  alors  $K_1 \lambda_{f,r} \leq 0 < c_{f,r}$  et l'équation (1) est vérifiée

deuxième cas  $\lambda_{f,r} > 0$  alors par définition  $K_1 \leq \frac{c_{f,r}}{\lambda_{f,r}} \forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f / \lambda_{f,r} > 0$  et l'équation

(1) est vérifiée aussi.

On en déduit que  $K_1 \pi$  est solution du problème dual du problème maître. Par le théorème fondamental de la dualité.

$K_1 \pi = K_1 \bar{z}$  est une borne inférieure du problème maître

$K_1 \bar{z}$  est une borne inférieure du problème maître avec

$$K_1 = \text{Min} \left( \frac{c_{f,r}}{\lambda_{f,r}} ; \forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f / \lambda_{f,r} > 0 \right)$$

Cette borne théorique n'est cependant pas facile à calculer : on va en tirer une autre borne plus facilement calculable durant l'algorithme.

On pose tout d'abord  $c_f = \text{Min}(c_{f,r} ; \forall r \in \mathcal{R}_f)$

On a ensuite  $\lambda_{f,r} = \sum_{s,p} (\delta_{f,r,s,p} \pi_{s,p}) + \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,g}$  avec  $\pi_{s,p} \leq 0$

D'où  $\lambda_{f,r} \leq \pi_f + \sum_{g \in C_f} a_{f,r} \pi_{f,g}$

Ensuite  $a_f \leq a_{f,r}$  car les trajectoires arrivant en avance sont considérées comme arrivant à l'heure (pas de bonification à l'arrivée en avance) où  $a_f$  est l'heure d'arrivée prévue pour le vol  $f$ .

Comme  $\pi_{f,g} \leq 0$

$$\lambda_{f,r} \leq \pi_f + \sum_{g \in C_f} a_f \pi_{f,g}$$

$$\forall f \in \mathcal{F}, \forall r \in \mathcal{R}_f / \lambda_{f,r} > 0 \quad c_f \leq c_{f,r} \text{ et } 0 < \lambda_{f,r} \leq \pi_f + \sum_{g \in C_f} a_f \pi_{f,g}$$

$$\text{d'où } \frac{c_f}{\pi_f + \sum_{g \in C_f} a_f \pi_{f,g}} \leq \frac{c_{f,r}}{\lambda_{f,r}}$$

On en déduit donc que

$$K_2 = \text{Min} \left( \frac{c_f}{\pi_f + \sum_{g \in C_f} a_f \pi_{f,g}}, f \in \mathcal{F} / \pi_f + \sum_{g \in C_f} a_f \pi_{f,g} > 0 \right) \leq \text{Min} \left( \frac{c_{f,r}}{\lambda_{f,r}}, f \in \mathcal{F}, r \in \mathcal{R}_f / \lambda_{f,r} > 0 \right) = K_1$$

$K_2 \bar{z}$  est une borne inférieure du problème maître avec

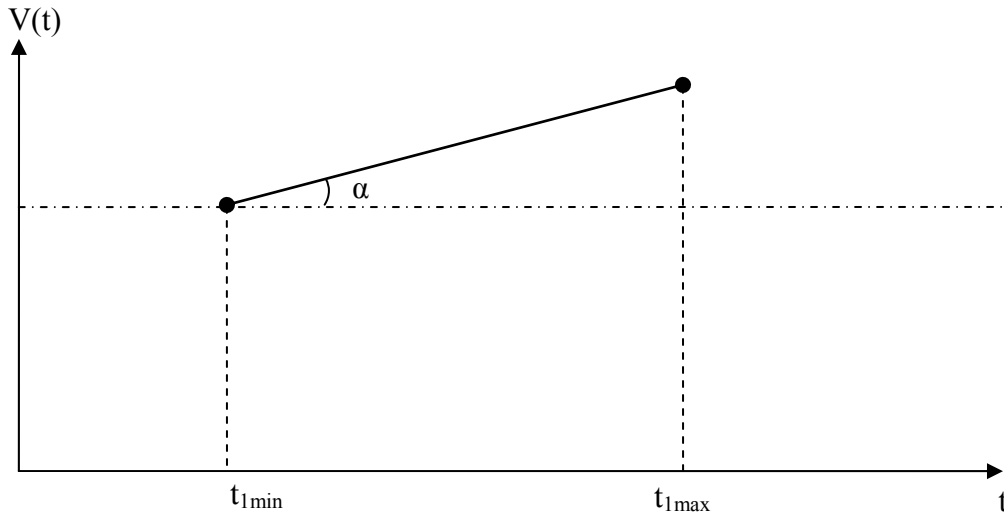
$$K_2 = \text{Min} \left( \frac{c_f}{\pi_f + \sum_{g \in C_f} a_f \pi_{f,g}}, f \in \mathcal{F} / \pi_f + \sum_{g \in C_f} a_f \pi_{f,g} > 0 \right)$$

