HAL
open science

# Planification de trajectoires pour l'optimisation du trafic aérien

Cyril Allignol

## To cite this version:

Cyril Allignol. Planification de trajectoires pour l'optimisation du trafic aérien. Autre [cs.OH]. Institut National Polytechnique de Toulouse - INPT, 2011. Français. NNT : 2011INPT0104 . tel-01293618

**HAL Id: tel-01293618**

**https://theses.hal.science/tel-01293618v1**

Submitted on 25 Mar 2016

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

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

Université de Toulouse

# THÈSE

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

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

**Discipline ou spécialité :**

Sûreté de Logiciel et Calcul à Haute Performance (SLCHP)

---

**Présentée et soutenue par :**
Cyril ALLIGNOL

**le :** mardi 13 décembre 2011

**Titre :**

Planification de trajectoires pour l'optimisation du trafic aérien

---

**Ecole doctorale :**
Mathématiques Informatique Télécommunications (MITT)

**Unité de recherche :**
IRIT – Algorithmes Parallèles et Optimisation (APO)

**Directeur(s) de Thèse :**
Nicolas Durand, Directeur de thèse
Nicolas Barnier, Co-Directeur de thèse

**Rapporteurs :**

Pr Pierre Flener, Uppsala University
Pr Jin-Kao Hao, Université d'Angers

**Membre(s) du jury :**
Dr Nicolas Barnier, ÉNAC, Examinateur
Dr Nicolas Durand, DSNA/DTI, Directeur de thèse
Pr Éric Féron, Georgia Institute of Technology, Examinateur
Pr Pierre Flener, Uppsala University, Rapporteur
Pr Jin-Kao Hao, Université d'Angers, Rapporteur
Dr Xavier Prats, Universitat Politècnica de Catalunya, Examinateur

# Résumé

Le trafic aérien en Europe représente environ 30 000 vols quotidiens actuellement. Selon les prévisions de l'organisme Eurocontrol, ce trafic devrait croître de 70 % d'ici l'année 2020 pour atteindre 50 000 vols quotidiens. L'espace aérien, découpé en zones géographiques appelées secteurs de contrôle, atteindra bientôt son niveau de saturation vis-à-vis des méthodes actuelles de planification et de contrôle.

Afin d'augmenter la quantité de trafic que peut absorber le système, il est nécessaire de diminuer la charge de travail des contrôleurs aériens en les aidant dans leur tâche de séparation des avions. En se fondant sur les demandes de plans de vol des compagnies aériennes, nous proposons une méthode de planification des trajectoires en 4D permettant de présenter au contrôleur un trafic dont la plupart des con its auront été évités en avance.

Cette planification s'établit en deux étapes successives, ayant chacune un unique degré de liberté : une allocation de niveaux de vol permettant la résolution des con its en croisière puis une allocation d'heures de décollage permettant de résoudre les con its restants. Nous présentons des modèles pour ces deux problèmes d'optimisation fortement combinatoires, que nous résolvons en utilisant la programmation par contraintes ou les algorithmes évolutionnaires, ainsi que des techniques permettant de prendre en compte des incertitudes sur les heures de décollage ou le suivi de trajectoire.

Les simulations conduites sur l'espace aérien français mènent à des situations où tous les con its sont évités, avec des retards alloués de l'ordre d'une minute en moyenne (80 à 90 minutes pour le vol le plus retardé) et un écart par rapport à l'altitude optimale limité à un niveau de vol pour la quasi totalité des vols. La prise en compte d'incertitudes de manière statique dégrade fortement ces solutions peu robustes, mais nous proposons un modèle dynamique utilisant une fenêtre glissante susceptible de prendre en compte des incertitudes de quelques minutes avec un impact réduit sur le coût de l'allocation.

# Abstract

Air traffic in Europe represents about 30,000 ights each day and forecasts from Eurocontrol predict a growth of 70 % by 2020 (50,000 ights per day). The airspace, made up of numerous control sectors, will soon be saturated given the current planification and control methods.

In order to make the system able to cope with the predicted traffic growth, the air traffic controllers workload has to be reduced by automated systems that help them handle the aircraft separation task. Based on the traffic demand by airlines, this study proposes a new planning method for 4D trajectories that provides con ict-free traffic to the controller.

This planning method consists of two successive steps, each handling a unique ight parameter : a ight level allocation phase followed by a ground holding scheme. We present constraint programming models and an evolutionary algorithm to solve these large scale combinatorial optimization problems, as well as techniques for improving the robustness of the model by handling uncertainties of takeoff times and trajectory prediction.

Simulations carried out over the French airspace successfully solved all con icts, with a mean of one minute allocated delay (80 to 90 minutes for the most delayed ight) and a discrepancy from optimal altitude of one ight level for most of the ights. Handling uncertainties with a static method leads to a dramatic increase in the cost of the previous non-robust solutions. However, we propose a dynamic model to deal with this matter, based on a sliding time horizon, which is likely to be able to cope with a few minutes of uncertainty with reasonable impact on the cost of the solutions.

# Remerciements

C'est à Pascal Brisset que je dois d'être arrivé jusque là. Lorsque j'étais encore élève, il a su me transmettre sa passion pour la recherche et m'a entraîné (pas tout à fait à mon insu) dans un master. Sa maîtrise, son calme, son humour et sa capacité extraordinaire à rendre simples les problèmes les plus ardus faisaient de lui le collègue idéal. Merci Pascal, tu nous manques.

Si j'ai pu effectuer cette thèse ici à la DTI, c'est essentiellement grâce à Jean-Marc Alliot et Nicolas Durand. Après une affectation malheureuse, ils ont uvré sans relâche à m'extraire de mon « purgatoire » parisien. Leur aide et leur soutien m'ont permis de garder mon optimisme durant cette période.

Je souhaite également remercier chaleureusement Nicolas Barnier, qui, en plus d'encadrer mes travaux, partage mon bureau et doit donc me supporter au quotidien. Il a ancré dans mon esprit la notion de rigueur scientifique ainsi que tout ce que je sais de la programmation par contraintes, et y a sans aucun doute planté quelques graines musicales, littéraires et politiques...

Merci à Jin-Kao Hao et Pierre Flener qui ont accepté la tâche de rapporteur pour cette thèse malgré leurs emplois du temps très chargés, ainsi qu'aux autres membres du jury, Éric Féron et Xavier Prats, qui traversent océans et montagnes pour venir juger mon travail.

Ces quelques années n'auraient pas été si bonnes sans l'ambiance conviviale qui règne dans les locaux de l'« ex-LOG ». Merci à tous ceux qui ont participé à ce climat chaleureux : David Gianazza, Jean-Baptiste Gotteland, Pascal Lezaud, Estelle Malavolti, Nicolas (encore un) Saporito, Alexandre Gondran, Sonia Cafieri, François Huchet, sans oublier les nouveaux thésards de l'*open space* (bon courage à tous) : Richard, Mohammad, Brunilde, Laureline, Charlie, Loïc et Olga.

J'exprime toute ma reconnaissance aux personnels administratifs et techniques dont la disponibilité et la gentillesse ont rendu facile l'exercice au quotidien : Martine Labruyère, Marie-Antoinette Leblé et Nicolas « SOS » Courtel. Merci également à l'ÉNAC pour le financement de *utube*, le serveur de calcul que tout le monde nous envie (ou pas ?) et à Jean-Luc Vinot pour l'imagerie.

Enfin, je remercie ma famille et mes amis pour m'avoir supporté au cours de ces années, et tout particulièrement Sarah dont la présence, le soutien et la patience (presque) sans limite ont éclairé mes journées et m'ont permis de tenir dans les moments difficiles.

# Table des matières

**Introduction** **1**

**Unités de mesure** **5**

**I Contexte de l'étude** **7**

**1 Gestion du tra c aérien** **9**
1.1 Gestion de l'espace aérien . . . . . 9
1.1.1 Différents types d'espace . . . . . 10
1.1.2 Routes aériennes . . . . . 11
1.1.3 Sectorisation . . . . . 13
1.2 Régulations . . . . . 15
1.2.1 Système tactique d'aide à la régulation . . . . . 15
1.2.2 Algorithme CASA pour l'aide à la régulation . . . . . 15
1.2.3 Régulations et retards . . . . . 17
1.2.4 Coûts des retards . . . . . 17
1.3 Contrôle du trafic aérien . . . . . 18
1.3.1 Différents types de contrôle . . . . . 18
1.3.2 Méthodes du contrôle . . . . . 19
1.4 Limites du modèle actuel . . . . . 21
1.4.1 Évolutions du trafic aérien en Europe . . . . . 21
1.4.2 Saturation de l'espace aérien . . . . . 23

**2 Évolution des systèmes de contrôle** **27**
2.1 Approches opérationnelles . . . . . 27
2.1.1 Nouveaux concepts ATM . . . . . 28
2.1.2 Tentatives d'automatisation . . . . . 28
2.2 Approches théoriques . . . . . 31
2.2.1 Avions autonomes . . . . . 31
2.2.2 Résolution centralisée . . . . . 32
2.2.3 Réseau de routes . . . . . 34
2.2.4 Planification et régulation . . . . . 35

**II Plani cation de trajectoires 4D : modèles** **39**

**3 Détection de con it** **41**
3.1 Simulation de trafic . . . . . 41
3.1.1 Principe général . . . . . 42
3.1.2 Données d'entrée . . . . . 42
3.1.3 Modèle de vol . . . . . 43
3.1.4 In uence du pas de temps de simulation . . . . . 44
3.2 Détection de con it . . . . . 45
3.2.1 Zone de con it . . . . . 46
3.2.2 Détection de con its 4D . . . . . 47
3.2.3 Boîtes englobantes . . . . . 52

**4 Allocation de créneaux de décollage** **55**
4.1 Hypothèses sur la précision des trajectoires . . . . . 55
4.2 Variables . . . . . 56
4.3 Traitement des données . . . . . 57
4.3.1 Filtrage des trajectoires . . . . . 57
4.3.2 Filtrage des con its . . . . . 58
4.3.3 Filtrage des vols . . . . . 59
4.3.4 Découpage du problème . . . . . 59
4.4 Optimisation . . . . . 60
4.4.1 Équité entre utilisateurs de l'espace . . . . . 60
4.4.2 Performance globale du système . . . . . 61
4.5 In uence du retard maximal autorisé . . . . . 61
4.6 Modélisation des incertitudes . . . . . 62
4.6.1 Prise en compte des incertitudes sur les heures de décollage . . . . . 62
4.6.2 Résolution dynamique avec fenêtre glissante . . . . . 62

**5 Allocation de niveaux de vol** **69**
5.1 Coût d'un changement de niveau de vol . . . . . 70
5.2 Modèle . . . . . 71
5.2.1 Variables . . . . . 71
5.2.2 Filtrage . . . . . 72
5.3 Allocation par ux . . . . . 72
5.3.1 Agrégation des vols en ux . . . . . 73
5.3.2 Séparation des ux en altitude . . . . . 74
5.3.3 Prise en compte de l'aspect temporel . . . . . 74
5.4 Allocation par vol . . . . . 75
5.4.1 Con its 2D + temps . . . . . 75
5.4.2 Raffinement du modèle en ux . . . . . 75
5.4.3 Séparation des vols en altitude . . . . . 76
5.4.4 Relaxation temporelle . . . . . 77

5.5 Optimisation . . . . . 77

**III Méthodes de résolution 79**

**6 Résolution par algorithmes évolutionnaires 81**
6.1 Principe des algorithmes génétiques . . . . . 81
6.1.1 Algorithme génétique élémentaire . . . . . 82
6.1.2 Raffinements . . . . . 88
6.2 Allocation de créneaux de décollage par AG . . . . . 92
6.2.1 Variables et structures de données . . . . . 92
6.2.2 Fonction d'évaluation . . . . . 92
6.2.3 Population initiale . . . . . 93
6.2.4 Opérateur de mutation . . . . . 93
6.2.5 Opérateur de croisement . . . . . 94
6.2.6 Paramètres de l'algorithme génétique . . . . . 95

**7 Résolution en programmation par contraintes 97**
7.1 Principes de la programmation par contraintes . . . . . 97
7.1.1 Principes généraux . . . . . 98
7.1.2 Problèmes de satisfaction de contraintes . . . . . 99
7.1.3 Résolution de CSP . . . . . 101
7.2 Allocation de créneaux de décollage en PPC . . . . . 107
7.2.1 Expression du CSP . . . . . 107
7.2.2 Optimisation . . . . . 109
7.2.3 Stratégies de recherche . . . . . 109
7.2.4 Contraintes supplémentaires . . . . . 111
7.3 Allocation de niveaux de vol en PPC . . . . . 112
7.3.1 Expression du CSP . . . . . 112
7.3.2 Optimisation . . . . . 113
7.3.3 Résolution . . . . . 114

**8 Résultats et analyses 115**
8.1 Phase d'allocation de niveaux de vol . . . . . 115
8.1.1 Allocation par ux de vols . . . . . 115
8.1.2 Allocation par vol . . . . . 117
8.2 Phase d'allocation de créneaux de décollage . . . . . 120
8.2.1 Évaluation de l'allocation de créneaux . . . . . 121
8.2.2 Comparaison des performances en routes standards et en routes directes 124
8.2.3 Modification de la norme de séparation . . . . . 126
8.2.4 In uence de l'allocation de niveaux de vol sur l'allocation de créneaux 126
8.2.5 Prise en compte des incertitudes . . . . . 131

**Conclusion et perspectives** **135**

**Glossaire** **139**

**Bibliographie** **141**

# Table des  gures


1.1 Extrait d'une carte d'un réseau de routes aériennes dans le sud de la France. 11
1.2 Extrait d'un formulaire de dépôt de plan de vol. . . . . 12
1.3 Secteurs de contrôle de l'espace aérien français. . . . . 13
1.4 Opérations de regroupement/dégroupement de secteurs de contrôle. . . . . 14
1.5 Les pays membres de la zone CEAC en 2011. . . . . 16
1.6 Poste de contrôle dans un centre en-route. . . . . 20
1.7 Un *strip* de contrôle annoté par un contrôleur. . . . . 21
1.8 Norme de séparation pour la phase en-route. . . . . 21
1.9 Visualisation des données radar. . . . . 22
1.10 Prévisions de trafic en Europe à l'horizon 2030. . . . . 23

2.1 Man  uvre de type « point tournant ». . . . . 32

3.1 Extrait d'un fichier de plans de vols COURAGE. . . . . 43
3.2 In  uence du pas de temps de simulation sur la détection de con  it. . . . . 44
3.3 Zone de con  it entre deux avions à vitesses constantes. . . . . 47
3.4 Taille de la zone de con  it en fonction de l'angle de convergence des trajectoires et des vitesses relatives. . . . . 48
3.5 Détection de points con  ictuels. . . . . 49
3.6 Trois con  its potentiels entre deux vols. . . . . 51
3.7 Filtrage par boîtes englobantes. . . . . 53
3.8 Comparaison des temps de détection des con  its avec et sans utilisation de boîtes englobantes. . . . . 53

4.1 In  uence du filtrage en niveau de vol sur la taille du problème. . . . . 58
4.2 Découpage en clusters. . . . . 59
4.3 Un vol en con  it avec plusieurs autres. . . . . 60
4.4 Résolution par fenêtre glissante. . . . . 63
4.5 Densité de probabilité de l'incertitude $\varepsilon$ sur les heures de décollage. . . . . 65

5.1 Répartition des différents types de con  its. . . . . 70
5.2 Plus longue phase de croisière possible pour un vol. . . . . 72
5.3 Distribution des retards alloués pour une journée de trafic. . . . . 77

6.1 Limites du codage binaire des individus. . . . . 83
6.2 Opérateur de croisement pour un codage discret. . . . . 87
6.3 Exemples de cas où la sélection risque d'être inefficace. . . . . 89
6.4 Effet d'un *scaling* exponentiel de facteur $k = 4$. . . . . 90
6.5 Illustration des effets du *sharing*. . . . . 91
6.6 Opérateur de croisement. . . . . 95

7.1 Parcours d'un arbre de recherche par algorithme de *backtrack*. . . . . 102
7.2 *Backtrack* et propagation de contraintes pour la recherche de solutions. . . 106

8.1 Répartition des écarts par rapport au RFL après allocation avec agrégation en ux. . . . . 116
8.2 Nombre de con its restants en fonction de $\Delta_{\max}$ pour l'allocation avec agrégation en ux. . . . . 117
8.3 Répartition des écarts par rapport au RFL après allocation par vol. . . . . 118
8.4 Nombre de con its restants en fonction de $\Delta_{\max}$ le 14/08/2008 pour l'allocation par vol. . . . . 119
8.5 Nombre de con its restants avant (en rouge) et après (en vert) l'allocation par vol. . . . . 119
8.6 Prise en compte de l'aspect temporel pour l'allocation de niveaux de vol : coût ($\text{cost}_{FL}$, échelle de gauche) et écart maximal au RFL ($\Delta_{\max}$, échelles de droite) en fonction de la valeur de $\delta_{\max}$. . . . . 120
8.7 Résultats de l'allocation de créneaux de décollage en PPC. . . . . 121
8.8 Temps de calcul de l'allocation de créneaux de décollage en fonction du nombre de vols. . . . . 123
8.9 Comparaison des résultats de l'allocation de créneaux de décollage en routes standards et en routes directes. . . . . 125
8.10 In uence de la valeur de la norme de séparation horizontale sur le coût de l'allocation de créneaux de décollage. . . . . 127
8.11 Comparaison des résultats de l'allocation de créneaux de décollage avec et sans allocation préalable de niveaux de vol. . . . . 128
8.12 Comparaison des résultats de l'allocation de créneaux de décollage en fonction de la valeur de $\Delta_{\max}$. . . . . 129
8.13 Comparaison des résultats de l'allocation de créneaux de décollage avec et sans allocation de niveaux de vol, en routes standards et en routes directes. 130
8.14 Distribution des retards pour une journée de trafic à l'échelle européenne. . 131
8.15 Nombre de con its détectés par CATS en fonction de l'incertitude sur les heures de décollage. . . . . 132
8.16 Pourcentage de résolutions invalides en fonction de l'extension des con its et de l'incertitude sur les heures de décollage. . . . . 133
8.17 Inlfuence de l'extension des con its sur le coût de la régulation. . . . . 133

# Liste des tableaux

1.1 Les différentes classes d'espace aérien. . . . . 10

3.1 Données de performances BADA tabulées pour l'Airbus A380. . . . . 43

5.1 Réduction du rayon d'action en fonction de la différence d'altitude par rapport à l'altitude optimale. . . . . 70

8.1 Taille des instances et coût de l'allocation de niveaux de vol avec agrégation en ux. . . . . 116

8.2 Comparaison des résultats du modèle CSP et de l'algorithme génétique. . . 123

# Liste des algorithmes


4.1 Résolution par fenêtre glissante . . . . . 64
4.2 Résolution par fenêtre glissante avec prise en compte des incertitudes . . . 66
6.1 Algorithme génétique . . . . . 82
6.2 évaluation$(P, F)$ . . . . . 84
6.3 sélection$(P, e)$ . . . . . 86
6.4 croisement$(P)$ . . . . . 86
6.5 mutation$(P)$ . . . . . 88
6.6 Opération de mutation . . . . . 94
7.1 Backtrack : $BT(\mathcal{N}, \mathcal{I})$ . . . . . 102
7.2 révise$(x, y)$ . . . . . 104
7.3 AC-3 . . . . . 105

# Introduction

Le trafic aérien est un secteur en forte croissance depuis plusieurs décennies, malgré les quelques crises qui l'ont impacté (attentats terroristes en 2001, virus SRAS en 2003, crise économique en 2008 notamment). Face à cette évolution, les services du contrôle ont apporté des solutions humaines et techniques afin d'assurer un niveau de sécurité toujours plus élevé : recrutements, outils de visualisation, méthodes de travail... Depuis quelques années, cependant, on observe que ces améliorations ne permettent plus d'accroître la capacité de l'espace aérien de manière significative. Dans ce contexte, il est nécessaire, afin de permettre la croissance future du trafic aérien, de remettre en cause la structure de l'espace aérien ainsi que les méthodes actuelles de contrôle et de régulation du trafic.