## II.1.c) Résolution de l'équation de Hamilton-Jacobi-Bellman dans un cas particulier

Soit  $[t_{1\min}, t_{1\max}]$  un intervalle sur lequel la fonction potentiel est linéaire (voir Figure 45):

$$V(t_1) = \alpha t_1 + b$$

On suppose que l'on ne peut partir de l'étape 1 qu'entre  $t_{1\min}$  et  $t_{1\max}$ .



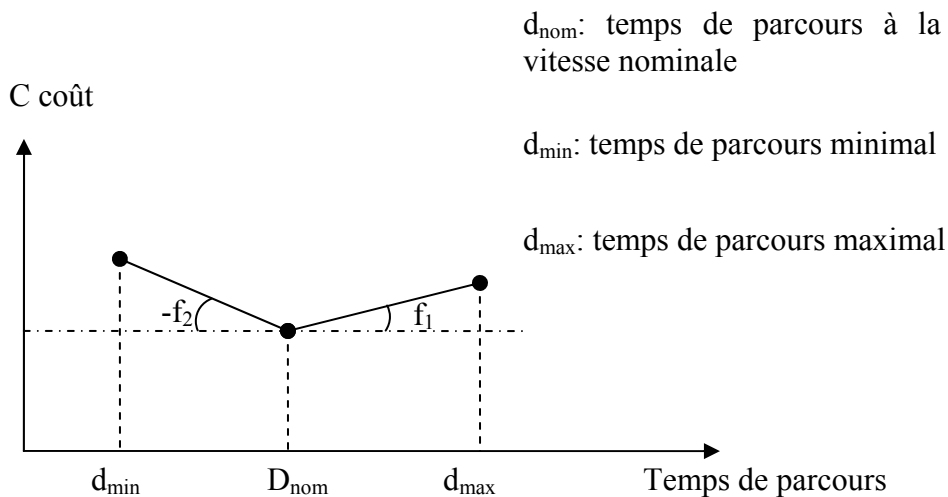
**Figure 45) Fonction potentiel à l'étape 1**

On souhaite calculer la fonction potentiel à la prochaine étape dans le cas de la fonction de transition entre l'étape 1 et l'étape 2 suivante (voir Figure 46) :

$$\text{Si } (t_2 - t_1) \geq d_{\text{nom}} \quad C(t_2 - t_1) = C(d_{\text{nom}}) + f_1 [(t_2 - t_1) - d_{\text{nom}}]$$

$$\text{Si } (t_2 - t_1) \leq d_{\text{nom}} \quad C(t_2 - t_1) = C(d_{\text{nom}}) - f_2 [(t_2 - t_1) - d_{\text{nom}}]$$

avec  $f_1, f_2 > 0$



**Figure 46) Fonction de transition entre l'étape 1 et l'étape 2.**

Dans notre problème spécifique de programmation dynamique  $\alpha$  ne peut prendre que deux valeurs :  $f_1$  ou  $-f_2$  car la composante en escalier représentant les coûts des secteurs-périodes a

une pente nulle et la composante dépendant des retards à l'arrivée n'intervient que dans la dernière étape (arrivée à l'aéroport). Ici on se limite donc à ces deux cas, sachant que le cas général ne présente pas de difficultés particulières : on obtient une fonction à segments convexe de pente  $-f_2$ ,  $f_1$  et  $\alpha$  classés dans l'ordre croissant des pentes.