Ainsi, un certain nombre d'études ont testé des méthodes et concepts nouveaux pour tenter d'augmenter la quantité de trafic que le système peut absorber et ce, à tous les étages du système : optimisation du réseau de routes aériennes, abandon du découpage de l'espace en secteurs de contrôle, automatisation complète du contrôle, délégation aux avions de la détection et de la résolution de con its...

En Europe, des mesures de régulation sont mises en uvre par un organisme appelé CFMU (Central Flow Management Unit), dont le principe est d'attribuer des créneaux de décollage aux avions qui traversent des secteurs de contrôle surchargés. Ces régulations sont fondées sur une mesure de capacité macroscopique qui représente mal la charge de travail des contrôleurs. Afin de contourner certaines limites du modèle actuel, nous proposons dans cette étude une méthode de planification et de régulation qui utilise la structure microscopique du trafic (les croisements entre trajectoires) pour représenter la charge de travail.

## Contributions

Dans un premier temps, nous proposons une nouvelle méthode de régulation qui consiste à détecter les « con its 4D » statiquement, avant le décollage, et à les éviter par l'attribution d'une heure de décollage à chaque vol. Un retard au décollage représente un coût pour une compagnie aérienne et il faudra donc minimiser leur quantité lors de la phase d'allocation. Un modèle original de con it 4D, adapté à ce type de régulation, est présenté. Cette mesure de planification repose sur l'hypothèse d'un suivi parfait des trajectoires 4D par les avions.

Afin de permettre une réduction de la quantité de retard attribuée lors de cette phase, nous proposons dans un second temps de combiner cette première mesure de régulation

avec une phase préalable d'allocation de niveaux de vol. Celle-ci consiste à séparer les vols en altitude durant la phase de croisière, afin de réduire en amont la complexité du trafic. Cependant, chaque type d'avion a une altitude de vol optimale dépendante du type de vol réalisé, qui correspond à une consommation de carburant minimale. Nous cherchons donc à optimiser la phase d'allocation de niveaux de vol suivant ce critère.

Ces deux problèmes sont des problèmes d'optimisation fortement combinatoires de grande dimension, que nous proposons de résoudre en utilisant deux méthodes d'optimisation : une métaheuristique, un algorithme génétique muni d'opérateurs spécifiques (adaptés à la fonction d'évaluation) et une méthode complète, la programmation par contraintes. La méthode sera validée sur des données de trafic simulé à partir de plans de vol réels.

L'hypothèse de précision sur le suivi des trajectoires n'est pas réaliste dans le cadre actuel. Il est donc important de considérer les incertitudes (notamment sur les heures de décollage et sur les vitesses d'évolution des avions) susceptibles de perturber les prévisions de con its. Nous proposons deux méthodes, l'une statique, l'autre dynamique avec horizon temporel glissant, permettant de prendre en compte ce type d'incertitudes.

## Organisation du document

Cette thèse est constituée de trois parties. La première partie place notre étude dans son contexte en présentant dans le chapitre 1 les méthodes actuelles de régulation et de contrôle du trafic aérien. Puis le chapitre 2 donne un aperçu des différentes études menées dans le but d'améliorer la gestion de l'espace aérien et d'augmenter la capacité du système de contrôle ainsi que de leurs limites.

Dans ce contexte, la deuxième partie présente la méthode de planification mise en uvre. Le chapitre 3 décrit le simulateur de trafic CATS ayant servi à la validation de nos travaux, ainsi qu'une méthode de détection et de modélisation fine des con its 4D adaptée à notre problème. Les chapitres suivants formulent les modèles pour les deux phases de la régulation : le chapitre 4 propose une technique d'allocation de créneaux de décollage sans con it ainsi que des techniques permettant la prise en compte d'incertitudes sur les heures de décollage, et le chapitre 5 une méthode d'allocation de niveaux de vol permettant la séparation des avions en altitude lors de la phase de croisière.

Les méthodes utilisées pour la résolution de ces deux problèmes d'optimisation combinatoires sont présentées dans la troisième partie. Le chapitre 6 propose un algorithme génétique pour la résolution du problème d'allocation de créneaux de décollage. Les résultats de cet algorithme sont intéressants du point de vue de la fonction d'évaluation mais les temps de calcul ne sont pas compatibles avec une approche opérationnelle de la méthode de planification. Nous avons donc choisi de tester un autre paradigme, la programmation par contraintes, dont l'application à nos deux problèmes d'allocation est décrite dans le chapitre 7. Le chapitre 8 montre les résultats obtenus suite à l'implémentation de ces modèles.

Nous exposerons enfin dans la conclusion quelques perspectives d'amélioration de notre modèle de régulation du trafic, notamment en ce qui concerne la robustesse et le réalisme

opérationnel de celui-ci.

Les travaux effectués dans le cadre de cette thèse ont fait l'objet de plusieurs publications et communications [Allignol 09, Barnier 09, Allignol 10, Durand 10, Allignol 11, Barnier 11] et [Barnier 12] (à paraître).

# Unités de mesure

Dans le domaine de l'aéronautique, les unités de mesure sont essentiellement issues du système de mesure anglo-saxon. Nous présentons ici les principales mesures que nous utiliserons tout au long de ce document, ainsi que leur correspondance avec les unités du Système international.

La navigation aéronautique présente de grandes similitudes avec la navigation maritime, principalement le fait de naviguer sur une sphère. Elle a notamment hérité d'une unité de mesure de distance : le *mille nautique* – Nautical Mile (NM). Le mille nautique représente la distance correspondant à une différence d'une minute d'angle sur un grand cercle [1] terrestre. La correspondance est la suivante : 1 NM = 1 852 m.

Les distances verticales sont mesurées à l'aide d'un altimètre ; l'altitude est donc connue par rapport à une surface isobare. Le calage de l'altimètre à partir d'une pression atmosphérique de référence permet de déduire une altitude (*i.e.* hauteur par rapport au niveau de la mer) ou une hauteur par rapport à un aéroport. L'unité de mesure utilisée est le *pied* (ft) : 1 ft = $0,3048$ m. Lorsque les avions sont proches du sol, il est important de connaître précisément leur altitude, et il est donc d'usage de caler l'altimètre avec l'altitude-pression du terrain de décollage ou d'atterrissage. Cependant, lors du parcours d'une grande distance, les pressions atmosphériques au sol sur les différents aéroports peuvent varier de manière significative. On risque alors de trouver dans une même portion de l'espace aérien des avions dont les altimètres ne sont pas calés sur la même valeur, et deux avions pensant évoluer à des altitudes différentes peuvent en fait se trouver à la même altitude. Pour remédier à ce problème, les avions évoluant au-dessus d'une certaine altitude appelée *altitude de transition* (en général 5 000 ft) doivent caler leur altimètre de manière à mesurer les altitudes par rapport à la surface isobare 1 013,25 hPa. On utilise alors le *niveau de vol* – Flight Level (FL) – pour mesurer les altitudes : un niveau de vol correspond à 100 ft mesurés à partir de la surface isobare 1 013,25 hPa. Les niveaux de vol utilisables sont FL 50, FL 60, FL 70... (c'est-à-dire 5 000 ft, 6 000 ft, 7 000 ft...).

Les unités de mesure de vitesse sont directement issues des unités de mesure de distance décrites ci-dessus :

- le *n ud* (kt) mesure les vitesses horizontales, 1 kt = 1 NM/h ;
- le *pied par minute* (ft/min) mesure les vitesses verticales.

1. Un grand cercle est un cercle tracé sur la surface d'une sphère et qui a le même diamètre qu'elle. Les arcs de grand cercle sont les plus courts chemins entre deux points sur une sphère.

# Première partie

# Contexte de l'étude

# Chapitre 1

# Gestion du tra c aérien

Le trafic aérien a augmenté de manière continue depuis plus de 40 ans. Seuls quelques rares événements ont interrompu cette croissance durant ces dernières années, notamment les attentats du 11 septembre 2001 aux États-Unis et la crise économique de 2008. Afin d'écouler ce trafic de manière sûre et uide, l'agence Eurocontrol, qui regroupe aujourd'hui 38 États européens, coordonne les différents services de la navigation aérienne.

La gestion du trafic aérien en Europe est organisée en plusieurs niveaux de « filtres » imbriqués, qui opèrent avec des horizons de prévision différents. Le filtre stratégique concerne l'organisation à long terme du trafic. Il regroupe toutes les activités liées à la gestion de l'espace aérien (section 1.1), notamment le découpage de l'espace, la géométrie des routes aériennes et la définition des procédures pour les phases de montée et de descente. Le filtre pré-tactique a un horizon de quelques jours à quelques heures et correspond à la mise en uvre de mesures de régulation (section 1.2), afin d'adapter la charge de trafic à la capacité de l'espace aérien telle que définie lors de la phase stratégique. Le filtre tactique correspond à la gestion du trafic effectuée par les contrôleurs aériens (section 1.3) et regroupe les tâches de surveillance du trafic, de coordination et de résolution des con its aériens. Enfin, le filtre d'urgence est activé en cas de défaillance de tous les filtres précédents. Si une collision est imminente (entre deux avions ou entre un avion et un obstacle), des systèmes au sol et embarqués alertent le contrôleur ou proposent des man uvres au pilote afin d'éviter l'accident. Ce chapitre détaille ces différents niveaux de filtre.

## 1.1 Gestion de l'espace aérien

Les avions souhaitant évoluer dans l'espace aérien européen peuvent choisir de le faire selon deux modes de vol : le vol à vue – Visual Flight Rules (VFR) – ou le vol aux instruments – Instrument Flight Rules (IFR). Le vol VFR est le plus libre des deux et est particulièrement adapté aux vols de tourisme. Il se base sur le principe « *voir et éviter* » et nécessite donc certaines conditions de visibilité. Le vol IFR repose quant à lui sur les équipements de radionavigation de l'avion. C'est le mode utilisé pour les vols commerciaux en Europe. Les différences dans les performances et les missions des vols VFR et IFR ont

conduit à un partage de l'espace aérien afin de réduire le risque de collision. En particulier, certaines parties de l'espace aérien ne sont pas accessibles aux vols VFR. Cette section présente les différentes catégories d'espace, ainsi que l'organisation du trafic aérien au sein de ces zones.

### 1.1.1 Di érents types d'espace

On distingue principalement deux types d'espaces aériens :

- les espaces aériens *contrôlés*, au sein desquels la séparation entre avions est assurée par un organisme de contrôle (dont les procédures sont détaillées dans la section 1.3) ;
- les espaces aériens *non contrôlés* pour lesquels seuls des services d'information et d'alerte sont fournis.

De manière plus précise, l'Organisation de l'Aviation Civile Internationale (OACI) a défini sept *classes* d'espace aérien en fonction des services qui y sont rendus et des avions autorisés à y pénétrer. Les caractéristiques principales de ces classes d'espace sont résumées dans le tableau 1.1.

Table 1.1 – Les différentes classes d'espace aérien.

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="3">Espacement</th><th rowspan="2">Contact radio obligatoire</th></tr>
<tr><th>IFR-IFR</th><th>IFR-VFR</th><th>VFR-VFR</th></tr>
<tr><td rowspan="4">Contrôlés</td><td>Classe A</td><td rowspan="4">Oui</td><td colspan="2">VFR non autorisé</td><td rowspan="4">Oui</td></tr>
<tr><td>Classe B</td><td colspan="2"></td></tr>
<tr><td>Classe C</td><td></td><td></td></tr>
<tr><td>Classes D et E</td><td colspan="2">Non</td></tr>
<tr><td>Non contrôlés</td><td>Classes F et G</td><td colspan="3">Non</td><td>Non</td></tr>
</table>

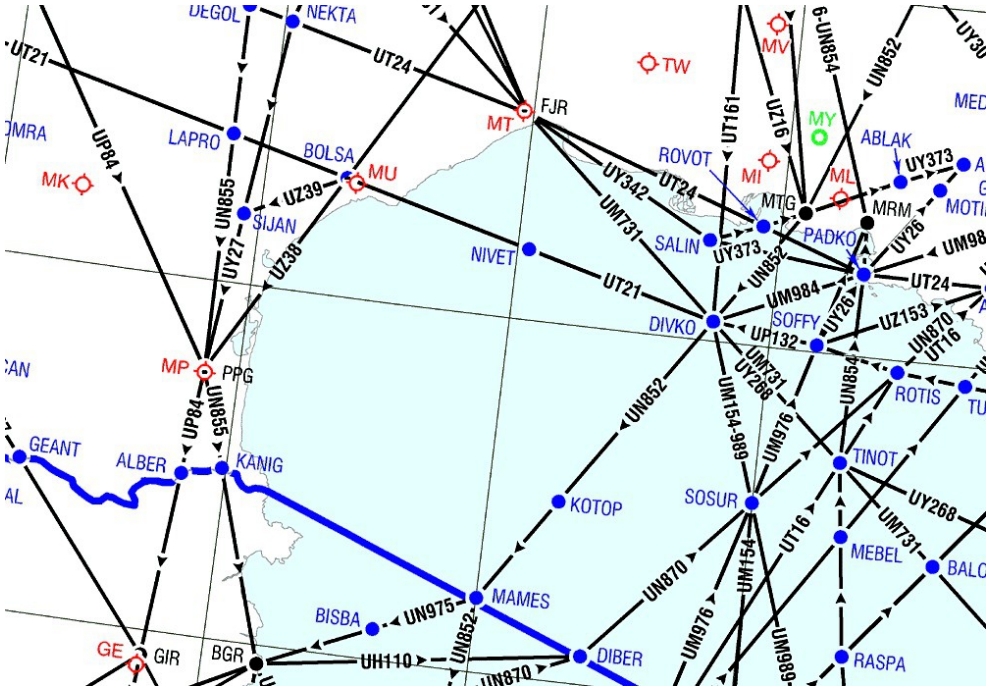
Chacune de ces classes d'espace a un usage particulier :

- Classe A : au-dessus du FL 180 et autour des plus gros aéroports (*e.g.* Roissy-Charles de Gaulle).
- Classes B, C et D : autour des aéroports. Plus l'aéroport est important, plus le niveau de sécurité est grand (séparation assurée entre tous les avions pour les gros aéroports, seulement entre les vols IFR pour les plus petits).
- Classe E : au-dessous du FL 180, en dehors des zones aéroportuaires.
- Classes F et G : pour le reste de l'espace aérien.

Il existe en outre des zones militaires, c'est-à-dire des espaces aériens dont l'usage est fortement réglementé, voire interdit à la circulation aérienne générale [1].

Les aéronefs évoluent au sein de ces différents espaces en suivant des *routes aériennes* prédéterminées dont nous allons détailler la construction dans la section suivante.

1. On oppose la circulation aérienne *générale* (CAG) à la circulation aérienne *militaire* (CAM).




Source : Eurocontrol

FIGURE 1.1 – Extrait d'une carte du réseau de routes aériennes dans le sud de la France. Les points noirs sont des points de report réels (moyens de radionavigation), les points bleus sont des points de report fictifs.

### 1.1.2 Routes aériennes

Une *route aérienne* est une succession de segments dans le plan horizontal, dont les extrémités – les points de report ou *waypoints* – sont situées au-dessus des moyens de radionavigation au sol (*e.g.* balises VOR [2]). Du fait de l'emplacement de ces moyens de radionavigation, qui nécessitent un terrain particulier pour leur installation et sont en nombre limité, le suivi de ces routes est contraignant et coûteux en termes de temps de parcours et de carburant consommé. La précision des moyens de radionavigation embarqués récents a permis la création d'un nouveau type de route : les points de report ne sont plus les balises elles-mêmes, mais des points fictifs qui peuvent être déduits des valeurs de relèvement [3] des balises voisines. Ce mode de navigation est appelé *Area Navigation (RNAV)*. La figure 1.1 montre une portion du réseau de routes aériennes tel qu'il existe actuellement et illustre les deux modes de navigation évoqués ci-dessus.

Le mode RNAV offre des possibilités très intéressantes en ce qui concerne la gestion du réseau de routes. En effet, puisque les points de report sont dématérialisés, il est possible de placer ceux-ci de telle sorte que les trajets soient raccourcis de manière significative. Il devient également possible de mettre en place des *routes directes*, c'est-à-dire des routes

2. VHF Omnidirectional Range.
3. Le relèvement d'un point B par rapport à un point A est l'angle entre le Nord et le segment [BA].

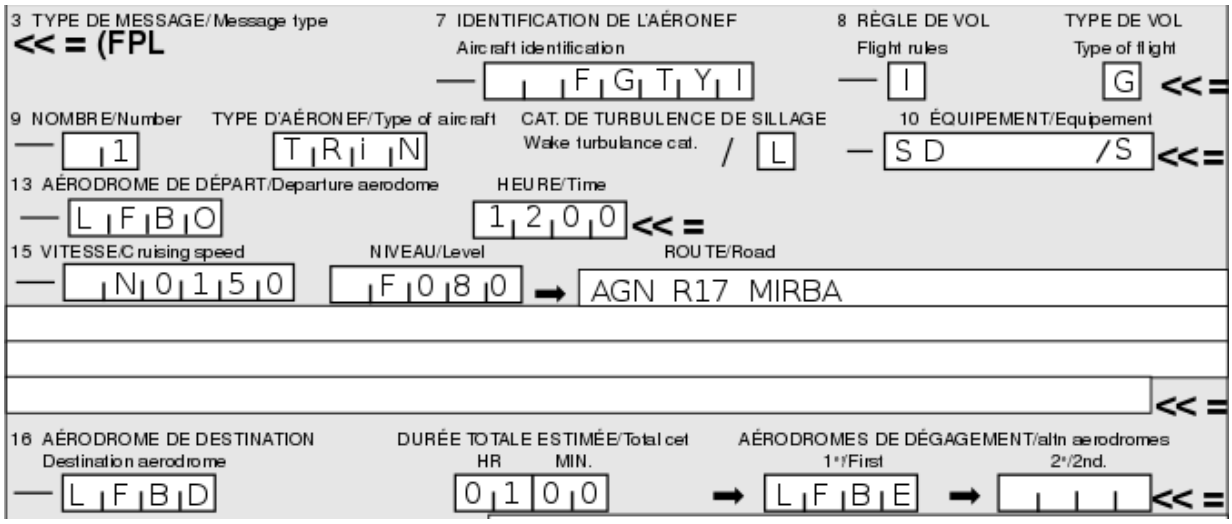



FIGURE 1.2 – Extrait d'un formulaire de dépôt de plan de vol pour un vol entre Toulouse (LFBO) et Bordeaux (LFBD). La route est constituée des points de report AGN (Agen), R17 et MIRBA et la croisière sera effectuée au FL 80 à une vitesse de 150 kt.

optimales entre deux points (*e.g.* entre aéroport de départ et aéroport d'arrivée ou, de manière plus réaliste, entre point d'entrée et point de sortie d'une zone contrôlée donnée).