On souhaite calculer le potentiel à l'étape 2. On note  $\mathcal{P}(t_2)$  l'ensemble des prédécesseurs de  $t_2$  :

$$\mathcal{P}(t_2) = [t_{1\min}; t_{1\max}] \cap [t_2 - d_{\max}; t_2 - d_{\min}] = [x_1; x_2]$$

avec  $x_1 = \text{Max} \{t_{1\min}; t_2 - d_{\max}\}$   $x_2 = \text{Min} \{t_{1\max}; t_2 - d_{\min}\}$   
 On a alors  
 $V(t_2) = \infty$  pour  $t_2 < t_{1\min} - d_{\min}$  ou  $t_2 > t_{1\max} + d_{\max}$   
 $V(t_2) = \text{Min} \{ V(t_1) + C(t_2 - t_1); t_1 \in \mathcal{P}(t_2) \}$

$$V(t_2) = \text{Min} \{ \alpha t_1 + b + C(t_2 - t_1); t_1 \in [x_1; x_2] \}$$

On note  $x_{\text{médian}} = t_2 - d_{\text{moy}}$

i) *Premier cas :  $x_{\text{médian}} \leq x_1$*

On a alors  $x_{\text{médian}} = t_2 - d_{\text{moy}} \leq x_1 = \text{Max} \{t_{1\min}; t_2 - d_{\max}\}$   
 Or  $d_{\text{nom}} < d_{\max} \Leftrightarrow -d_{\text{nom}} > -d_{\max}$   
 $\Leftrightarrow t_2 - d_{\text{nom}} > t_2 - d_{\max}$

On en déduit que  $\text{Max} \{t_{1\min}; t_2 - d_{\max}\} = t_{1\min} = x_1$

De plus  $x_{\text{médian}} = t_2 - d_{\text{moy}} \leq x_1 = t_{1\min}$

D'où  $t_2 \leq t_{1\min} + d_{\text{moy}} \leq t_1 + d_{\text{moy}}$

$t_2 - t_1 \leq d_{\text{moy}}$

$$\begin{aligned} V(t_2) &= \text{Min} \{ \alpha t_1 - f_2 ((t_2 - t_1) - d_{\text{nom}}) + C(d_{\text{nom}}); t_1 \in [x_1; x_2] \} + b \\ &= \text{Min} \{ \alpha t_1 - f_2 ((t_2 - t_1) - d_{\text{nom}}); t_1 \in [t_{1\min}; x_2] \} + b + C(d_{\text{nom}}) \\ &= \text{Min} \{ (\alpha + f_2) t_1; t_1 \in [t_{1\min}; x_2] \} + f_2 (d_{\text{nom}} - t_2) + b + C(d_{\text{nom}}) \end{aligned}$$

▪ Premier sous cas :  $\alpha = f_1$

Comme  $f_1, f_2 > 0$   $(\alpha + f_2) > 0$

$$\boxed{V(t_2) = (f_1 + f_2) t_{1\min} + f_2 (d_{\text{nom}} - t_2) + b + C(d_{\text{nom}})}$$

▪ Deuxième sous cas  $\alpha = -f_2$

$$\boxed{V(t_2) = f_2 (d_{\text{nom}} - t_2) + b + C(d_{\text{nom}})}$$

ii) *Deuxième cas  $x_1 \leq x_{\text{médian}} \leq x_2$*

$x_{\text{médian}} \geq x_1 = \text{Max} \{t_{1\min}; t_2 - d_{\max}\} \Leftrightarrow x_{\text{médian}} \geq t_{1\min}$  et  $x_{\text{médian}} \geq t_2 - d_{\max}$

$t_2 - d_{\text{nom}} \geq t_{1\min} \Leftrightarrow t_2 \geq t_{1\min} + d_{\text{nom}}$

De même  $x_{\text{médian}} \leq x_2 = \text{Min} \{t_{1\max}; t_2 - d_{\min}\} \Rightarrow t_2 - d_{\text{nom}} \leq t_{1\max}$

D'où  $t_{1\min} + d_{\text{nom}} \leq t_2 \leq t_{1\max} + d_{\text{nom}}$

$$\begin{aligned} V(t_2) &= \text{Min} \{ \alpha t_1 + b + C(t_2 - t_1); t_1 \in [x_1; x_2] \} \\ &= \text{Min} \{ \text{Min} \{ \alpha t_1 + C(t_2 - t_1); t_1 \in [x_1; x_{\text{médian}}] \}; \text{Min} \{ \alpha t_1 + C(t_2 - t_1); t_1 \in [x_{\text{médian}}; x_2] \} \} + b \\ &= \text{Min} \{ \text{Min} \{ \alpha t_1 + f_1 ((t_2 - t_1) - d_{\text{nom}}) + C(d_{\text{nom}}); t_1 \in [x_1; x_{\text{médian}}] \}; \text{Min} \{ \alpha t_1 - f_2 ((t_2 - t_1) - d_{\text{nom}}) + C(d_{\text{nom}}); t_1 \in [x_{\text{médian}}; x_2] \} \} + b \\ &\text{car si } t_1 \leq x_{\text{médian}} = t_2 - d_{\text{nom}} \text{ alors } t_2 - t_1 \geq d_{\text{nom}} \\ &\text{et si } t_1 \geq x_{\text{médian}} = t_2 - d_{\text{nom}} \text{ alors } t_2 - t_1 \leq d_{\text{nom}} \end{aligned}$$



$$V(t_2) = \text{Min} \{ \text{Min} \{ (\alpha - f_1) t_1 ; t_1 \in [x_1; x_{\text{médian}}] \} + f_1 (t_2 - d_{\text{nom}}) ; \text{Min} \{ (\alpha + f_2) t_1 ; t_1 \in [x_{\text{médian}}; x_2] \} + f_2 (d_{\text{nom}} - t_2) \} + b + C(d_{\text{nom}})$$

- Premier sous cas :  $\alpha=f_1$

$$V(t_2) = \text{Min} \{ f_1 (t_2 - d_{\text{nom}}) ; (f_1 + f_2) x_{\text{médian}} + f_2 (d_{\text{nom}} - t_2) \} + b + C(d_{\text{nom}})$$

Avec  $x_{\text{médian}} = t_2 - d_{\text{nom}}$

$$V(t_2) = \text{Min} \{ f_1 (t_2 - d_{\text{nom}}) ; (f_1 + f_2) (t_2 - d_{\text{nom}}) + f_2 (d_{\text{nom}} - t_2) \} + b + C(d_{\text{nom}})$$

$$= \text{Min} \{ f_1 (t_2 - d_{\text{nom}}) ; f_1 (t_2 - d_{\text{nom}}) \} + b + C(d_{\text{nom}})$$

$$\boxed{V(t_2) = f_1 (t_2 - d_{\text{nom}}) + b + C(d_{\text{nom}})}$$

- Deuxième sous cas  $\alpha=-f_2$

$$V(t_2) = \text{Min} \{ -(f_1 + f_2) x_{\text{médian}} + f_1 (t_2 - d_{\text{nom}}) ; f_2 (d_{\text{nom}} - t_2) \} + b + C(d_{\text{nom}})$$

$$= \text{Min} \{ -(f_1 + f_2) (t_2 - d_{\text{nom}}) + f_1 (t_2 - d_{\text{nom}}) ; f_2 (d_{\text{nom}} - t_2) \} + b + C(d_{\text{nom}})$$

$$\boxed{V(t_2) = f_2 (d_{\text{nom}} - t_2) + b + C(d_{\text{nom}})}$$

### iii) Troisième cas $x_2 \leq x_{\text{médian}}$

$$x_2 = \text{Min} \{ t_{1\text{max}} ; t_2 - d_{\text{min}} \} \leq x_{\text{médian}} = t_2 - d_{\text{nom}}$$

Or  $d_{\text{min}} < d_{\text{nom}}$  d'où  $t_2 - d_{\text{min}} > t_2 - d_{\text{nom}}$