Ce concept de route directe est en essai opérationnel depuis le mois de mars 2011 au sein de l'espace aérien contrôlé par le centre de Maastricht, dans le cadre du projet Free Route Airspace Maastricht (FRAM) [Eurocontrol 11]. Dans ce contexte, 142 routes directes sont utilisables pour traverser l'espace concerné. Eurocontrol annonce un gain de 624 000 NM sur une année, en n'autorisant ces routes que lors des périodes peu fréquentées (en semaine de 00h00 à 08h00 et le week-end). Il est prévu que ces routes soient ouvertes en continu à partir de 2013, ce qui implique la mise en place de nouvelles procédures de contrôle dans la mesure où, du point de vue du contrôleur aérien, le trafic en routes directes est plus difficile à écouler qu'en routes standards (notamment à cause des trajectoires dont l'angle de convergence est faible ainsi que du plus grand nombre de points d'intersection entre les différentes routes).

Quel que soit le mode de navigation choisi, il est nécessaire pour planifier un vol IFR de déposer un *plan de vol*, c'est-à-dire d'informer les organismes de contrôle de la route que l'on souhaite emprunter. Le plan de vol consiste notamment en une série de points de report, ainsi qu'une heure prévue de décollage et un niveau de vol de croisière souhaité (voir figure 1.2).

Le dépôt de ce plan de vol doit se faire au minimum trois heures avant le décollage. Ce délai permet aux autorités de contrôle et de régulation de connaître les intentions de chaque avion, et ainsi de déterminer les volumes de l'espace aérien qu'ils prévoient de traverser. Les plans de vol pour les vols réguliers assurés par des compagnies aériennes peuvent en outre être déposés automatiquement, le dépôt pouvant être fait un ou deux jours avant.

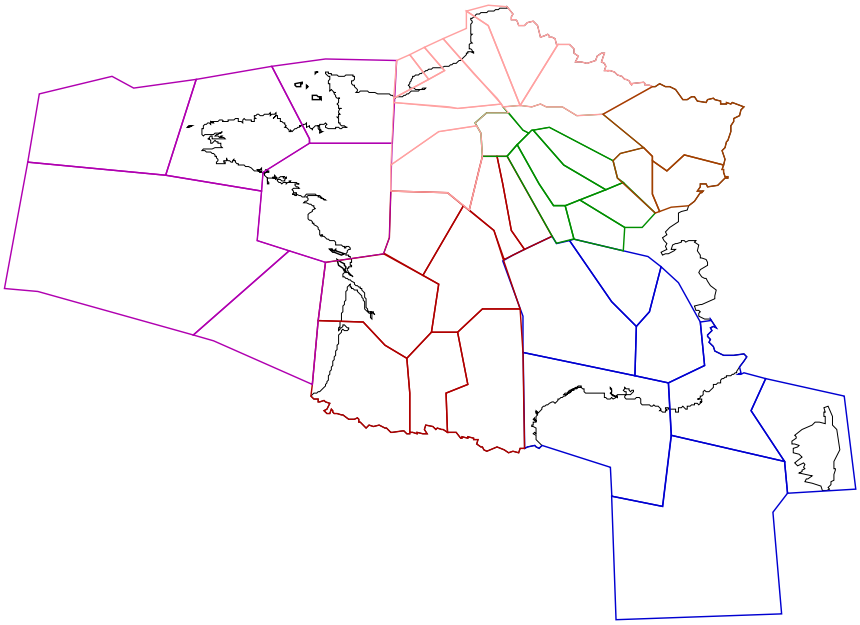


FIGURE 1.3 – Secteurs de contrôle de l'espace aérien français. Les secteurs ayant la même couleur dépendent du même centre de contrôle.

### 1.1.3 Sectorisation

Étant donné le volume de trafic sur un territoire comme celui de la France, il n'est pas envisageable pour un opérateur humain de contrôler l'ensemble des avions sur un unique poste de contrôle. L'espace aérien est donc découpé en entités fonctionnelles, ce qui permet de répartir la charge de travail sur plusieurs postes mais nécessite de la coordination entre les postes.

Cependant, la classification présentée précédemment (section 1.1.1) ne correspond pas directement au découpage fonctionnel de l'espace aérien. Celui-ci est découpé en entités appelées *secteurs élémentaires*. Un secteur élémentaire est un volume d'espace délimité par un contour géographique, un niveau de vol plancher et un niveau de vol plafond. Certains secteurs élémentaires peuvent être regroupés afin de créer un *secteur de contrôle*, c'est-à-dire une portion de l'espace dont un binôme (voir section 1.3.2) de contrôleurs a la charge.

**Centres de contrôle** La gestion des secteurs de contrôle est répartie entre plusieurs *centres de contrôle*[4], qui ont pour mission de définir les secteurs de contrôle ouverts et d'affecter des contrôleurs à la gestion de chacun de ces secteurs. La figure 1.3 montre le découpage de l'espace aérien français en secteurs de contrôle.

À chaque secteur de contrôle est associée une certaine *capacité* qui ne doit jamais être dépassée et qui est attribuée par le centre de contrôle en charge du secteur.

4. En France, il y a cinq centres de contrôle situés à Athis-Mons (région Parisienne, parfois découpé en deux centres), Reims, Brest, Aix-en-Provence et Bordeaux.

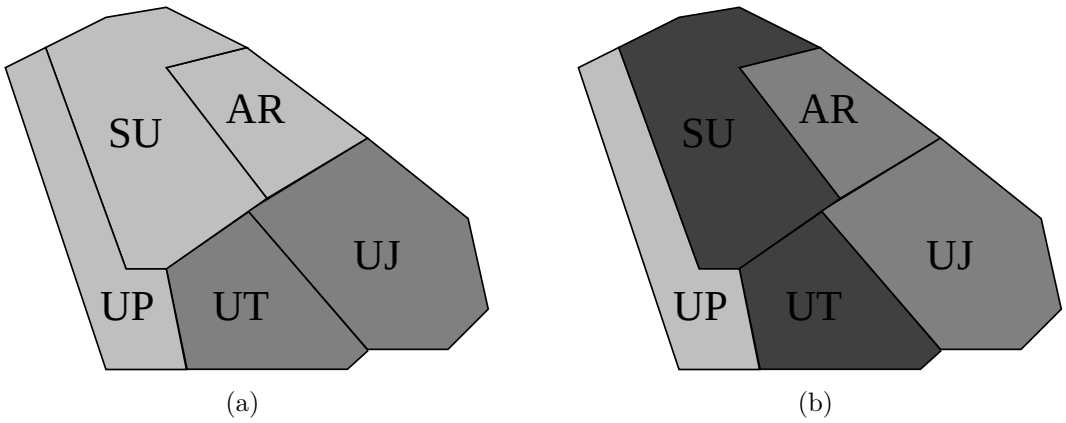



FIGURE 1.4 – Opérations de regroupement/dégroupement de secteurs : chaque intensité de gris représente un secteur de contrôle. La configuration (a) est adaptée à des ux est-ouest tandis que la configuration (b) est adaptée à des ux nord-sud.

**Dé nition 1 (Capacité)** *La* capacité *d un secteur de contrôle est dé nie en Europe comme le nombre maximum d avions entrant en une heure.*

La capacité d'un secteur de contrôle est définie en fonction d'un certain nombre de critères empiriques, parmi lesquels :

- la taille du secteur ;
- la nature des ux – un ux d'avions évolutifs (en montée ou en descente) est plus difficile à contrôler qu'un ux stable ;
- le nombre d'intersections de ux ;
- la présence de zones militaires ;
- l'expérience du trafic passé.

**Schémas d'ouvertures** Les secteurs de contrôle sont modifiables au cours de la journée en fonction de la quantité de trafic, c'est-à-dire que le nombre de secteurs élémentaires qui le composent peut être augmenté ou réduit. Ces opérations sont appelées *regroupement* et *dégroupement* et sont réalisées au sein des secteurs d'un même centre de contrôle. La figure 1.4 illustre ces opérations de changement de configuration. La configuration des secteurs de contrôle pouvant évoluer dans la journée, la « capacité globale » de l'espace peut varier, même si la capacité d'un secteur de contrôle donné, elle, n'évolue pas. Chaque centre de contrôle effectue une prévision, appelée *schéma d ouvertures*, des man uvres de regroupement et de dégroupement qui devront avoir lieu le jour suivant, en fonction des prévisions de trafic et des ressources disponibles (nombre de contrôleurs notamment).

Une fois les capacités déterminées pour l'ensemble des secteurs de contrôle, il est nécessaire de mettre en place un système de régulation permettant d'écouler le trafic sans les dépasser.

## 1.2 Régulations

La *Central Flow Management Unit (CFMU)* est un organisme européen chargé de coordonner les différents organismes de contrôle nationaux de la zone Conférence Européenne de l'Aviation Civile (CEAC), afin de garantir la sécurité des vols tout en optimisant l'écoulement du trafic. Elle a notamment pour mission de réguler les ux de trafic afin d'éviter les zones de congestion et d'utiliser l'espace aérien au maximum de sa capacité. Cette section présente les moyens de régulation mis en place par la CFMU pour mener à bien ces missions, ainsi que les coûts (en termes de retard) de ces régulations.

### 1.2.1 Système tactique d'aide à la régulation

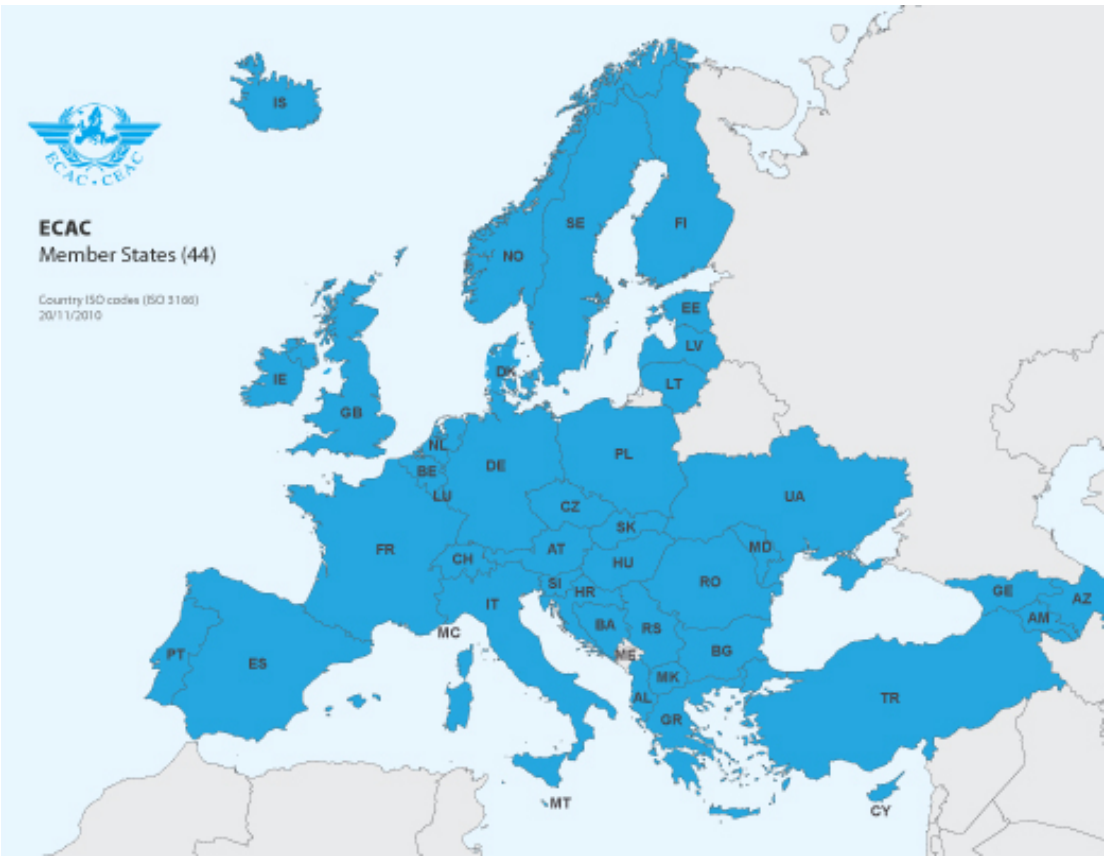
Le système *Enhanced Tactical Flow Management System (ETFMS)* mis en uvre par la CFMU [CFMU 11] propose un certain nombre de données et de représentations de la situation actuelle ainsi que de la demande de trafic. Il propose également un outil d'aide à la régulation : *Computer Assisted Slot Allocation (CASA)*. Cet outil possède deux degrés de liberté pour proposer des régulations : les heures de décollage ou le re-routement.

ETFMS est alimenté par les demandes de plans de vol pour les prochaines 48 heures, ainsi que par les positions courantes des vols européens, fournies par les services de contrôle nationaux. À partir de ces informations, des profils de vol en quatre dimensions (4D) sont calculés pour chaque vol prévu. Ces profils permettent de calculer les charges de trafic de tous les secteurs de la zone CEAC, contrôlée par la CFMU (voir figure 1.5). Les charges de trafic ainsi calculées sont mises à disposition des différents acteurs du contrôle via des outils de visualisation. Si certains vols (ou ux de vols) sont re-routés afin d'éviter des dépassements de capacité, les modifications sont automatiquement propagées dans le système et les différentes visualisations sont mises à jour. Les visualisations proposées fournissent également une interface pour l'utilisation de l'outil CASA.

### 1.2.2 Algorithme CASA pour l'aide à la régulation

CASA est un système dont le rôle est de fournir des *créneaux de décollage* aux avions traversant des zones saturées, celles-ci étant déterminées lors de la phase de calcul des charges de trafic décrite précédemment. Un créneau de décollage est une heure de décollage devant être respectée à $-5/+10$ minutes.

La CFMU et les organismes de contrôle définissent conjointement les zones pour lesquelles des mesures de régulation – Air Traffic Flow and Capacity Management (ATFCM) – sont nécessaires ; ces zones sont décrites par des frontières géographiques (souvent un secteur de contrôle) ainsi que par des heures de début et de fin de régulation. Pour chacune de ces zones, l'algorithme gère une *Liste d Allocation de Créneaux (LAC)*, vide à l'initialisation de l'algorithme, dont la taille dépend de la capacité (définition 1) de la zone considérée. L'exemple 1 détaille le calcul de la taille d'une LAC.




Source : Eurocontrol

FIGURE 1.5 – Les pays membres de la zone CEAC en 2011.

**Exemple 1 (Liste d'allocation de créneaux)** *Soit une zone de régulation d une durée de deux heures ayant une capacité de* 28 *avions par heure. La LAC associée à cette zone a une taille de* $56 = 2 \times 28$. *Les créneaux sont répartis approximativement toutes les* 2 *minutes (fréquence* $= \frac{durée}{capacité} = \frac{2\times60}{56}$*).*

L'algorithme fonctionne selon le principe « premier arrivé, premier servi », c'est-à-dire que les vols doivent passer dans la zone régulée dans le même ordre que s'il n'y avait pas eu de régulation. Les vols dont les plans de vol sont connus [5] se voient attribuer un créneau de la LAC en fonction de l'heure estimée d'arrivée dans la zone régulée ; ce créneau est interne au système (c'est-à-dire qu'il n'est pas diffusé) et est susceptible d'être mis à jour à l'apparition de nouveaux plans de vol – on parle de *phase de pré-allocation*.

Lors de la réception d'un nouveau plan de vol, la pré-allocation est effectuée comme décrite précédemment. Si le créneau correspondant est déjà occupé, celui-ci est attribué à celui des deux vols dont l'heure d'arrivée dans la zone est la plus proche. Le vol qui avait été prévu dans ce créneau prendra le créneau suivant. Si celui-ci est occupé, le processus se répète. Notons que ce procédé peut conduire à la ré-allocation successive de plusieurs

5. L'acte de dépôt de plan de vol constitue de fait une demande de créneau de décollage.

créneaux. Puis le créneau est fixé et diffusé quelques minutes avant le départ du vol. À partir de ce moment, le créneau ne peut plus être modifié et alloué à un autre vol.

Lorsqu'un vol est soumis à plusieurs régulations (c'est-à-dire qu'il traverse plusieurs zones régulées), c'est le retard le plus élevé qui est appliqué. Par exemple, si un vol traverse deux zones régulées, et se voit attribuer un retard de 10 minutes pour la première zone et un retard de 15 minutes pour la seconde, alors ce vol se verra attribuer un retard de 15 minutes ; le créneau de passage dans la première zone sera forcé à cette valeur. Cette procédure constitue une des limites de ce modèle : en effet, il n'est pas toujours possible de faire en sorte que le créneau imposé par une zone soit disponible dans les autres zones, ce qui peut conduire à une violation de contrainte de capacité.

Les régulations imposées par la CFMU peuvent mener à des retards importants, comme mentionnés dans la section suivante. C'est particulièrement le cas des régulations mises en place dans les secteurs de contrôle en-route les plus chargés.

### 1.2.3 Régulations et retards

Les rapports annuels édités par Eurocontrol [CODA 09] fournissent des détails sur les quantités ainsi que les causes des retards dans la zone CEAC. Dans ces rapports, un vol est considéré comme retardé lorsque son retard est supérieur à cinq minutes. Les résultats numériques présentés dans cette section tiennent compte de cette définition.

En 2009 [6], il y a eu 38 % de vols retardés avec une moyenne de 28 min. Les quatre composantes principales de ces retards [7] sont les compagnies aériennes (problèmes techniques sur l'aéronef, embarquement de passagers ou de fret, grèves...) pour 49 %, les mesures de régulation ATFCM pour 25 %, la congestion aéroportuaire pour 18 % et les mauvaises conditions météorologiques pour 8 %. Ces retards concernent essentiellement les vols évoluant entre les FL 330 et 390, c'est-à-dire la plupart des vols commerciaux.

Les retards dus aux régulations ATFCM constituent donc une part importante du retard total observé : ils concernent 7,5 % des vols, avec en moyenne 21 minutes par vol retardé. Les coûts associés à ces retards pèsent essentiellement sur les compagnies aériennes. La section qui suit donne un ordre de grandeur de ces coûts.

### 1.2.4 Coûts des retards

De nombreux éléments rentrent en ligne de compte pour le calcul du coût d'un retard : coûts des personnels, du carburant, de la maintenance, prise en charge et indemnisation des passagers... [Cook 04, Cook 09] étudient précisément les différents types de retards et

6. Les chi res de retards pour 2010 (retard moyen par vol retardé de 33 minutes, en hausse de 17 % par rapport à 2009) sont moins représentatifs car l'année fut perturbée en Europe, notamment par l'éruption du volcan Eyjafjallajökull qui a eu un impact très fort sur le nombre d'annulations de vols ainsi que sur les retards.

7. Ces composantes sont celles du retard dit *initial*, par opposition au retard *secondaire* qui est la propagation du retard d'un vol sur un autre vol (correspondance, utilisation du même avion...).

fournit une analyse détaillée des coûts de chacun. Les résultats [8] qui ressortent de ces études sont les suivants : en moyenne, le coût des retards liés aux régulations ATFCM sont de 0,30 € par passager, par minute de retard et par vol retardé. Un retard sur un vol long courrier (avec souvent plus de 300 passagers) coûtera plus cher qu'un retard sur un vol domestique (entre 100 et 200 passagers suivant le type d'avion). En considérant la  otte au moment de l'étude, on obtient une moyenne de 72 € par minute de retard et par vol retardé. [Bontemps 04] remet en cause certaines des hypothèses de ces études, concernant notamment le coût en perte de parts de marché [9] et la qualification en « retard long » de certains retards réels. L'estimation corrigée du coût moyen de la minute de retard est alors conforme aux chiffres avancés par [ITA 00], c'est-à-dire entre 35,5 et 50,9 € par minute.