On en déduit que  $t_2 - d_{\text{nom}} \geq t_{1\text{max}}$  et que  $x_2 = t_{1\text{max}}$

$$t_2 - d_{\text{nom}} \geq t_{1\text{max}} \Rightarrow t_2 - d_{\text{nom}} \geq t_1 \Rightarrow t_2 - t_1 \geq d_{\text{nom}}$$

$$V(t_2) = \text{Min} \{ \alpha t_1 + f_1 ((t_2 - t_1) - d_{\text{nom}}) + C(d_{\text{nom}}) ; t_1 \in [x_1 ; t_{1\text{max}}] \} + b$$

$$V(t_2) = \text{Min} \{ (\alpha - f_1) t_1 ; t_1 \in [x_1 ; t_{1\text{max}}] \} + f_1 (t_2 - d_{\text{nom}}) + b + C(d_{\text{nom}})$$

- Premier sous cas :  $\alpha=f_1$

$$\boxed{V(t_2) = f_1 (t_2 - d_{\text{nom}}) + b + C(d_{\text{nom}})}$$

- Deuxième sous cas  $\alpha=-f_2$

$$-(f_1 + f_2) < 0$$

$$V(t_2) = -(f_1 + f_2) t_{1\text{max}} + f_1 (t_2 - d_{\text{nom}}) + b + C(d_{\text{nom}})$$

$$\boxed{V(t_2) = f_1 t_2 - (f_1 + f_2) t_{1\text{max}} - f_1 d_{\text{nom}} + b + C(d_{\text{nom}})}$$

### iv) Récapitulatif

En ce qui concerne les prédécesseurs optimaux, on remarque que dans plusieurs cas on a pris un minimum sur une fonction de pente nulle et l'on peut choisir le prédécesseur n'importe où parmi les prédécesseurs possibles : on construit ainsi le tableau des prédécesseurs optimaux. Cela traduit le fait qu'il y a plusieurs politiques optimales possibles. Sur le schéma récapitulatif on n'a gardé que l'expression correspondant à la politique choisissant quand c'est possible la vitesse nominale.

Intervalle	Cas $\alpha=f_1$	Cas $\alpha=f_2$
$t_{1\text{min}} + d_{\text{min}} \leq t_2 \leq t_{1\text{min}} + d_{\text{nom}}$	$t_{1\text{min}}$	$[\text{Max} \{t_{1\text{min}}, t_2 - d_{\text{nom}}\} ; t_2 - d_{\text{min}}]$
$t_{1\text{min}} + d_{\text{nom}} \leq t_2 \leq t_{1\text{max}} + d_{\text{nom}}$	$[\text{Max} \{t_{1\text{min}}, t_2 - d_{\text{max}}\} ; t_2 - d_{\text{nom}}]$	$[t_2 - d_{\text{nom}} ; \text{Min} \{t_{1\text{max}}, t_2 - d_{\text{nom}}\}]$

$$\begin{array}{lll}
 t_{1\max} + d_{\text{nom}} \leq t_2 & [t_2 - d_{\max} ; \text{Min} & t_{1\max} \\
 \leq t_{1\max} + d_{\max} & \{t_{1\max}, t_2 - d_{\text{nom}}\}] &
 \end{array}$$

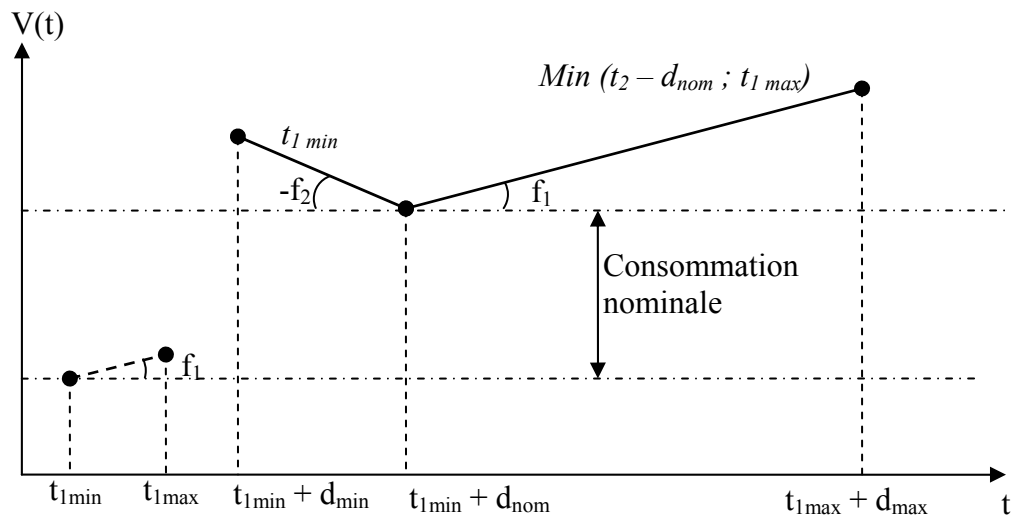
Tableau 7) Intervalle des prédécesseurs optimaux en fonction de  $t_2$

- Premier sous cas :  $\alpha = f_1$

● - - - ● Potentiel à la période  $s-1$

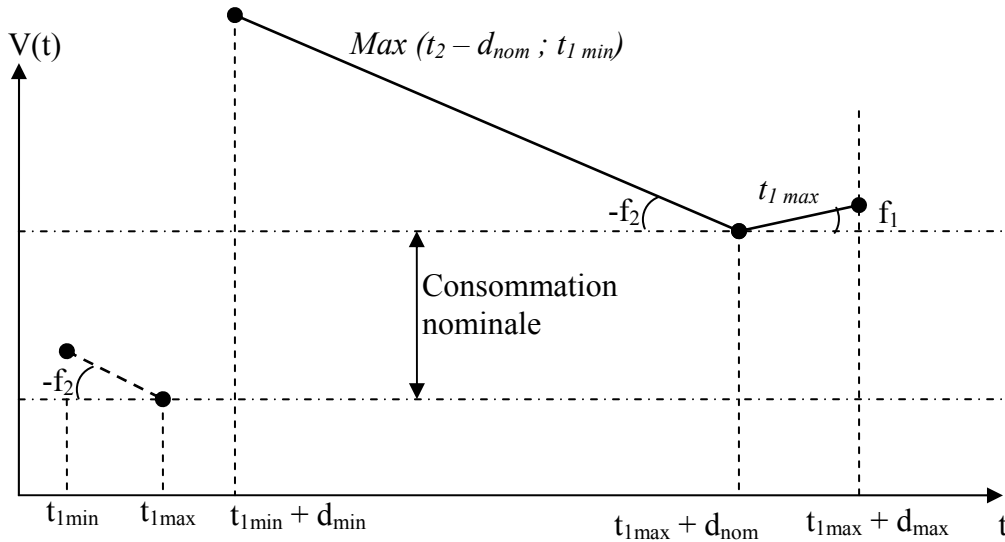
● — ● Potentiel à la période  $s$

$t$  : prédécesseur optimal



- Deuxième sous cas  $\alpha = -f_2$

- - - - ● Potentiel à la période s-1
  - — ● Potentiel à la période s
- $t$  : prédecesseur optimal



#### II.1.d) Description des piles de priorité utilisées dans l'algorithme de résolution du sous problème de tarification

Les piles de priorité (voir notamment [65], [58]) décrites ici sont celles qui ont été implémentées dans l'algorithme de résolution du sous problème de tarification. Leur rôle principal est de permettre de trouver à peu de frais le minimum d'une liste d'éléments en conservant cette liste sous une structure particulière. Si l'on note  $n$  le nombre d'éléments de notre liste alors chacune des trois opérations suivantes peut être exécutée en un temps proportionnel au logarithme de  $n$  : insertion d'un élément, suppression d'un élément, identification et suppression de l'élément minimum. L'identification du minimum ne demande qu'une seule opération mais c'est sa suppression de manière à garder la structure de pile qui demande  $\ln(n)$  opérations.

La pile de priorité utilisée dans notre application est appelée arbre binaire tassé à gauche partiellement ordonné.

- Arbre binaire

Un arbre binaire est un élément qui est soit vide, soit composé d'un sommet que l'on nomme « racine » et de deux arbres fils associés, « fils gauche » et « fils droit ». Les sommets dont les deux fils sont l'élément vide sont appelés « feuilles ».