Cependant, quelle que soit l'approche retenue, les coûts associés aux retards dus aux régulations ATFCM sont très élevés : pour l'année 2002 sur l'Europe (zone CEAC), ils ont été estimés entre 840 et 1 200 millions d'euros. La réduction de ces retards de régulation constitue donc un enjeu essentiel pour les compagnies aériennes. C'est également un défi pour les autorités de contrôle et de régulation, qui doivent tenter de réduire ces retards tout en maintenant la sécurité à son niveau actuel.

## 1.3 Contrôle du tra  c aérien

Les services du contrôle de la navigation aérienne ont pour objectif d'assurer la sécurité et la  uidité du trafic. Afin de réaliser cet objectif, les cinq missions suivantes doivent être menées à bien [SCA 06] :

- empêcher les collisions entre avions ;
- empêcher les collisions entre les avions et les éventuels obstacles situés sur l'aire de man  uvre ;
- accélérer et ordonner la circulation aérienne ;
- fournir les avis et les renseignements utiles à l'exécution sûre et efficace des vols ;
- alerter les organismes de secours (recherche et sauvetage) lorsqu'un avion en a besoin.

### 1.3.1 Di  érents types de contrôle

Les méthodes de contrôle varient selon les phases du vol, car les vitesses d'évolution et les man  uvres possibles sont différentes. Par exemple, dans les zones proches des aérodromes, les avions ont une vitesse limitée à 250 kt, alors que les vitesses en croisière sont de l'ordre de 450 kt pour la plupart des avions effectuant les vols commerciaux. Ainsi, on distingue trois types de contrôle : le contrôle d'aérodrome, le contrôle d'approche et le contrôle en-route.

8. Ces résultats s'appliquent au tra  c aérien européen et nécessiteraient une étude complémentaire pour pouvoir être transposés à d'autres espaces aériens.

9. L'étude de l'Université de Westminster fait l'hypothèse que, pour des retards récurrents, les passagers se tournent vers d'autres modes de transport. En réalité, une majorité de ces passagers se tournent simplement vers d'autres compagnies aériennes et la perte associée ne peut pas être comptée dans le coût du système.

#### Contrôle d'aérodrome

Le contrôle d'aérodrome a la charge d'une zone restreinte autour de l'aérodrome. Ses principales fonctions concernent la gestion des pistes, au décollage comme à l'atterrissage. Il gère également la circulation des avions au sol [10], entre la piste et la zone de parking. Ce contrôle s'effectue depuis la vigie de l'aérodrome – communément appelée la tour de contrôle.

#### Contrôle d'approche

Le contrôle d'approche gère les phases de montée et de descente des vols. Ces phases sont particulièrement compliquées, car elles suivent des procédures bien précises, qui varient en fonction de la météo (notamment en fonction du vent). Les espaces concernés sont situés autour des aéroports (10 NM de la piste en général) et ont une densité de trafic élevée, car toutes les trajectoires convergent vers un même point avant la piste.

#### Contrôle en-route

Le contrôle en-route gère la phase de croisière, phase qui concerne plus particulièrement notre étude. Les vols évoluent sur des routes aériennes, comme décrit dans la section 1.1.2. Le contrôle en-route s'effectue depuis les centres de contrôle en-route, au nombre de 75 pour la zone CEAC (5 en France).

### 1.3.2 Méthodes du contrôle

Chaque secteur est contrôlé par une paire de contrôleurs : le contrôleur *organique* et le contrôleur *tactique*, comme le montre la figure 1.6.

Le contrôleur organique est chargé d'organiser le trafic du secteur à moyen terme. Il s'occupe notamment de la phase de *coordination*, qui consiste à effectuer le transfert des avions entre les secteurs de contrôle, c'est-à-dire fournir (resp. récupérer) les informations importantes du vol lors de la sortie (resp. de l'entrée) du secteur. Pour chaque vol entrant dans son secteur, le contrôleur organique imprime une bandelette de papier appelée *strip*, qui compile les données nécessaires au suivi du vol à l'intérieur du secteur (voir figure 1.7) : altitude, vitesse, route suivie... Ces strips sont transmis au contrôleur tactique qui les organise sur son espace de travail.

Le contrôleur tactique s'assure de la sécurité des vols et de leur écoulement uide. Il prévient les collisions en s'assurant que l'espacement entre les vols est suffisant, c'est-à-dire qu'il respecte la *norme de séparation* (voir définition 2).

**Dé nition 2 (Norme de séparation)** *La* norme de séparation *est une distance minimale à respecter a n d assurer la sécurité des avions. Elle est composée de :*

$N_h$ *le minimum de séparation horizontale ou* norme de séparation horizontale *;*

10. Sur les plus grands aéroports, un service indépendant gère la circulation au sol pour diminuer la charge de travail des contrôleurs.

FIGURE 1.6 – Poste de contrôle dans le centre en-route d'Athis-Mons. Le contrôleur organique est à droite, le contrôleur tactique à gauche.

$N_v$ *le minimum de séparation verticale ou* norme de séparation verticale.
*L une ou l autre de ces deux normes doit être respectée pour assurer la bonne séparation.*

Pour la phase en-route du vol, les valeurs des normes de séparation sont [OACI 96] : $N_h = 5\,\text{NM}$ et $N_v = 1\,000\,\text{ft}$. Dans les zones terminales, gérées par le contrôle d'approche, la séparation horizontale peut être réduite à 3 NM. La figure 1.8 représente le volume de protection inhérent à la norme de séparation pour la phase en-route du vol.

Le contrôleur tactique dispose d'une visualisation de données radar, lui permettant de connaître pour chaque avion son identifiant, sa position et sa vitesse (voir figure 1.9). Cette visualisation lui permet d'indiquer aux avions (par radio) des man uvres d'évitement efficaces [11]. Les man uvres d'évitement horizontales sont les changements de cap ou de vitesse, les man uvres verticales sont les changements de niveau de vol, les anticipations de descente ou interruptions de montée.

11. Auparavant, les contrôleurs disposaient uniquement des informations de position données par les pilotes, menant à des trajectoires d'évitement plus longues a n d'assurer l'espacement avec su samment de marge de man uvre.

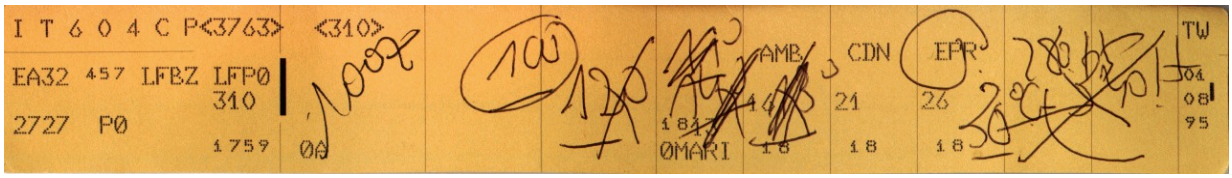



FIGURE 1.7 – Un *strip* de contrôle annoté par un contrôleur.

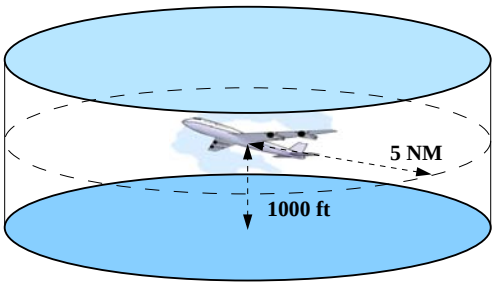



FIGURE 1.8 – Norme de séparation pour la phase en-route. Aucun autre avion ne doit se trouver dans ce volume.

## 1.4 Limites du modèle actuel

Le système de gestion du trafic aérien actuel est efficace et réactif. Néanmoins, certaines zones de l'espace aérien sont déjà régulièrement saturées, mettant ainsi en évidence des limites structurelles liées au modèle. Le trafic aérien est quant à lui dans une phase de croissance qui devrait, selon les prévisions réalisées par Eurocontrol, se poursuivre pour les décennies à venir (cf. section 1.4.1). Cette augmentation de la demande risque de saturer de manière rédhibitoire le système actuel (cf. section 1.4.2).

### 1.4.1 Évolutions du tra c aérien en Europe

Les prévisions de trafic en Europe effectuées par l'unité *Statistics and Forecasts (STATFOR)* d'Eurocontrol [STATFOR 10] indiquent une tendance à la hausse malgré la crise économique. Plusieurs scénarios sont envisagés à l'horizon 2030, prenant en compte un certain nombre de paramètres économiques, technologiques et environnementaux. Ces scénarios, ainsi que leurs principales caractéristiques, sont les suivants (du plus optimiste au plus pessimiste) :

- Scénario A : croissance globale – croissance économique forte, mondialisation en plein essor, technologies au point pour la maîtrise des ressources (notamment pétrolières).
- Scénario C : croissance régulée – croissance économique modérée et mise en place de mesures de régulation pour prendre en compte les enjeux environnementaux et sociaux liés au développement durable.
- Scénario D : fragmentation géopolitique – tensions accrues entre les différentes régions du globe, échanges économiques réduits et économies plus faibles.

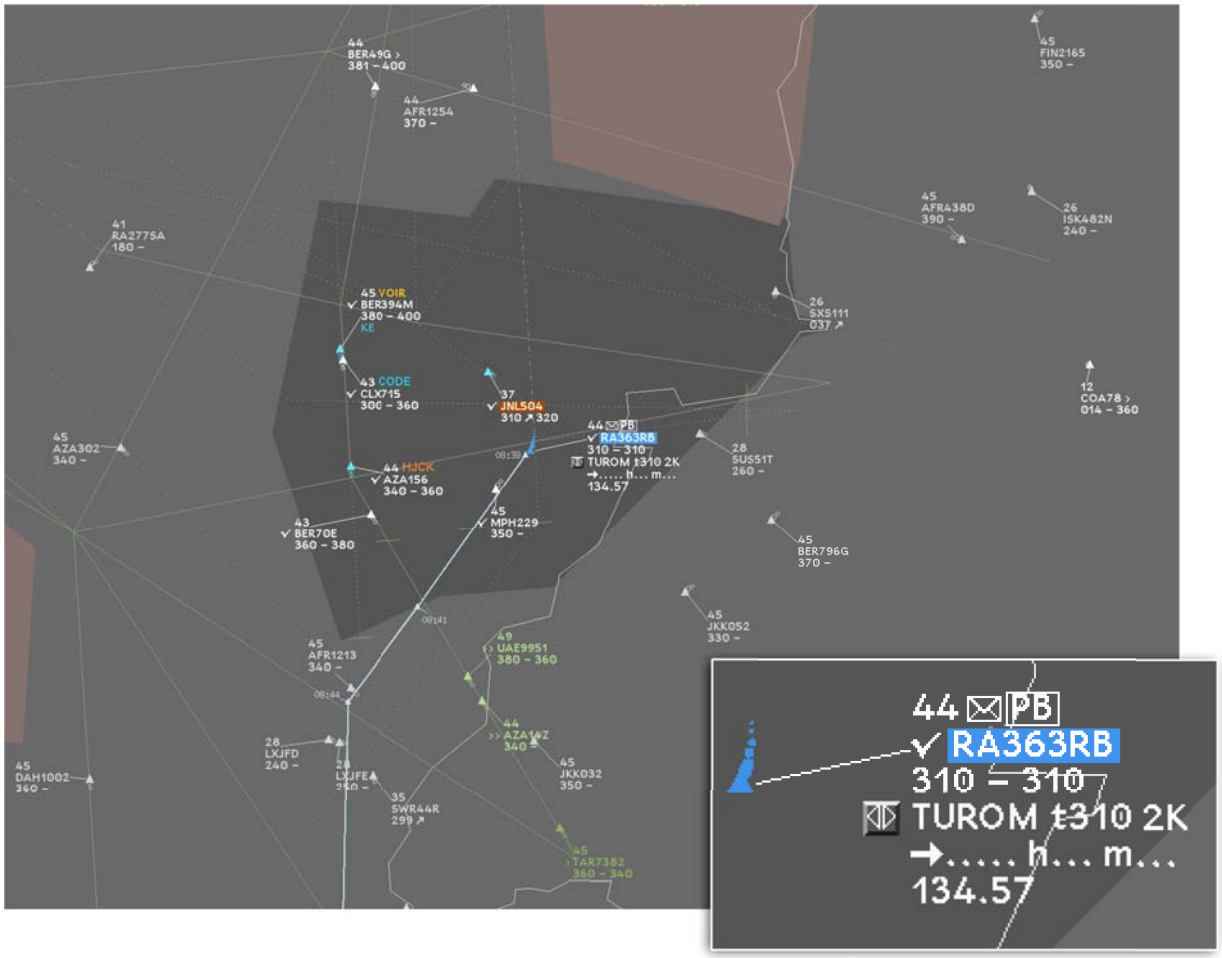
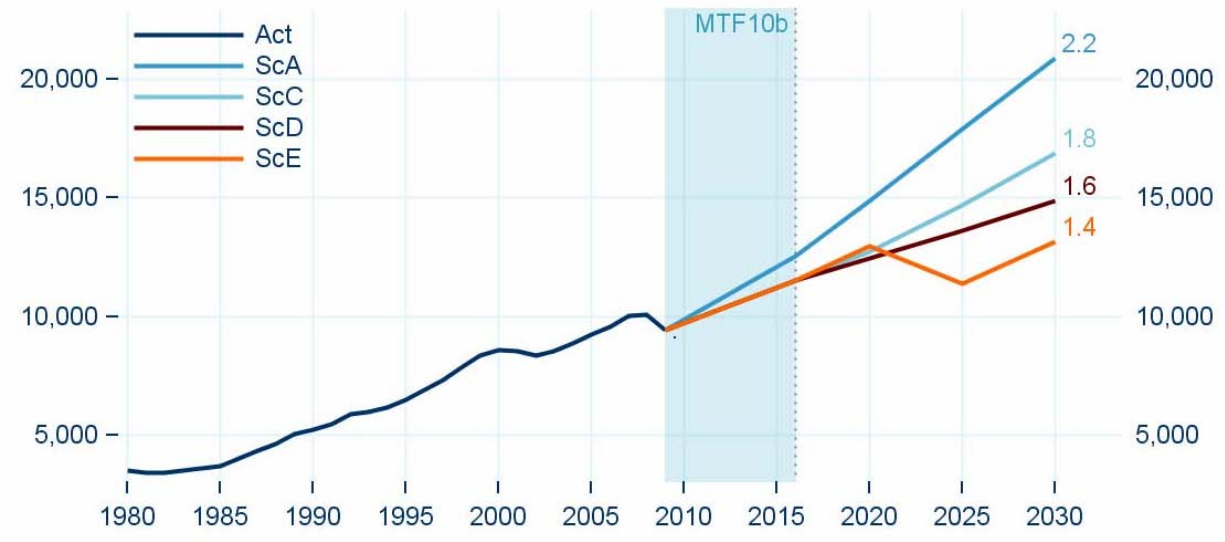



FIGURE 1.9 – Visualisation des données radar. À chaque avion est associée une étiquette qui contient sa vitesse, son identifiant, son niveau de vol et son évolution verticale. La trace laissée derrière l'avion donne une indication visuelle de sa vitesse. Le détail (en bas, à droite) montre une « macro-étiquette » affichée lors de la sélection d'un vol, qui contient plus d'informations (prochaine balise de la route, niveau de vol de sortie du secteur...).




Source : Eurocontrol-STATFOR

FIGURE 1.10 – Prévisions de trafic en Europe à l'horizon 2030 : nombre de vols annuels (en milliers) selon les différents scénarios envisagés.

- Scénario E : ressources limitées (*pic pétrolier*[12]) – économies fortes, mais orientées sur le profit à court terme plutôt que sur le développement durable, et incapables de s'adapter rapidement à une décroissance de la production pétrolière (après un pic de production en 2020).

Le scénario B, présent dans la précédente version de l'étude, et qui se basait sur le maintien au niveau actuel des indicateurs de croissance, a été écarté au profit du scénario E jugé plus pertinent.

Bien que le scénario C soit le plus mis en avant par cette étude, de nombreuses études prévoient le pic pétrolier mondial avant 2020, certaines le disent déjà franchi [IAE 09, EWG 07, Deffeyes 06], ce qui privilégie le scénario le plus pessimiste en terme de trafic aérien.

Néanmoins, quel que soit le scénario envisagé, la STATFOR prévoit une croissance significative du trafic aérien européen : entre 40 % et 120 % de vols en plus en 2030 par rapport au trafic de 2009, comme le montre le graphique de la figure 1.10.

### 1.4.2 Saturation de l'espace aérien

Si le trafic augmente, l'espace aérien quant à lui reste le même, et sa gestion actuelle sera saturée dans quelques années. [Eurocontrol 08] identifie principalement cinq défis à relever afin que la demande croissante du trafic aérien puisse être absorbée. Deux d'entre eux ont un intérêt particulier dans le cadre de la gestion de l'espace aérien et de sa capacité :

12. On parle de pic pétrolier pour désigner la période pendant laquelle la production mondiale de pétrole stagne à une valeur maximale avant de décliner à cause de l'épuisement des ressources.

- l'augmentation de la capacité aéroportuaire, qui sera la ressource limitante dans les scénarios A et C (voir section 1.4.1) ;
- l'écoulement du trafic dans des secteurs de contrôle dont la capacité, telle que définie actuellement (définition 1), sera dépassée.

Notre étude étant focalisée sur la phase en-route des vols, c'est ce dernier point qui nous intéresse particulièrement.

Le découpage de l'espace en secteurs de contrôle a permis de diminuer la charge de travail des contrôleurs en augmentant leur nombre et en leur répartissant le trafic. Mais les secteurs de contrôle situés dans la *core area*[13] sont saturés au cours d'une grande partie de la journée. Un découpage plus fin permettrait en théorie d'augmenter la capacité de cet espace en réduisant la charge de travail du contrôleur tactique, mais le temps passé à réaliser les coordinations (tâche du contrôleur organique) augmenterait alors de manière rédhibitoire, si bien que les secteurs de contrôle actuels sont déjà à leur taille minimale.

Il paraît néanmoins intéressant de se demander si les capacités des secteurs de contrôle sont une métrique adaptée à la mesure de la charge de travail des contrôleurs. En effet, pour un même volume d'espace et un même taux horaire de vols traversant ce volume, la difficulté du contrôle va être différente selon la topologie du trafic. [Gianazza 06] met en uvre une méthode d'apprentissage afin de déterminer les métriques les plus significatives (parmi un panel de 27 métriques présélectionnées) pour représenter la charge de travail d'un contrôleur aérien. Le résultat de cette étude est un ensemble de six métriques, qui sont les suivantes :

- volume du secteur de contrôle ;
- nombre d'avions à l'intérieur du secteur ;
- vitesse verticale moyenne des avions ;
- ux entrants à 15 et 60 minutes d'horizon ;
- nombre de croisements potentiels avec un angle supérieur à 20 degrés.

On remarque que les métriques retenues prennent en compte le nombre d'avions et les ux entrants, comme la définition actuelle de la capacité, mais également les trajectoires des avions contrôlés via leur évolution verticale et leur interaction avec les autres avions. Un modèle de régulation basé sur des métriques de ce type permettrait de calculer les capacités de manière plus juste et d'obtenir des meilleures prévisions pour celles-ci. En effet, on constate, avec le système actuel, un écart significatif entre les schémas d'ouvertures et les secteurs réellement ouverts [Gianazza 10], mettant en évidence une surestimation de la charge de travail des contrôleurs.

Cependant, même si on modifie la définition de la capacité de manière à mieux prendre en compte la charge de travail réelle des contrôleurs, les régulations reposent sur l'algorithme CASA (voir section 1.2.2) qui est une heuristique gloutonne[14]. Il serait d'ailleurs difficile d'adapter cet algorithme à une capacité définie à l'aide des métriques présentées précédemment, puisque celles-ci dépendent de chaque vol (notamment pour les angles de

13. La *core area* est la portion d'espace aérien située sur l'axe Londres-Francfort-Rome, où la densité de tra c est la plus élevée en Europe.

14. Une heuristique gloutonne consiste à e ectuer, étape par étape, un choix localement optimal. La solution obtenue n'est pas toujours optimale, puisqu'aucun choix n'est remis en cause.

croisement et l'évolutivité verticale) plutôt que d'une mesure globale (le nombre de vols entrants). Il serait donc possible de réduire les retards liés aux régulations en utilisant d'autres algorithmes d'optimisation travaillant sur ce type de métriques, écoulant de fait une plus grande quantité de trafic.

L'ensemble de ces difficultés ouvre la voie à des améliorations du système actuel, tant sur le plan de la modélisation (découpage de l'espace en secteurs de contrôle, organisation du réseau de routes, définition des capacités) que sur celui des outils d'aide aux contrôleurs (visualisation, algorithme de régulation, communication). Le chapitre suivant donne un aperçu d'études réalisées dans ce contexte, ainsi que des tentatives d'automatisation du contrôle aérien.

# Chapitre 2

# Évolution des systèmes de contrôle

La progression du trafic aérien nécessite la mise en  uvre de nouveaux concepts et outils qui permettront aux contrôleurs d'assurer la sécurité des vols dans les années à venir. L'une des difficultés de cette tâche est de trouver un compromis satisfaisant entre une automatisation complète du contrôle et une absence totale d'automatisation. L'automatisation complète du contrôle ne paraît pas envisageable à l'heure actuelle, tant à cause des solutions techniques disponibles que des conséquences sociales et des difficultés de transition. Nous verrons à ce propos (section 2.1.2) que les projets d'automatisation mis en place ces dernières décennies ont pour la plupart été abandonnés. À l'inverse, une absence d'automatisation paraît également irréaliste : la quantité de données à traiter augmente avec le trafic et l'apparition de nouveaux moyens de communication entre les avions et le contrôle (*Data-Link*) ne fait qu'amplifier le phénomène. Le contrôleur aura donc besoin d'outils adaptés pour un traitement et une visualisation efficaces de ces informations. D'autre part, la croissance du trafic nécessite également une augmentation de la capacité du système de contrôle dans sa globalité, tout en maintenant la sécurité des vols au niveau actuel, voire en l'améliorant. Cette augmentation de capacité passera nécessairement par une automatisation de la gestion pré-tactique des  ux aériens.

De nombreuses études proposent des améliorations du système existant ou de nouveaux concepts de contrôle du trafic. On peut distinguer deux types d'approches pour les problèmes de résolution et de prévention des con  its aériens. Les études qui proviennent du monde opérationnel (section 2.1), en général à l'initiative des services du contrôle, prennent souvent en compte de nombreuses contraintes opérationnelles mais se soucient peu de l'optimalité des solutions. Les études théoriques (section 2.2) se focalisent à l'inverse sur l'optimisation du problème, avec souvent des hypothèses peu réalistes.

## 2.1 Approches opérationnelles

Les organismes de contrôle du trafic envisagent l'automatisation de tout ou partie du système de contrôle depuis quelques décennies. Cela se traduit par l'émergence de nouveaux concepts (section 2.1.1) remettant en cause de manière significative certains aspects limitants

du modèle actuel ainsi que par le lancement de plusieurs projets d'envergure (section 2.1.2) utilisant ces nouveaux concepts pour proposer une automatisation complète de la séparation des avions.

### 2.1.1 Nouveaux concepts ATM

Parmi les nouveaux concepts pour l'*Air Tra c Management (ATM)*, on trouve notamment les concepts *sector-less* et *free- ight*. L'objectif du *sector-less* [Duong 01] est de supprimer les secteurs de contrôle en « affectant » directement les avions aux contrôleurs aériens. Ceux-ci suivront donc un groupe d'avions pendant toute la durée de leur vol en évitant les con its avec le reste du trafic. Le *free- ight* [RTCA 95], quant à lui, consiste à accorder aux pilotes et compagnies aériennes une plus grande liberté dans le choix de leurs trajectoires, qui pourront alors prendre la forme de routes directes entre aéroports de départ et d'arrivée. Ce concept peut être abordé de plusieurs manières en se basant soit sur des équipements d'anticollision embarqués (c'est l'approche européenne) avec des avions qui assurent eux-mêmes leur séparation en négociant entre eux les trajectoires de façon automatique, soit sur un système d'anticollision au sol (c'est l'approche américaine, aussi appelée *free-route* [dos Santos 02]) qui négocie avec chaque avion une trajectoire sans con it. Le concept *free-route* nécessite que les avions soient équipés d'un système de navigation permettant le suivi d'une trajectoire 4D. Les différences de point de vue entre *free- ight* et *free-route* font apparaître une question importante à se poser lors de l'élaboration d'un nouveau modèle : contrôle centralisé ou contrôle autonome ?

### 2.1.2 Tentatives d'automatisation

Ces nouveaux concepts ont ouvert la voie à des évolutions potentiellement significatives des systèmes du contrôle aérien, de sorte que l'on peut se demander si une automatisation complète de celui-ci est envisageable [Durand 99]. Cette section présente quelques projets provenant du monde opérationnel qui ont tenté de mettre en uvre une telle automatisation [Alliot 03].

**AERA** Le projet *Automated En-Route Air Tra c Control (AERA)* [Niedringhaus 83], financé par la *Federal Aviation Administration (FAA)*, était composé de trois phases. L'objectif de la première phase, AERA-1, était de prévoir les trajectoires des avions en fonction des intentions des pilotes et de détecter les éventuels con its entre ces trajectoires. Cet outil d'aide à la décision ne proposait aucune solution pour la résolution des con its détectés.

Cette possibilité a été ajoutée à l'outil dans la seconde phase du projet, en proposant au contrôleur une liste de résolutions recommandées. AERA-2 apportait également un outil d'aide à la coordination entre secteurs de contrôle.

Enfin, AERA-3 devait proposer un outil responsable de la séparation des avions, avec une structure en trois niveaux :

1. ASF (Automated Separation Function) séparant les paires d'avions ;
2. MOM (Man uver Option Manager) assurant le respect du contexte global par ASF ;
3. AMPF (Airspace Manager Planning Functions) faisant en sorte que MOM puisse opérer dans un espace de densité suffisamment faible.

Ce projet n'a pas abouti et n'a pas permis d'envisager l'optimisation globale du problème de résolution de con it, car aucun rapport ne précise quels algorithmes devaient être utilisés pour les outils MOM et AMPF.

**ARC 2000** La première tentative européenne d'automatisation complète se nomme ARC 2000 (Automatic Radar Control) [Fron 93]. Le principe était d'attribuer à chaque avion, dans l'ordre de leur arrivée dans le système, un « tube 4D » évitant les tubes déjà attribués, sans remise en cause de ceux-ci. Par la suite, des heuristiques différentes du « premier arrivé, premier servi » ont été mises en uvre, mais comme pour le projet AERA, l'optimalité globale n'a jamais fait partie des objectifs affichés. Or celle-ci est nécessaire à la mise en place d'un système automatisé dans la mesure où les usagers de l'espace aérien (notamment les compagnies aériennes) doivent pouvoir compter sur un système au moins aussi performant que le système actuel en termes de retards ou de consommation de carburant.

Le projet a été arrêté au milieu des années 90, en laissant tout de même derrière lui un outil de représentation des trajectoires et des con its et d'aide à la décision nommé HIPS (Highly Interactive Problem Solver [Meckiff 94]).

**SAINTEX** Le projet SAINTEX [1] [Angerand 92] propose trois scénarios d'automatisation pour le trafic en-route dans les espaces de faible densité :

- Un scénario « détection-résolution » est basé sur un système expert qui tente de reproduire le comportement du contrôleur. Les con its sont détectés en extrapolant les trajectoires de quelques minutes, puis classés selon différents critères (*e.g.* angle formé par les trajectoires, vitesses relatives...). Une man uvre prédéfinie est appliquée pour chaque classe de con it.
- Un scénario « 4D » est similaire à l'approche du projet ARC 2000, puisque chaque avion entrant dans le secteur de contrôle considéré se voit attribuer une trajectoire évitant les vols déjà présents.
- Un scénario hybride propose de traiter les avions stables (*i.e.* en croisière) par l'approche « détection-résolution » et les autres par l'approche « 4D ».

L'utilisation d'un système expert pour la résolution de con its à deux avions n'est pas particulièrement adapté à un système d'automatisation, car il est possible à l'aide d'algorithmes plus simples de proposer des man uvres de coût optimal avec un temps de calcul très faible dans ce cas. Le problème des con its impliquant plus de deux avions ou des con its dépendants (la déviation d'un avion pour éviter un con it peut entraîner un autre con it par la suite) était mis en évidence mais aucune solution n'a été proposée pour le résoudre.

1. Le nom fait référence à *Vol de nuit* (Antoine de Saint-Exupéry, 1930), le projet étant prévu pour être mis en uvre dans les espaces aériens de faible densité, comme c'est le cas la nuit.

**FREER** Le projet Free-Route Experimental Encounter Resolution (FREER) proposé par Eurocontrol [Duong 97] utilise le concept de *free-route* évoqué précédemment dans une méthode de contrôle à deux niveaux, basée sur un principe similaire au projet AERA. Le niveau FREER-1 consiste à laisser les avions assurer leur séparation avec le reste du trafic de manière autonome dans les espaces de faible densité (zones maritimes, altitude élevée...). Les règles décrites permettent la résolution de con its impliquant deux ou trois avions. Le niveau FREER-2 est un filtre assurant que les résolutions puissent se dérouler correctement en veillant à ce que les espaces de *free-route* ne soient pas saturés.

**ERASMUS** Le projet ERASMUS (En-Route Air Traffic Soft Management Ultimate System, [Villiers 04]) est plus récent que les projets précédents. L'idée est de proposer un système centralisé permettant de détecter les trajectoires en con it avec une anticipation de 20 min et de résoudre ces con its par des régulations en vitesse lorsque c'est possible. Dans le cas contraire, *i.e.* si la régulation en vitesse n'est pas suffisante, le contrôleur aérien résoudra le con it avec ses méthodes habituelles. Cette méthode peut donc être vue comme un filtre pour le contrôleur, qui peut être intégré de deux manières différentes dans le système actuel :

- soit comme une action subliminale : les modifications de vitesses sont imposées et suffisamment faibles pour s'insérer dans la marge d'incertitude qu'ont les contrôleurs dans leur perception de la vitesse des avions ;
- soit en avertissant le contrôleur, voire même en lui demandant une confirmation avant application de la modification.

Pour les con its qui ne peuvent pas être résolus par la régulation en vitesse, la version actuelle d'ERASMUS ne fournit qu'une aide à la détection, sans proposer de man uvre de résolution.

**Bilan** Ces projets sont aujourd'hui tous terminés et n'ont pas permis d'aboutir à des systèmes opérationnels, ni même à des expérimentations sur trafic réel. Certaines modélisations sont intéressantes et mériteraient d'être étudiées d'un point de vue plus théorique, afin de mettre en uvre des algorithmes d'optimisation qui manquent à la plupart de ces approches. On pourra se référer à [Garot 05] pour plus de détails à propos de ces projets. La conclusion qui en ressort est la suivante : il est important de travailler à l'automatisation des systèmes de contrôle du trafic aérien, mais celle-ci doit être réalisée en gardant le contrôleur au centre du système et en lui proposant des outils d'aide à la détection et à la résolution des con its. Cette évolution du système de contrôle devra se faire par étapes successives pour des raisons de coût et de sécurité. Depuis quelques années, deux grands programmes de recherche, Single European Sky ATM Research (SESAR) [SESAR 07] en Europe et *NextGen* [FAA 11] aux États-Unis, tentent de fédérer la recherche dans le domaine de la gestion du trafic aérien en proposant un cadre (*i.e.* un ensemble de concepts utilisables, procédures, hypothèses...) et des projets pour l'évolution des systèmes de contrôle à l'horizon 2020. Même si ces programmes ont des objectifs particulièrement ambitieux en termes de sécurité et de performances, une partie des spécifications prend en compte les constats

réalisés lors des projets précédemment cités.

## 2.2 Approches théoriques

Parallèlement aux projets proposés par les organismes de contrôle, de nombreuses études se sont focalisées sur des problèmes de l'ATM en recherchant en priorité l'optimalité des solutions. Ces approches sont complémentaires des précédentes, qui ne s'intéressent que très rarement à l'optimalité des choix effectués mais prennent en compte un grand nombre de contraintes opérationnelles.

On trouve dans la littérature des études s'intéressant aux différents niveaux de filtrage du système de contrôle : résolution réactive des con its par des systèmes embarqués (section 2.2.1) ou centralisés (section 2.2.2), modification du réseau de routes (section 2.2.3) et amélioration des mesures de régulation du trafic (section 2.2.4).

### 2.2.1 Avions autonomes

Plusieurs études proposent une approche dans laquelle les avions assurent eux-mêmes leur séparation avec le reste du trafic. Ces méthodes reposent sur des man uvres horizontales (changement de cap de l'avion). Une des premières études de ce type [Zeghal 94] consiste à considérer les avions comme des objets sur lesquels s'exercent différentes forces représentant le trafic. Ainsi, chaque avion est soumis à :

- des forces attractives représentant les objectifs de l'avion (destination ou balises de la route) ;
- des forces répulsives représentant les obstacles à éviter (les autres avions et les zones interdites) ;
- des forces de glissement permettant de contourner ces obstacles.

Le paramétrage du système s'effectue en modifiant l'intensité des différentes forces. Ce modèle vise une bonne robustesse du système d'évitement plutôt que son optimalité globale. L'une des limites réside dans le fait que les trajectoires ainsi construites sont continues et donc difficilement réalisables par un pilote ou un système de contrôle du vol [2].

L'approche proposée par [Granger 98] est plus réaliste sur ce dernier point, puisque les trajectoires d'évitement sont du type « point tournant » (voir figure 2.1), c'est-à-dire avec un changement de cap à un instant $t_0$ et un second changement à l'instant $t_1$ pour retrouver la route prévue (cap vers la prochaine balise pour une route standard, cap vers la destination pour une route directe). Cette approche repose sur la définition d'un ordre de priorité global avec distribution de jetons : chaque avion distribue un jeton aux avions de son voisinage qui sont moins prioritaires (par exemple par ordre de numéro de transpondeur [3])

2. Les systèmes de vol permettent de prendre en charge les changements de cap associés à un ensemble de points (souvent les points de report de la route), mais ne sont pas capables de réaliser un changement continu de cap.

3. Le transpondeur est un appareil embarqué permettant d'identi er un avion de manière unique. Un code est attribué à chaque avion avant le décollage par les organismes du contrôle.

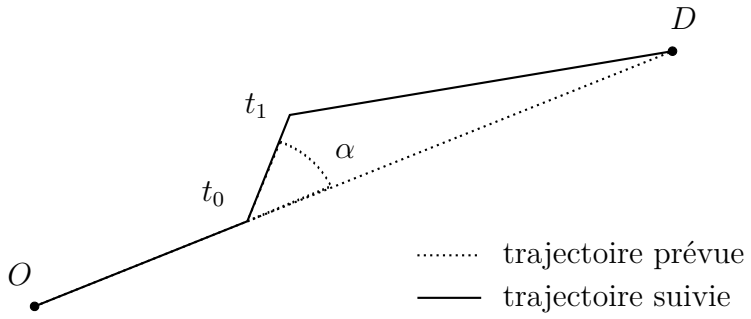



FIGURE 2.1 – Man uvre de type « point tournant » appliquée sur une route directe. $t_0$, $\alpha$ et $t_1$ sont les paramètres de la man uvre, $O$ l'origine de l'aéronef et $D$ sa destination.

que lui. Puis les avions n'ayant pas de jetons calculent leur trajectoire optimale évitant les avions plus prioritaires à l'aide d'un algorithme $A^\star$ avant de reprendre les jetons qu'ils ont distribués. Cette opération est répétée jusqu'à ce que chaque avion ait déterminé sa trajectoire. Ce modèle a l'avantage de reposer sur des trajectoires facilement réalisables mais la qualité des solutions dépend fortement de l'ordre de priorité choisi pour les avions et le principe même de priorité induit une résolution inéquitable vis-à-vis des compagnies aériennes. [Archambault 04b] présente plusieurs heuristiques d'ordonnancement permettant d'améliorer les solutions et de rendre la méthode plus robuste. Cependant, malgré ces améliorations, les résultats montrent que celle-ci n'est utilisable que dans les espaces de faible densité de trafic.

D'autres études envisagent ce type de résolution coopérative, notamment [Chiang 97], qui propose une résolution à l'aide d'algorithmes géométriques ou encore [Pallottino 02], qui utilise la programmation en nombres entiers mixte.

### 2.2.2 Résolution centralisée

D'autres études proposent des outils de résolution centralisés, avec divers degrés de liberté : cap, vitesse ou altitude.

[Carlier 03] utilise des techniques d'ordonnancement pour la résolution centralisée de con its. Un con it est défini comme une machine sur laquelle une seule tâche peut être effectuée, *i.e.* un seul avion peut occuper la zone de con it. Pour un con it impliquant deux avions, la résolution consiste à déterminer quel avion passe en premier et quelle durée est nécessaire entre le passage du premier et du second pour assurer la séparation. Les résultats présentés montrent une bonne résolution pour un con it (impliquant éventuellement plus de deux avions). Cependant, cette étude ne donne aucune idée concernant le type de man uvre à appliquer pour faire respecter ces contraintes. En outre, elle ne garantit rien dès lors qu'il s'agit de résoudre plusieurs con its : en effet, selon l'ordonnancement retenu, un avion peut avoir une attente trop importante entre deux con its pour que la solution soit réalisable (un avion ne peut pas s'arrêter en vol).

L'approche de [Granger 01] utilise le même type de man uvre « point tournant » que [Granger 98], mais dans le cadre d'un système centralisé. Les avions sont regroupés en

*clusters*[4] indépendants, puis un algorithme génétique est utilisé pour la résolution des con its de chaque *cluster*. Cette approche permet de résoudre la totalité des con its en-route pour une journée de trafic dans l'espace aérien français. Elle a été validée sur du trafic simulé, avec une anticipation de 12 à 15 minutes pour la détection des con its. Cette étude montre ses limites pour des temps d'anticipation plus élevés, avec des temps de calcul qui deviennent prohibitifs dans le cadre d'une résolution en temps réel. [Feydy 05] considère un problème identique en le modélisant par un problème de satisfaction de contraintes mixte. Les résultats obtenus sont comparables et cette étude apporte en outre une preuve de non existence ou d'optimalité de la solution. Cependant, la méthode est limitée à un nombre très faible d'avions.