- Arbre tassé à gauche

Un arbre est tassé si tous les sommets internes ont deux fils non vides, sauf éventuellement un sommet de l'avant dernier niveau. Toutes les feuilles sont tassées à gauche, c'est-à-dire qu'on remplit chaque niveau de gauche à droite.

- Arbre partiellement ordonné

Enfin un arbre est partiellement ordonné s'il vérifie la condition suivante : si b et c sont deux fils de a on a  $\text{clé}(a) \leq \text{clé}(b)$  et  $\text{clé}(a) \leq \text{clé}(c)$ . Un exemple d'arbre binaire partiellement ordonné tassé à gauche est présenté dans la Figure 47, les étiquettes des feuilles étant les clés.

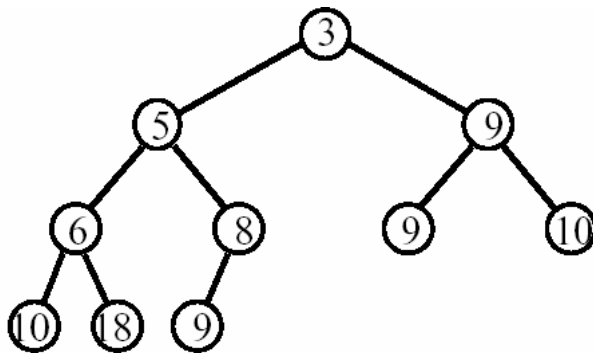


Figure 47) Exemple d'arbre binaire tassé à gauche partiellement ordonné

Pour conserver les propriétés de cette structure, on ne peut effectuer que certaines opérations comme par exemple :

- Supprimer le minimum.

On ne peut supprimer que l'élément situé dans la feuille la plus à droite du dernier niveau. Le minimum de cet arbre est obligatoirement la racine. Pour supprimer la racine on l'échange donc avec la dernière feuille de l'arbre. Ensuite on redescend l'élément que l'on a placé à la racine, en l'échangeant avec son fils qui a la clé la plus basse jusqu'à ce qu'on retrouve un arbre partiellement ordonné (voir Figure 48).

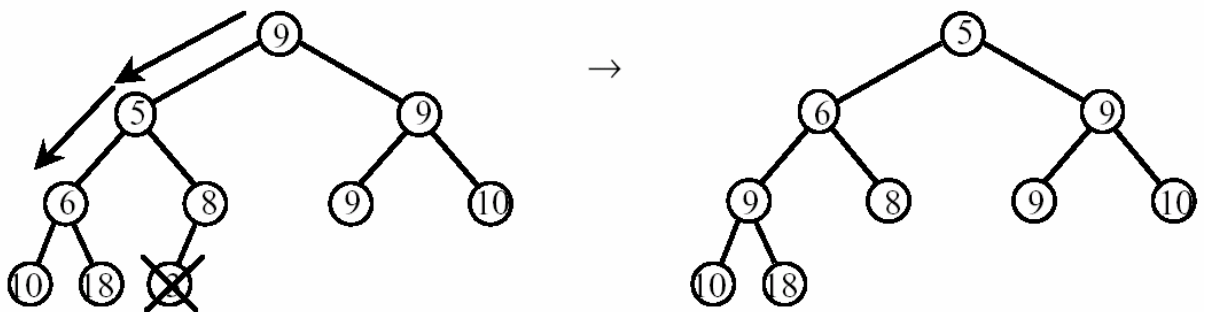


Figure 48) Suppression du minimum dans l'arbre de l'exemple

- Insérer un élément.

Pour insérer un élément, on crée une nouvelle feuille que l'on place en bas à droite, puis on l'échange avec son père tant que la condition « partiellement ordonnée » n'est pas satisfaite (voir Figure 49).

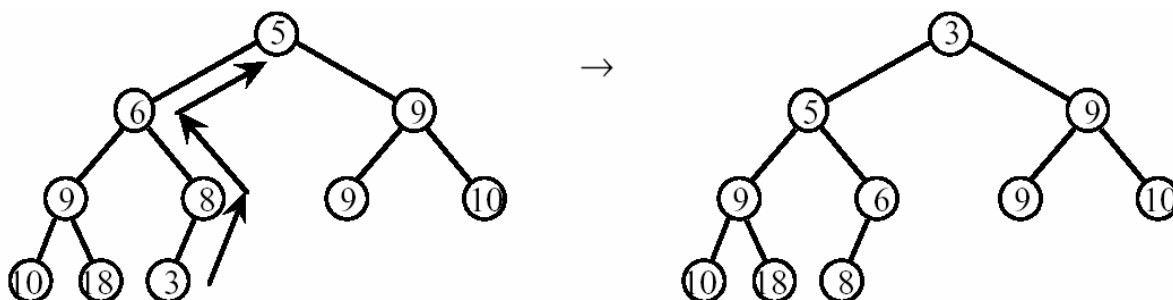


Figure 49) Insertion d'un élément de clé 3 dans l'arbre précédent

On peut stocker cet arbre sous forme de tableau en numérotant tous ses éléments en partant de 0 pour la racine puis en le parcourant en largeur d'abord de gauche à droite puis de haut en bas. Chaque élément noté  $i$  dans un tel tableau a ses fils numérotés  $2*i$  et  $2*i+1$ , son père est numéroté  $i/2$  (arrondi le cas échéant à l'entier inférieur).

La hauteur d'un arbre binaire tassé à gauche étant de l'ordre de  $\ln(n)$  où  $n$  est le nombre de sommets, les opérations décrites ci-dessus ont bien une complexité de  $O(\ln(n))$ .

## II.2. Lexique et acronymes

- ADP : ATFM Daily Plan
- AIRAC : Aeronautical Information, Regulation and Control
- ANM : ATFM Notification Message
- AOs: Aircraft Operators
- ATC: Air Traffic Control
- ATCCC: Air Traffic Control System Command Center
- ATFCM: Air Traffic Flow & Capacity Management
- ATFM: Air Traffic Flow Management
- ATO : Actual Time Over
- ATS : Air Traffic Service
- BADA : Base of Aircraft DATA
- CAMES : Co-operative ATM Measures for a European Single Sky
- CASA : Computer Assisted Slot Allocation
- CDM : Collaborative Decision Making
- CFMU: Central Flow Management Unit
- CPR's : Correlated Position Reports
- CSP Constrained Shortest Path problem
- CTO : Calculated Time Over
- CTOT : Calculated Take-Off Time
- EEC : Eurocontrol Experimental Center

- EOBT: Estimated Off-Block Time
- ETFMS : Enhanced Tactical Flow Management System
- ETO : Estimated Time Over
- ETOT : Estimated Take-off Time
- FAM : Future ATFM Measures
- FIFO First In First Out
- FIR Flight Information Region
- FMP : Flow Management Positions
- GHP: Ground Holding Problem
- HJB Hamilton Jacobi Bellman
- ICAO : International Civil Aviation Organisation
- IFPS : Initial Flight Plan Processing System
- IFR Instrument Flight Rules
- INRETS: Institut National de Recherche sur les Transports et leur Sécurité
- LICIT : Laboratoire Ingénierie Circulation Transports
- MAGHP Multi Airports Ground Holding Problem
- PFC : Predictive Functional Control
- PLNE: Programmation Linéaire en Nombres Entiers
- PRC: Performance Review Commission
- RCSP: Resource Constrained Shortest Paths problem
- RMP: Restricted Master Problem
- RVSM : Reduced Vertical Separation Minimum
- SAGHP: Single Airport Ground Holding Problem
- SAL: Slot Allocation List
- SAM : Slot Allocation Message
- SAP : Slot Amendment Procedure
- SEP Séparation et évaluation progressive
- SIP : Slot Improvement Proposal
- SIT : Slot Issue Time
- SPT: Shortest Path Tree problem
- SRM Slot Revision Message
- TB : Time Band
- TFMP: Traffic Flow Management Problem
- TFMRP: Traffic Flow Management Rerouting Problem
- TIS : Time to Insert into the Sequence

- TRS : Time to Remove from the Sequence
- VFR : Visual Flight Rules