En se fondant sur le même modèle centralisé avec man uvres de résolution horizontales, [Dodin 99] utilise un algorithme de programmation semi-définie afin d'ordonner les avions impliqués dans un con it, puis une méthode d'optimisation convexe permettant de déterminer les man uvres à effectuer en fonction de l'ordre défini dans la première étape. Les relaxations nécessaires à l'optimisation convexe rendent ce modèle très peu réaliste et empêchent donc une application concrète.

[Archambault 04a] propose une résolution des con its par ajustement des vitesses des avions. Cette méthode se base sur les processus de simulation et les algorithmes de résolution déjà mis en uvre dans [Granger 01]. Les man uvres sont décrites par le temps $t_0$ de changement de vitesse, la valeur modifiée de celle-ci et le temps $t_1$ de reprise de la vitesse nominale. Cette idée a été intégrée dans le cadre du projet ERASMUS [Villiers 04], avec des modifications de vitesses faibles (entre $-3\,\%$ et $+6\,\%$ par rapport à la vitesse nominale) afin de réaliser ces modifications de manière subliminale pour le contrôleur. Celui-ci trouve alors un trafic avec moins de con its, sans que la perception qu'il a de ce trafic ne soit altérée.

Une étude similaire [Constans 05a] propose de résoudre les con its par des modifications de vitesse. La résolution, effectuée avec un horizon glissant, utilise une méthode de contrôle optimal appliquée sur chaque sous-problème. [Rey 10] utilise également ce modèle à horizon glissant, mais propose une résolution par un modèle linéaire en nombres entiers mixte. L'inconvénient de ces modèles est la faible marge de man uvre laissée par les changements de vitesse : chaque avion a une « enveloppe de vol[5] » dont il ne doit pas sortir, notamment un intervalle de vitesse relativement restreint lors de la phase de croisière.

[Vela 09a] modélise le problème de résolution de con its par modifications des vitesses par un programme linéaire en nombres entiers mixte. Les con its sont résolus de manière réactive et des changements de niveau de vol sont également autorisés (avec, selon les simulations, un, deux ou trois niveaux disponibles). La programmation linéaire en nombres entiers mixte est utilisée dans une autre étude [Vela 09b], qui propose une résolution de con its par changements de cap en minimisant de manière globale la consommation de carburant. La charge de travail du contrôleur est également prise en compte en limitant le

4. Un *cluster* est la clôture transitive de la relation est en con it avec (voir section 4.3.4).

5. L'enveloppe de vol décrit les valeurs possibles de certains paramètres (vitesse, inclinaison...) en fonction notamment de l'altitude, de la température extérieure ou de la masse de l'appareil.

nombre de man uvres dans un intervalle de temps donné afin de permettre au contrôleur de transmettre par radio les informations nécessaires à la man uvre. Ce dernier modèle élude cependant certains aspects de la résolution, notamment l'in uence d'une man uvre de résolution sur la complexité du reste du trafic (la modification de la trajectoire permet l'évitement pour le con it considéré, mais peut par la suite générer un autre con it qui n'était pas prévu).

Enfin, [Flener 07] utilise la programmation par contraintes pour réduire la complexité du trafic dans un ensemble de secteurs. La complexité est calculée, pour chaque secteur, en fonction du nombre d'avions dans le secteur ou près de ses frontières ainsi que du nombre de vols en évolution (montée ou descente). Les degrés de liberté de cette méthode sont, pour chaque vol, son heure de décollage (dans un intervalle de -5/+10 minutes autour de l'heure prévue), sa vitesse et son altitude (dans un intervalle de -30/+10 FL autour du niveau de vol prévu). Les expérimentations sur des instances d'environ 2 000 vols montrent une réduction de la complexité d'environ 50 %.

### 2.2.3 Réseau de routes

Les études précédentes ont pour objet la résolution des con its dans un cadre réactif [6]. Mais il est également possible de modifier le réseau de routes ou l'utilisation de ce réseau [7] afin de réduire le nombre d'intersections potentielles. [Nace 03] utilise la programmation linéaire pour affecter une route et un niveau de vol aux avions dans le réseau de routes actuel. La réduction du nombre de con its potentiels est importante (entre 50 % et 60 %), mais les aspects concernant le coût des solutions en termes de durée de vol et de consommation de carburant ne sont pas abordés.

Afin de tenter de valider le concept *sector-less*, [Rivière 06] propose de créer un réseau de routes adapté à ce mode de contrôle du trafic. Le principe est de partir d'un réseau régulier (une grille de 4000 km de côté avec des mailles de 240 km) puis de modifier la position des n uds afin de minimiser la distance totale parcourue. La déformation du réseau initial est effectuée à l'aide d'un algorithme génétique ou d'un algorithme de recuit simulé. L'évaluation des solutions est faite en calculant pour chaque vol le plus court chemin dans le graphe ainsi créé. La solution optimale est ensuite évaluée du point de vue de la charge de travail du contrôleur à l'aide de méthodes et de mesures présentées dans [Averty 98]. Cette étude conclut que le concept *sector-less* n'est pas transposable dans la réalité du transport aérien actuel, notamment à cause d'une charge de travail trop élevée pour les contrôleurs. D'autre part, le réseau de route créé induit une augmentation de près de 20 % de la distance parcourue par rapport à l'utilisation de routes directes et les man uvres de changement de route impliquent des changements de niveau de vol systématiques, ce qui augmente encore la consommation de carburant.

[Gianazza 04b] propose d'attribuer à chaque ux de vols [8] une trajectoire 3D décrite par une route 2D et une succession de niveaux de vol. Deux méthodes sont comparées pour

6. Action au niveau du ltre tactique .
7. Action au niveau du ltre stratégique .
8. Un ux est un ensemble de vols partageant les mêmes origine, destination et altitude de croisière.

la résolution de ce problème : la première consiste à ordonner les ux puis à leur affecter tour à tour une trajectoire la plus courte possible évitant les ux précédents à l'aide d'un algorithme $A^\star$, la seconde méthode utilise un algorithme génétique. L'algorithme génétique permet de construire des trajectoires sans con it pour les ux de vols dans l'espace aérien européen. Les résultats de cette étude pourraient cependant être améliorés en prenant en compte l'aspect temporel du problème. En effet, un ux dans lequel il y a très peu de vols n'est pas occupé pendant la totalité de l'intervalle de temps considéré, de sorte qu'une intersection en 3D ne représente pas toujours un con it réel.

### 2.2.4 Plani cation et régulation

Dans cette thèse, nous nous intéressons plus particulièrement aux mesures de régulation au niveau du « filtre pré-tactique », comme le sont actuellement les régulations ATFCM réalisées par la CFMU (voir section 1.2). Le principe de ce type de régulation est de modifier un (ou plusieurs) aspect(s) des plans de vol prévus pour les adapter à la structure de l'espace définie en phase stratégique et aux possibilités de man uvre pour la résolution tactique par les contrôleurs.

#### Allocation de créneaux de décollage

Plusieurs études ont eu pour objectif l'amélioration de l'allocation de créneaux de décollage effectuée par le logiciel CASA. Le problème avec le système actuel est que, lorsqu'un avion rate son créneau, le système doit lui trouver un nouveau créneau disponible, ce qui est difficile lors des heures de pointe. L'avion se voit alors attribuer un créneau induisant un fort retard. [Ferchaud 05] propose de réduire légèrement la capacité de chaque secteur de contrôle en réservant un certain nombre de créneaux, appelés « aires d'absorption », pour ce type de cas. Une analyse probabiliste est présentée, permettant de déterminer le nombre optimal d'aires d'absorption ainsi que leur placement chronologique. Les résultats de simulation sur un secteur de contrôle en-route montrent une diminution des retards et une augmentation globale du débit de trafic dans le secteur. Cependant, l'algorithme de CASA, dont nous avons mis en évidence les limites dans le chapitre précédent, n'est pas remis en cause. D'autre part, les simulations étant réalisées sur un unique secteur, les résultats présentés occultent certains problèmes globaux : par exemple, un avion qui rate son créneau pour un secteur donné se voit attribuer un nouveau créneau en fonction de la charge de celui-ci, sans prendre en compte la charge de trafic des autres secteurs qu'il doit traverser.

[Dalichampt 97] compare plusieurs stratégies d'attribution de créneaux de décollage (dont la stratégie « premier arrivé, premier servi » utilisée dans CASA) en utilisant la programmation par contraintes. Le retard moyen par vol régulé est ainsi réduit de 13 min avec CASA à 9 min avec la meilleure des stratégies testées. Cette méthode est cependant limitée à une affectation statique des créneaux de décollage. Elle ne permet donc pas de prendre en compte les modifications de plans de vol ou de capacité éventuelles dans un cadre opérationnel.

[Barnier 01b] utilise également la programmation par contraintes afin d'assurer le respect des contraintes de capacité des secteurs. Plusieurs modèles sont évalués : un modèle avec des fenêtres temporelles contiguës, un modèle avec fenêtre temporelle glissante et un modèle utilisant une contrainte de tri. Le premier modèle induit des pics de trafic en début de chaque fenêtre temporelle, rendant plus difficile la tâche du contrôleur. Les deux autres modèles permettent de « lisser » la charge de trafic dans le temps, au prix d'un coût en termes de retards un peu plus élevé.

Ces études permettent une amélioration significative de la régulation par allocation de créneaux de décollage par rapport au système CASA, mais elles restent fondées sur la notion de capacité des secteurs de contrôle, qui représente mal la charge de travail des contrôleurs (voir section 1.4.2).

En se fondant sur un réseau de routes directes, [Prot 10] tente de maximiser la densité de trafic en évitant les con its. Sur chaque « axe », composé d'une origine, d'une destination et d'un profil vertical de vol, sont répartis de manière régulière des points mobiles qui pourront être autorisés ou interdits afin d'éviter les con its. Il s'agit alors de répartir les avions sur ces points mobiles (qu'ils devront suivre précisément pendant le vol) de façon à maximiser l'utilisation du réseau tout en évitant les con its aux croisements des routes. Des bornes supérieure et inférieure sont données pour une mesure de la densité de trafic sur le réseau. Cependant, cette étude occulte le problème de la définition du réseau de routes initial ainsi que celui de la fréquence de distribution optimale des points mobiles sur chaque « axe ». La résolution de ces deux problèmes NP-difficiles est pourtant une étape nécessaire à la mise en uvre de ce modèle de planification.

### Allocation de niveaux de vol

Comme indiqué dans le chapitre 1, les mesures de régulation actuelles n'ont qu'une in uence sur les créneaux de décollage des vols. Il apparaît pourtant légitime de vouloir répartir les avions sur différents niveaux de vol afin de rendre le trafic moins dense aux altitudes les plus demandées par les compagnies aériennes.

Dans ce cadre, [Letrouit 98] étudie trois problèmes d'affectation de niveaux de vol pour des ux d'avions volant sur des routes directes :

- un problème de minimisation du nombre de niveaux de vol nécessaires afin que deux ux sécants soient sur des niveaux différents ;
- un problème de minimisation du nombre de croisements entre ux lorsqu'on dispose d'un nombre de niveaux de vol $N$ donné ;
- un problème de minimisation du nombre de changements de niveaux de vol lorsqu'on ne dispose que de deux niveaux, en évitant les intersections.

Ces problèmes se rapportent à des problèmes de coloration de graphe (ou de $k$-coloration) et des algorithmes présentés dans [Sahni 76, Goemans 94, Frieze 95] sont utilisés pour leur résolution.

[Barnier 02b] propose une approche similaire utilisant la programmation par contraintes comme méthode de résolution. Les expérimentations sont conduites sur du trafic simulé dans l'espace aérien français et montrent que 20 à 24 niveaux de vol sont nécessaires afin

d'éliminer tous les croisements entre ux de vols.

Ces résultats permettent de savoir combien de niveaux sont nécessaires pour éviter les con its entre ux, mais ne donnent pas de méthode pour affecter ces différents niveaux aux ux. Or, nous verrons (chapitre 5) que des vols effectués à des altitudes différentes induisent des consommations de carburant différentes. De plus, ces méthodes ne permettent pas de résoudre les con its au sein d'un ux de vols, contrairement à ce que nous essayons de faire dans notre étude.

[Abad 04] présente un modèle en programmation linéaire en nombres entiers mixte qui optimise un critère prenant en compte le retard de l'avion et sa consommation pendant le vol. Cependant, l'objectif est d'optimiser ce critère pour des vols partageant un même « couloir aérien » (les résultats sont donnés pour une portion de l'espace aérien nord-américain). L'évitement de con it n'est donc pas pris en compte dans ce modèle.

L'étude présentée dans [Constans 05b] utilise un algorithme génétique pour effectuer une allocation de niveaux de vol en phase pré-tactique. Chaque vol (dans l'espace aérien européen) se voit attribuer un niveau de vol parmi trois : son niveau optimal ou un niveau au-dessus ou au-dessous. L'implémentation proposée a permis de résoudre environ 60 % des con its en croisière pour les instances considérées.

### Plani cation de trajectoires 4D

Dans cette étude, nous proposons de combiner deux types de mesures de régulation : une allocation de niveaux de vol permettant de séparer les avions pour la phase stabilisée de leur vol, puis une allocation de créneaux de décollage pour la prise en compte des autres phases du vol.

L'allocation de créneaux de décollage ne prendra pas en compte le critère de capacité des secteurs, comme c'est le cas pour l'algorithme actuel de la CFMU, mais tentera d'éviter les con its potentiels entre les trajectoires prévues des avions. L'allocation préalable de niveaux de vol permettra de réduire la complexité du trafic avant la phase d'allocation de créneaux, afin de diminuer la quantité globale de retard attribuée. Une optimisation sera réalisée afin de réduire l'écart par rapport au niveau de vol demandé par les compagnies ainsi que les retards attribués lors de la seconde phase du processus.

Cette planification en deux phases n'aura donc aucune in uence sur la route choisie par les compagnies aériennes, ce qui assure que la trajectoire sera réalisable. Nous ferons cependant l'hypothèse d'un suivi précis des trajectoires 4D par les avions, ce qui n'est pas le cas avec les systèmes de vol actuels, mais fait partie des hypothèses de performances du programme SESAR.

# Deuxième partie

# Plani cation de trajectoires 4D : modèles

# Chapitre 3

# Détection de con it

La méthode de planification de trajectoires que nous proposons dans cette étude repose sur la résolution des con its en-route. La première étape de sa mise en uvre est donc la détection de ces con its, ainsi que leur représentation, qui doit être adaptée à la méthode de résolution.

Il existe de nombreuses manières d'effectuer la détection et la modélisation des con its aériens, selon la méthode de résolution utilisée, l'horizon temporel de la prévision ou le caractère continu ou discret du modèle. Cependant, les spécificités de la planification de trajectoires en quatre dimensions telle que nous la proposons dans cette étude nous ont conduit à modifier les modèles existants.

Nous proposons donc dans ce chapitre un nouveau modèle adapté à la détection et à la résolution de con its entre des trajectoires 4D. Ce modèle est défini pour des trajectoires échantillonnées avec un pas de temps fixe et permet une prise en compte fine de l'aspect temporel des con its. Ce modèle sera validé sur des trajectoires simulées, générées à partir de données réelles de trafic. La section 3.1 présente le simulateur de trafic utilisé pour construire les trajectoires 4D et la section 3.2 décrit le modèle utilisé pour la représentation des con its, ainsi que certains détails sur sa mise en uvre.

## 3.1 Simulation de tra c

Dans un domaine d'application tel que le contrôle ou la régulation du trafic aérien, la validation d'un modèle sur des « cas d'école » de petite taille n'est pas suffisante, du fait de la quantité de vols – environ 10 000 vols quotidiens en France en 2011 – et de leur disparité. Il est au contraire primordial de pouvoir confronter ce modèle à des données de trafic réalistes en termes de volume et de structure.

Beaucoup d'études dans le domaine de la gestion du trafic aérien utilisent des données réelles enregistrées par les radars de contrôle afin de valider leurs modèles. Bien que ces données permettent de travailler sur des trajectoires réalistes, leur usage n'est pas adapté à notre étude. En effet, l'objectif de celle-ci est de prévoir et résoudre les con its aériens. Or, sur des données issues de l'observation du trafic telles que les données radar, les con its ont

été résolus par les organismes du contrôle. Ces données ne peuvent donc pas être utilisées comme entrées de notre programme.

Afin de valider notre modèle, nous avons donc utilisé des trajectoires 4D *simulées* à partir de *données réelles* de trafic français et européen. Ces trajectoires simulées ont été fournies par l'outil *Complete Air Tra c Simulator (CATS)*, un simulateur de trafic en temps accéléré développé à la fin des années 90 au Centre d'Études de la Navigation Aérienne (CENA) [Alliot 97a]. Cet outil permet de fabriquer des trajectoires « réalistes » à partir de données de plans de vol en utilisant un modèle de vol tabulé (voir section 3.1.3). Cette section donne un aperçu du fonctionnement du simulateur CATS.

### 3.1.1 Principe général

Le simulateur CATS est un simulateur de trafic en-route [1] en temps accéléré, avec un modèle d'exécution discret : les positions et les vitesses des avions sont évaluées de manière itérative avec un pas de temps fixe. Le pas de temps de la simulation sera noté $\Delta t$ dans la suite de ce document.

Ce simulateur est écrit en langage OCaml [2] [Leroy 11], et la structure du noyau du système est légère, composée d'environ 5 000 lignes de code. En plus de la simulation de trajectoires, il possède deux modules de résolution de con its : l'un propose une résolution centralisée basée sur des algorithmes génétiques [Durand 97], l'autre une résolution embarquée basée sur un algorithme $A^\star$ [Granger 98].

### 3.1.2 Données d'entrée

Les données d'entrée du simulateur CATS sont principalement de deux natures : les données d'espace aérien et les données de plans de vol. Les données d'espace contiennent les positions des balises de navigation (les points de report) et des aéroports, ainsi que les données géographiques des secteurs de contrôle.

Les données de plans de vol sont issues du système de traitement des plans de vol et regroupent tous les plans de vols déposés pour une journée. Ces données sont au format COURAGE, dont on peut voir un exemple figure 3.1. Pour chaque vol sont compilées les données des plans de vol déposés, des plans de vol régulés par la CFMU, ainsi que les données pour les vols réalisés (ce sont principalement les heures de décollage qui changent entre ces trois types de données). Nous utiliserons dans nos expérimentations les données des plans de vol déposés, afin de valider notre modèle de régulation.

CATS propose deux modes de navigation pour la simulation : un mode standard où les avions suivent la route prévue par leur plan de vol et un mode direct où les avions suivent une route directe (voir section 1.1.2) de l'aéroport de départ vers l'aéroport d'arrivée. Ces deux modes seront utilisés pour la validation de notre modèle.

1. Le tra c est simulé depuis le sol, mais les trajectoires proches des aéroports (quelques NM) ne sont pas réalistes, car les procédures de départ et d'arrivée ne sont pas modélisées.
2. OCaml est un langage fonctionnel typé développé par l'INRIA depuis 1985 (`http://caml.inria.fr`).

```
20 AF101NR LFBO LFPO 9710 0 A320 0
21 255 280 490
22 O N N O
31 LFBO 32AW TOU FISTO DPRE PERIG DIBAG TUDRA BEVOL AMB CAD VAORL ODRAN PO
32 255 255 255 263 263 268 272 277 280 283 289 289 292 298
33 0 140 152 280 280 280 280 270 270 260 156 152 110 40
41 AW TA X1 R1 OG OY OT OA
42 236 253 253 256 257 257 270 270
43 260 261 262 268 277 283 284 300
```

FIGURE 3.1 – Extrait d'un fichier de plans de vols COURAGE. La ligne 20 donne des informations sur le vol : identifiant (AF101NR), origine (Toulouse LFBO), destination (Paris-Orly LFPO), modèle d'avion (A320) ; la ligne 31 décrit la route souhaitée (avec les niveaux de vol souhaités correspondant aux points de report sur la ligne 32) et la ligne 41 les secteurs de contrôle traversés.

### 3.1.3 Modèle de vol

Le calcul des trajectoires des avions à partir des plans de vol utilise les tables de performances *Base of Aircraft DAta (BADA)*, calculées à partir du modèle de performance à énergie totale d'Eurocontrol [Nuic 11]. Le tableau 3.1 est un extrait de la table de performances de l'Airbus A380. À chaque pas de temps, en fonction de l'altitude de l'avion et de la phase de vol (croisière, montée ou descente), la table de performances donne les paramètres de vitesse, de consommation de carburant et de taux de montée ou de descente à appliquer pour le calcul de la position suivante (voir exemple 2).

TABLE 3.1 – Extrait de la table de performances BADA tabulées pour l'Airbus A380 entre les niveaux de vol 260 et 390. Les vitesses (AS) sont exprimées en kt, les consommations de carburant en kg/min et les taux de montée (ROC) et de descente (ROD) en ft/min.

| | CRUISE | | | | CLIMB | | | | | DESCENT | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FL | AS | fuel | | | AS | ROC | | | fuel | AS | ROD | fuel |
| | | lo | nom | hi | | lo | nom | hi | nom | | nom | nom |
| 260 | 465 | 221.6 | 249.7 | 271.1 | 465 | 1950 | 1293 | 989 | 403.0 | 438 | 1931 | 41.7 |
| 280 | 479 | 222.1 | 250.9 | 272.8 | 479 | 1714 | 1098 | 809 | 381.6 | 452 | 1969 | 40.0 |
| 290 | 487 | 222.4 | 251.5 | 273.6 | 487 | 1593 | 999 | 717 | 370.8 | 459 | 1987 | 39.2 |
| 310 | 499 | 220.6 | 250.7 | 273.6 | 499 | 2013 | 1245 | 792 | 348.2 | 473 | 2024 | 37.4 |
| 330 | 494 | 206.9 | 239.9 | 264.9 | 494 | 1789 | 1022 | 576 | 321.3 | 488 | 1952 | 35.7 |
| 350 | 490 | 195.0 | 231.0 | 258.4 | 490 | 1747 | 777 | 337 | 294.6 | 490 | 2835 | 34.0 |
| 370 | 488 | 184.9 | 224.5 | 254.6 | 488 | 1312 | 462 | 70 | 268.6 | 488 | 2504 | 32.3 |
| 390 | 488 | 176.9 | 220.5 | 253.7 | 488 | 1024 | 200 | 0 | 243.2 | 488 | 2485 | 30.5 |

Source : Eurocontrol BADA

**Exemple 2 (Performances de l'Airbus A380)** *Si on se place au FL 290, la vitesse de l avion en croisière sera de 487 kt, et la consommation de carburant sera comprise entre 222,4 kg/min et 273,6 kg/min selon la masse de l appareil (la masse varie en fonction du taux de remplissage et de la quantité de carburant dans les réservoirs).*

Le modèle BADA tabulé n'est bien sûr pas aussi précis que le modèle à énergie totale dont il est dérivé. Cependant, il a été éprouvé depuis de nombreuses années et il offre un bon compromis, dans le cadre de simulations en temps accéléré, entre le réalisme des trajectoires générées et le temps de calcul nécessaire à cette génération.

### 3.1.4 In uence du pas de temps de simulation

Le choix de la valeur du pas de temps pour la simulation doit être judicieux : en effet, un pas de temps trop petit génère un volume de données trop important et avec une précision trop élevée par rapport à l'utilisation que l'on souhaite en faire, tandis qu'un pas de temps trop grand risque de mener à une détection incomplète des con its.

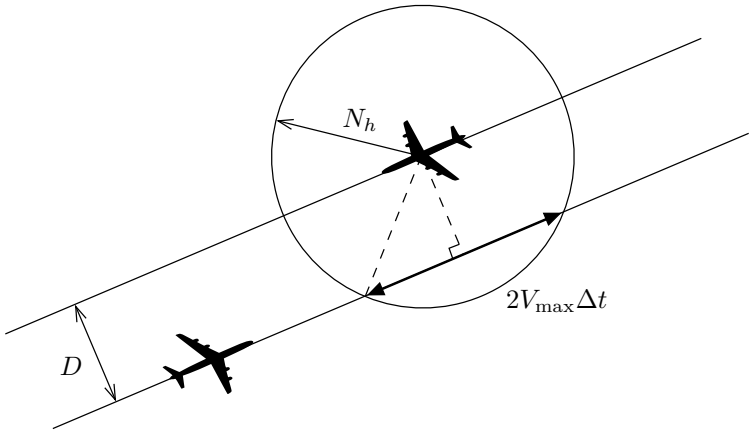



FIGURE 3.2 – In uence du pas de temps de simulation sur la détection de con it. La figure représente le cas critique pour la détection de con it.

La figure 3.2 montre le cas le plus défavorable pour la détection de con it, *i.e.* le con it le plus court possible : deux avions suivant des trajectoires parallèles séparées d'une distance $D \leq N_h$ ($N_h$ =5 NM étant la norme de séparation horizontale, voir définition 2 section 1.3), dans des directions opposées et à vitesse maximale $V_{\max}$. Dans cette situation, étant donné un pas de temps $\Delta t$ pour la discrétisation des trajectoires, nous avons d'après le théorème de Pythagore :

$$N_h^2 = D^2 + V_{\max}^2 \Delta t^2$$

Des con its non détectés peuvent donc survenir sous la condition :

$$\Delta t > \frac{\sqrt{N_h^2 - D^2}}{V_{\max}}$$

Pour un pas de temps $\Delta t$ donné, des faux négatifs pour la détection surviennent donc uniquement lorsque :

$$D > \sqrt{N_h^2 - V_{\max}^2 \Delta t^2}$$

Les avions présents dans les simulations réalisées ont tous une vitesse inférieure à $V_{\max} = 500\,kt$. De ce fait, en choisissant $\Delta t = 15\,\mathrm{s}$ pour l'échantillonnage des trajectoires, des faux négatifs ne sont possibles que pour $4{,}5\,\mathrm{NM} < D \leq 5\,\mathrm{NM}$, ce qui est communément accepté pour des simulations en temps accéléré puisque de telles situations ne sont pas autorisées par les procédures standards du contrôle aérien [3]. C'est cette valeur qui a été retenue pour l'ensemble des simulations.

## 3.2 Détection de con it

Un con it aérien est une situation dans laquelle deux avions sont séparés par une distance inférieure à la norme de séparation. Les différents filtres qui composent le contrôle du trafic aérien (voir chapitre 1) ont pour objectif d'éviter de telles situations. Actuellement, la détection de con its est faite par les contrôleurs, qui sont aidés par des outils visuels ainsi que des alarmes sonores en cas d'urgence. Mais dans le cadre de la mise en uvre de méthodes de contrôle ou de régulation reposant sur l'évitement des con its, comme c'est le cas dans notre étude, il est nécessaire de détecter les con its potentiels de manière automatique, à partir des données de trajectoires connues.

La section 3.2.1 présente un premier modèle utilisé par [Granger 02, Gourmel 07] pour la détection de con it, ainsi que les raisons qui ont conduit à son abandon. La section 3.2.2 propose un nouveau modèle de détection de con it 4D exact, adapté au problème de planification de trajectoires 4D (voir définition 3) et la section 3.2.3 présente une méthode de filtrage par boîtes englobantes permettant d'accélérer la détection des con its.

**Dé nition 3 (Trajectoire 4D)** *Une* trajectoire 4D *est une séquence de points décrivant les pro ls vertical, latéral et longitudinal du vol d un avion. Chaque point est décrit par trois coordonnées de position (e.g. latitude, longitude et altitude) et une coordonnée de temps.*

**Notation** Étant donné un vol $i$, on notera $\mathcal{T}_i = \left\{(p_i^k, t_i^k),\ k \in [1, n_i]\right\}$ sa trajectoire 4D, où les $p_i^k$ sont les points 3D, $t_i^k$ est l'instant de passage au point $p_i^k$ et $n_i$ est le nombre de points 4D qui composent la trajectoire.

3. Notamment, la *règle semi-circulaire* stipule que les avions volant vers l'est (cap compris entre 0 et 179 ) doivent utiliser un niveau de vol *impair* (*e.g.* FL 290, FL 310), tandis que les avions volant vers l'ouest (cap compris entre 180 et 359 ) doivent utiliser un niveau de vol *pair* (*e.g.* FL 280, FL 300). Dans certains pays, cette règle est identique à l'exception du fait que les ux sont discriminés entre nord et sud plutôt qu'entre est et ouest. Des exceptions à cette règle sont possibles, à l'initiative du contrôleur.

### 3.2.1 Zone de con it

Un con it aérien peut être vu comme une zone géographique dans laquelle deux avions ne doivent pas se trouver simultanément. La difficulté de ce modèle est de déterminer cette zone en fonction de la configuration du con it. Notre première approche, reprise des travaux de [Gourmel 07], a été de considérer un con it comme la clôture transitive de points en con it (voir définition 4).

**Dé nition 4 (Points en con it)** *Soient $i$ et $j$ deux vols distincts ($i < j$). Deux points $p_i^k$ et $p_j^l$ sont en con it si la norme de séparation entre $p_i^k$ et $p_j^l$ n est pas respectée, i.e. :*

$$d_h(p_i^k, p_j^l) < N_h \quad et \quad d_v(p_i^k, p_j^l) < N_v$$

*où $d_h$ est la fonction de distance dans le plan horizontal et $d_v$ la fonction de distance dans le plan vertical.*

*On note $\mathcal{C}(p_i^k, p_j^l)$ la relation de con it entre les points $p_i^k$ et $p_j^l$.*

[Granger 02] donne une caractérisation d'une telle zone de con it pour deux avions $i$ et $j$ volant à vitesses constantes $v_i$ et $v_j$ et dont les trajectoires convergent avec un angle $\alpha$. Ce type de con it est décrit par la longueur $L_{ij}$ de la portion de la trajectoire de l'avion $j$ qui est con ictuelle avec l'avion $i$ et donnée par la formule suivante :

$$L_{ij}(r, \alpha) = 2N_h \frac{\sqrt{1 - 2r\cos\alpha + r^2}}{\sin\alpha}, \quad \text{avec } r = \frac{v_j}{v_i}$$

pour $\alpha \neq 0 \bmod \pi$, et réciproquement par la longueur $L_{ji}$ de la portion de la trajectoire de l'avion $j$ qui est con ictuelle avec l'avion $i$ :

$$L_{ji}(r, \alpha) = L_{ij}\left(\frac{1}{r}, \alpha\right) = \frac{L_{ij}(r, \alpha)}{r}$$

Les portions de trajectoires de longueurs $L_{ij}$ et $L_{ji}$ déterminent la zone de con it entre les avions $i$ et $j$, comme illustré sur la figure 3.3. Un modèle simple consisterait alors à déterminer toutes ces zones de con it et considérer ceux-ci comme autant de ressources unitaires : un seul avion peut occuper une zone de con it à un instant donné.

Cependant, une telle description des con its ne permet pas de traiter tous les cas de manière satisfaisante. La figure 3.4 montre l'évolution de la taille de la zone de con it ainsi définie en fonction des valeurs de $r$ et de $\alpha$. On peut notamment observer que :

$$\begin{aligned} \lim_{\alpha \to 0 \bmod \pi} L(r, \alpha) &= +\infty \\ \lim_{r \to +\infty} L(r, \alpha) &= +\infty \end{aligned}$$

ce qui signifie que les con its en rattrapage ($\alpha \approx 0$), en face-à-face ($\alpha \approx \pi$) ou entre des avions ayant des vitesses très différentes donnent lieu à des zones de con it artificiellement grandes, pouvant même s'étendre à la totalité de la trajectoire. Dans un tel cas, si deux

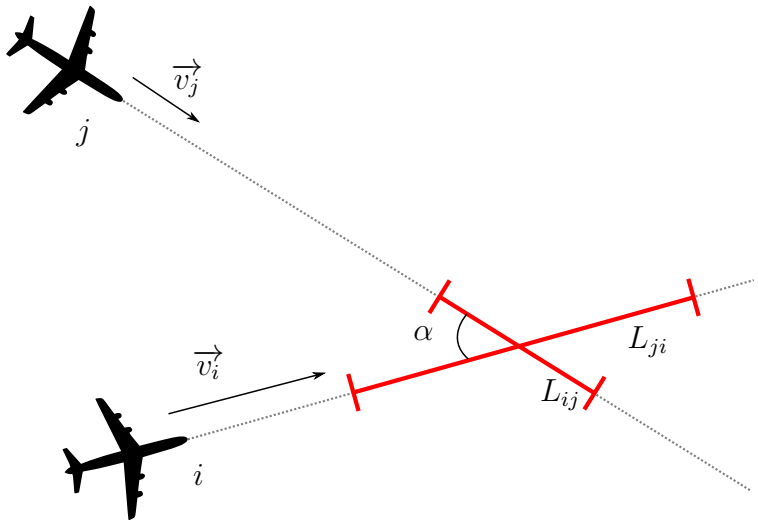


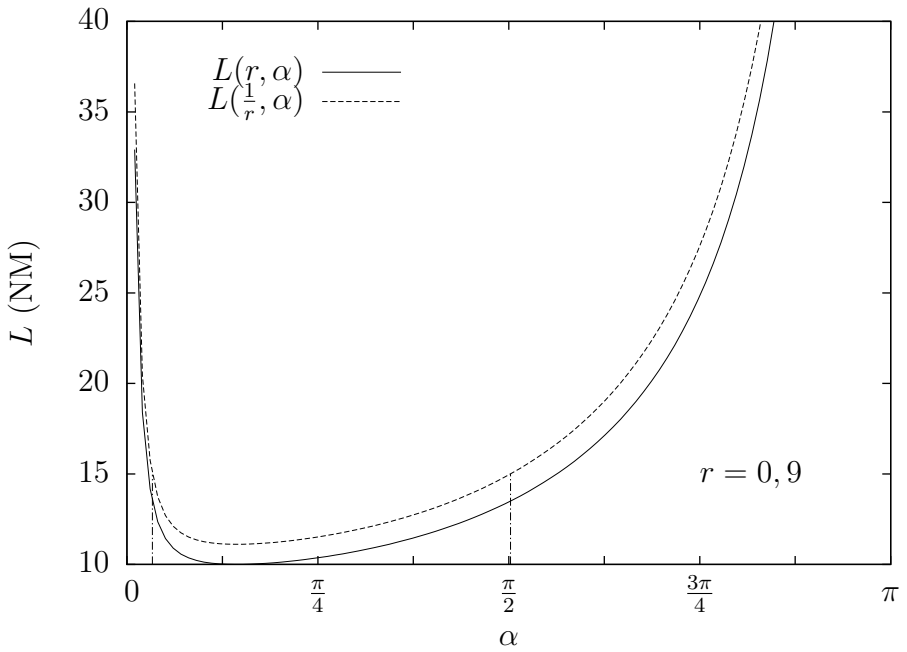

FIGURE 3.3 – Zone de con it entre deux avions à vitesses constantes.

vols suivent la même route, le deuxième ne peut pas décoller tant que le premier n'a pas atterri pour libérer la ressource, ce qui n'est pas acceptable d'un point de vue opérationnel. En effet, ce modèle ne tient pas compte du déplacement simultané des deux avions, de telle manière que tout un ensemble de points en « con it 3D » ne seraient pas réellement en con it. Ainsi, il suffirait que deux avions suivant la même route soient espacés de 5 NM pour éviter tout con it, alors que ce modèle interdit qu'ils soient en vol en même temps. En pratique, cette représentation des con its n'est pas trop pénalisante que pour des valeurs « modérées » de $\alpha$ et $r$ ($\alpha \in [5\ ,90\ ]$, $r \in [0,6;\ 1,5]$, voir figure 3.4). Au-delà de ces valeurs, les zones de con its sont artificiellement grandes (plus de 15 NM) et imposent donc des contraintes beaucoup trop fortes. Ces limites nous ont conduit à abandonner ce modèle au profit d'une méthode permettant de mieux prendre en compte l'aspect temporel de la détection de con it.

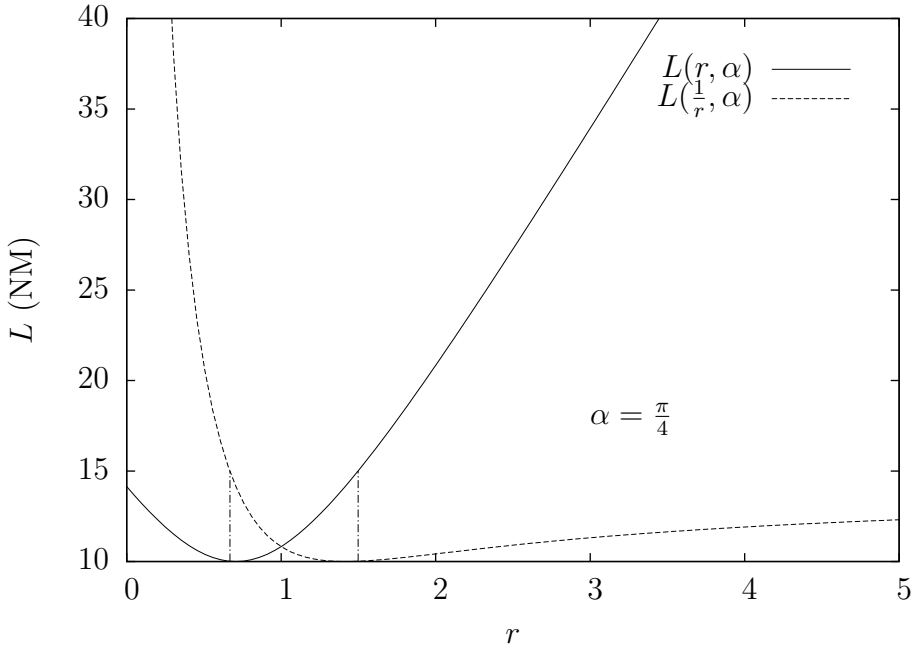
### 3.2.2 Détection de con its 4D

Afin de résoudre le problème rencontré avec le précédent modèle, nous proposons une méthode de détection de con its basée sur le concept de trajectoire 4D (voir définition 3). Ce nouveau modèle tient compte de l'aspect temporel du con it. Le principe est de parcourir l'ensemble des points, pour l'ensemble des trajectoires échantillonnées. Si deux points sont trop proches, c'est-à-dire s'ils ne respectent pas la norme de séparation, alors les instants de passage des deux avions sur ces points devront être différents. On retrouve ici la notion de ressource unitaire sous-jacente au modèle précédent, mais avec un niveau de granularité bien plus fin.

**Notation** La description de ce modèle nécessite l'introduction des quelques notations suivantes :




(a) Évolution de la taille de la zone de con it en fonction de l'angle de convergence des trajectoires pour $r = 0,9$.




(b) Évolution de la taille de la zone de con it en fonction des vitesses relatives des avions pour $\alpha = \frac{\pi}{4}$.

FIGURE 3.4 – Taille de la zone de con it en fonction de l'angle de convergence des trajectoires et des vitesses relatives.

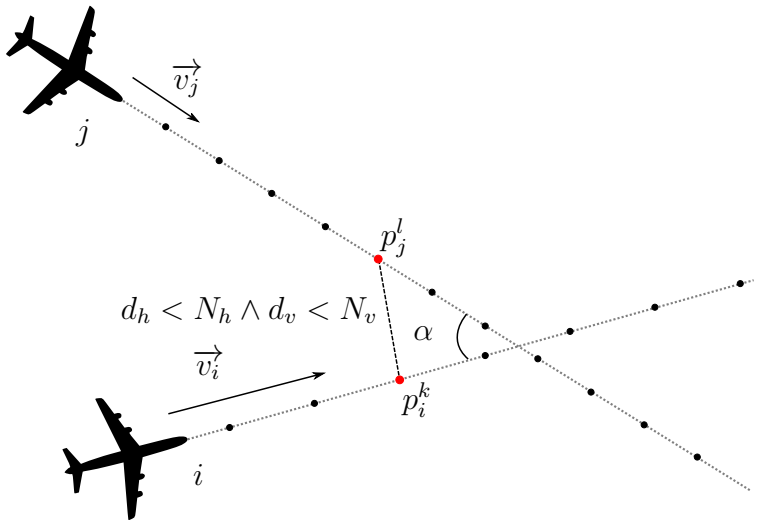



FIGURE 3.5 – Détection de points con ictuels.

- $\delta_i$ est le retard alloué au vol $i$ ;
- $\theta_i^k = t_i^k + \delta_i$ est le temps de passage effectif du vol $i$ au point $p_i^k$ ;
- $d_h$ et $d_v$ sont les fonctions de distance dans le plan horizontal et dans le plan vertical.

Pour chaque paire de trajectoires $i$ et $j$, la distance entre chaque paire de points est comparée à la norme de séparation afin de détecter les points de con it (voir définition 4), comme illustré sur la figure 3.5 dans le plan horizontal.

Étant donnés deux points $p_i^k$ et $p_j^l$ en con it, il faut assurer que les avions correspondants ne se trouvent pas en ces points au même instant. Cette contrainte temporelle se traduit par :

$$\neg\mathcal{C}(p_i^k, p_j^l) \Leftrightarrow \theta_i^k \neq \theta_j^l$$

que l'on peut également écrire :

$$\neg\mathcal{C}(p_i^k, p_j^l) \Leftrightarrow t_i^k + \delta_i \neq t_j^l + \delta_j$$

ou encore, en utilisant la notation $d_{ij} = \delta_j - \delta_i$ :

$$\neg\mathcal{C}(p_i^k, p_j^l) \Leftrightarrow d_{ij} \neq t_i^k - t_j^l \tag{3.1}$$

En débutant par le premier point $p_i^k$ en con it avec un point de la trajectoire du vol $j$, on peut considérer l'ensemble des points de la trajectoire de $j$ en con it avec $p_i^k$ :

$$\left\{p_j^l, \forall l \in [l_k, l_{k+r-1}]\right\} \tag{3.2}$$

où $r$ est le nombre de points de la trajectoire de $j$ en con it avec $p_i^k$. La contrainte temporelle implique donc :

$$d_{ij} \notin \left\{t_i^k - t_j^l, \forall l \in [l_k, l_{k+r-1}]\right\} \tag{3.3}$$

Les trajectoires considérées pour notre problème sont constituées de portions quasiment rectilignes [4] entre les différents points de reports du plan de vol correspondant. Les rayons de virage sont relativement élevés (*i.e.* il n'y a pas de grands changements de direction) car les avions suivent la route la plus courte possible pour rejoindre leur destination. De ce fait, les points de l'ensemble décrit dans la formule 3.2 sont successifs et cet ensemble est donc un segment de la trajectoire. On peut donc décrire la contrainte de l'équation 3.3 par :

$$d_{ij} \notin [lb_k, ub_k] \tag{3.4}$$

où $lb_k$ et $ub_k$ sont respectivement les bornes inférieure et supérieure de l'ensemble des $t_i^k - t_j^l$.

L'écriture de la contrainte sous la forme précédente n'est pas valable pour des trajectoires particulières, *e.g.* trajectoires d'hélicoptères, d'avions ravitailleurs (trajectoires circulaires) ou d'avions d'observation. Cependant, les trajectoires pour ces types d'appareils sont gérées par des procédures de contrôle particulières et sont séparées du reste du trafic. Nous conserverons donc l'écriture de la formule 3.4 pour la suite.

La détection du con it se poursuit en considérant le point suivant $p_i^{k+1}$ de la trajectoire de $i$. S'il est en con it avec un autre segment de la trajectoire de $j$, on obtient une autre contrainte pour $d_{ij}$ :

$$d_{ij} \notin [lb_{k+1}, ub_{k+1}]$$

toujours pour le même con it.

Sous les mêmes conditions sur la forme des trajectoires évoquées précédemment, les deux segments de la trajectoire du vol $j$ se chevauchent. Pour assurer la séparation des vols, il faut donc :

$$d_{ij} \notin [\min(lb_k, lb_{k+1}), \max(lb_k, lb_{k+1})]$$

En répétant cette opération, si on considère tous les points successifs de la trajectoire du vol $i$, en débutant à $p_i^k$, qui sont en con it avec des segments de la trajectoire du vol $j$, jusqu'à un point $p_i^{k+s}$, et si on note :

$$\underline{lb} = \min_{u \in [0,s]} (lb_{k+u}) \quad \text{et} \quad \overline{ub} = \max_{u \in [0,s]} (ub_{k+u})$$

les bornes inférieure et supérieure pour l'ensemble de ces segments, alors on peut définir le premier con it entre le vol $i$ et le vol $j$ par la contrainte :

$$d_{ij} \notin \left[\underline{lb}, \overline{ub}\right]$$

Deux vols peuvent avoir plusieurs con its disjoints, comme illustré sur la figure 3.6. Plusieurs intervalles disjoints peuvent donc être interdits pour la même variable $d_{ij}$. Pour deux vols $i$ et $j$ ayant $\sigma$ con its, on a donc :

$$d_{ij} \notin \left[\underline{lb}_1, \overline{ub}_1\right] \cup \cdots \cup \left[\underline{lb}_\sigma, \overline{ub}_\sigma\right]$$

4. Les avions suivent des portions d'*orthodromies*, une orthodromie étant le plus court chemin entre deux points sur une sphère. La projection d'une orthodromie sur un plan n'est pas rectiligne.

FIGURE 3.6 – Trois con its potentiels entre deux vols. Le temps (en minutes) est représenté par un dégradé de gris.

**Notation** Ces intervalles de valeurs interdites constituent donc une représentation des con its entre deux vols. Pour deux vols $i$ et $j$ distincts, on notera ce con it :

$$\mathcal{C}_{ij} = \left[\underline{lb}_1, \overline{ub}_1\right] \cup \cdots \cup \left[\underline{lb}_\sigma, \overline{ub}_\sigma\right] \tag{3.5}$$

Cette modélisation des con its est exacte, contrairement à celle de la section 3.2.1, car il y a équivalence entre $d_{ij} \in \left[\underline{lb}_1, \overline{ub}_1\right] \cup \cdots \cup \left[\underline{lb}_\sigma, \overline{ub}_\sigma\right]$ et une situation de con it réel.

Dans la formule 3.5, $\sigma$ dépend de $i$ et de $j$ et devrait donc être noté $\sigma_{ij}$. Cependant, nous n'utiliserons pas cet indice afin de ne pas alourdir la notation.

Contrairement au modèle précédent, la détection de con it est dans ce cas antisymétrique. Plus précisément, pour un con it $\mathcal{C}_{ij}$ défini par l'équation 3.5, on a :

$$\mathcal{C}_{ji} = \left[-\overline{ub}_\sigma, -\underline{lb}_\sigma\right] \cup \cdots \cup \left[-\overline{ub}_1, -\underline{lb}_1\right]$$

Nous profitons de cette propriété pour n'effectuer la détection de con it que pour les paires ordonnées de vols, c'est-à-dire les vols $i, j$ tels que $i < j$.

Malgré cela, si on note $n$ le nombre de vols et $p_{\max}$ le nombre maximum de points par trajectoire, la complexité de la méthode décrite ci-dessus est en $\mathcal{O}(n^2 p_{\max}^2)$. Pour une instance typique sur l'espace aérien français, les ordres de grandeur sont $n = 9000$ vols, $p_{\max} = 1000$ avec une moyenne de $p_{moyen} = 250$ points par trajectoire, soit environ $n^2 p_{moyen}^2 = 5 \cdot 10^{12}$ calculs de distance entre points en moyenne. Ce nombre est encore plus important lorsqu'on considère l'espace aérien européen : environ $n = 25000$ vols, $p_{\max} = 3000$ avec une moyenne de 500 points par trajectoire pour plus de $10^{14}$ calculs de distances. La section suivante propose un filtrage permettant de réduire le nombre d'opérations à effectuer lors de la détection de con it.

### 3.2.3 Boîtes englobantes

Afin de réduire la complexité en moyenne de la détection de con it, un mécanisme de filtrage par *boîtes englobantes* est mis en place. Le principe est d'encapsuler chaque segment de trajectoire dans une « boîte » et d'effectuer la détection décrite dans la section 3.2.2 uniquement pour les boîtes sécantes.

**Dé nition 5 (Segment de trajectoire)** *Soit $i$ un vol, et $T_i = \left\{p_i^k, \forall k \in [1, n]\right\}$ sa trajectoire échantillonnée. On dira qu une sous-séquence contiguë $S$ de $T_i$ est un* segment *de la trajectoire de $i$ si le cap de l avion est constant lors du parcours de cette sous-séquence de points, c est-à-dire :*

$$S = \left\{p_i^k, \forall k \in [k_1, k_2], k_1 < k_2 \in [1, n]^2\right\} \text{ segment de } T_i$$

$$\text{si } k_2 - k_1 \leq 1 \text{ ou } \forall k \in [k_1 + 1, k_2 - 1], \left|\widehat{p_i^{k-1} p_i^k p_i^{k+1}}\right| < \varepsilon$$

*où $\widehat{p_i^{k-1} p_i^k p_i^{k+1}}$ est l angle formé par les projections des points $p_i^{k-1}, p_i^k$ et $p_i^{k+1}$ sur le plan horizontal et $\varepsilon$ une valeur positive proche de zéro correspondant à la tolérance pour la détection des segments.*

Selon la définition 5, chaque trajectoire $T_i$ est découpée en segments successifs $S_i = \{S_i^1, S_i^2, \ldots, S_i^m\}$ de telle manière que $S_i$ réalise une partition de $T_i$. Chaque segment est ensuite encapsulé dans un volume parallélépipédique, de telle manière que chaque point du segment soit distant des parois de ce volume englobant d'au moins la moitié de la norme de séparation. On obtient ainsi un ensemble de boîtes $B_i = \{B_i^1, B_i^2, \ldots, B_i^m\}$, où $B_i^k$ est la boîte englobante du segment $S_i^k$. La figure 3.7 illustre le principe des boîtes englobantes dans le plan horizontal. La complexité de ce découpage est en $\theta(p)$ si $p$ est le nombre de points de la trajectoire et donne lieu à une dizaine de boîtes par trajectoires en moyenne (sur l'espace aérien français).

L'amplitude verticale de la boîte englobante d'un segment $S$ est l'intervalle :

$$\left[\min_{p \in S} altitude(p) - \frac{N_v}{2}, \max_{p \in S} altitude(p) + \frac{N_v}{2}\right]$$

Considérons deux vols $i$ et $j$, dont les trajectoires respectives sont découpées en segments $\{S_i^1, S_i^2, \ldots, S_i^n\}$ et $\{S_j^1, S_j^2, \ldots, S_j^m\}$, et encapsulées dans les boîtes englobantes $\{B_i^1, B_i^2, \ldots, B_i^n\}$ et $\{B_j^1, B_j^2, \ldots, B_j^m\}$. Si seules les deux boîtes $B_i^k$ et $B_j^l$ sont sécantes, alors la détection de points de con its décrite section 3.2.2 n'est réalisée qu'entre les points des segments $S_i^k$ et $S_j^l$, évitant ainsi de nombreux tests inutiles. Par extension, si plusieurs boîtes sont sécantes, la détection sera réalisée sur les segments correspondants.

Cette méthode de filtrage a permis de réduire de manière drastique le temps nécessaire à la détection de con it dans nos expérimentations, comme on peut le voir sur la figure 3.8 pour du trafic dans l'espace français, avec un gain supérieur à 95 % dans la quasi totalité des instances générées. Ainsi, une instance est générée en 15 min environ contre plus de six heures sans filtrage.

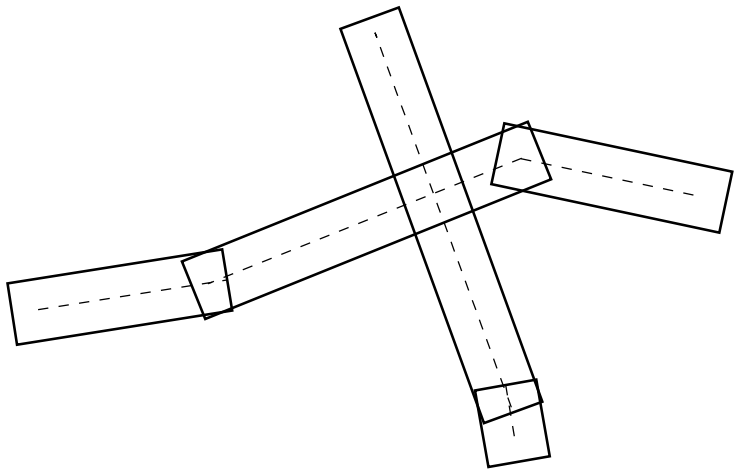


FIGURE 3.7 – Deux trajectoires sécantes et leurs boîtes englobantes. La détection de points en con it n'est faite que pour les points situés dans les boîtes sécantes.

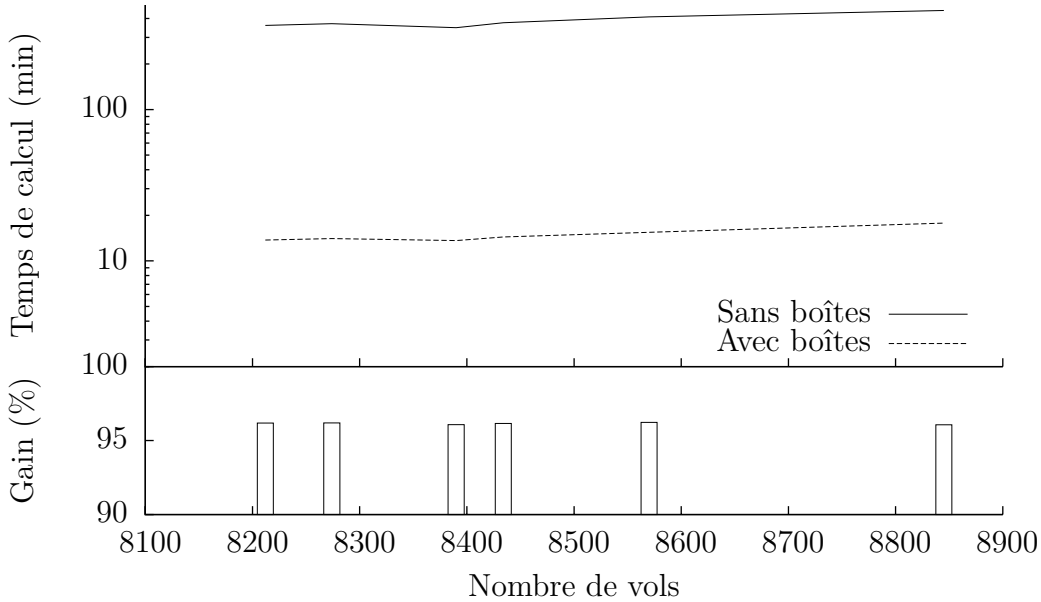



FIGURE 3.8 – Comparaison des temps de détection des con its avec et sans utilisation de boîtes englobantes. Les temps (en minutes) sont représentés sur une échelle logarithmique.

D'autres méthodes ont déjà été testées pour déterminer les intersections entre trajectoires. [Becker 10] notamment utilise un algorithme de type *sweep line* [de Berg 98] pour déterminer les intersections entre des trajectoires composées de segments de droite. Ce type d'algorithme pourrait être utilisé pour accélerer la détection des intersections entre boîtes englobantes.

# Chapitre 4

# Allocation de créneaux de décollage sans con it

Les méthodes actuelles de régulation du trafic aérien par allocation de créneaux de décollage sont basées sur des mesures macroscopiques : les capacités des secteurs de contrôle (voir section 1.2). Ce modèle a l'avantage d'être relativement efficace [1] en conditions opérationnelles, mais présente certaines limites évoquées précédemment : la capacité des secteurs de contrôle n'est pas suffisamment représentative de la charge de travail des contrôleurs et l'algorithme d'allocation de créneaux est une heuristique gloutonne, incomplète et inconsistante pour ce problème.

Le projet Episode 3 [Graham 06], coordonné par la Commission Européenne, a pour objectif de valider les concepts opérationnels proposés dans le cadre du programme de recherche SESAR, et notamment le concept de trajectoire 4D (définition 3). Dans ce cadre, nous proposons un nouveau modèle pour l'allocation de créneaux de décollage, qui prend directement en compte chaque con it comme contrainte d'un algorithme d'optimisation. Cette méthode consiste à détecter avec suffisamment d'anticipation les con its qui pourraient survenir à l'aide de la méthode décrite dans le chapitre 3 et les résoudre en retardant les vols impliqués dans ces con its. Ce modèle sera validé sur des trajectoires simulées à partir de données réelles de trafic, grâce au simulateur CATS décrit dans la section 3.1.

## 4.1 Hypothèses sur la précision des trajectoires

La détection de con it se base sur le concept de trajectoire 4D décrit par la définition 3. La difficulté sous-jacente à ce concept est sa mise en opération. En effet, les systèmes de vol actuels – les *Flight Management Systems (FMS)* – ne permettent pas d'effectuer un suivi précis d'une trajectoire 4D. Le suivi géographique est bon, mais la précision temporelle est pour sa part limitée à une seule contrainte appelée *Estimated Time of Arrival (ETA)* d'une

1. Comme nous l'avons évoqué dans le chapitre 1, on observe des écarts parfois importants entre les schémas d'ouverture de secteurs et les secteurs réellement ouverts. Il y a en fait, le jour J, des négociations entre la CFMU et les centres de contrôle.

précision de 30 s pour la phase en-route, avec un horizon de 30 minutes. Cependant, les objectifs de précision du suivi de trajectoire des deux principaux programmes de recherche mondiaux en ATM, SESAR en Europe et *NextGen* aux États-Unis, sont particulièrement ambitieux. Notamment, [SESAR 07] se base pour l'horizon 2020 sur un suivi de trajectoire 4D avec les précisions suivantes :

- suivi latéral à $\pm 0,3$ NM ;
- suivi vertical à $\pm 150$ ft ;
- suivi longitudinal : plusieurs contraintes temporelles à $\pm 10$ s.

Le WG-85 de l'*European Organisation for Civil Aviation Equipment (EUROCAE)*[2], quant à lui, fixe les objectifs de suivi longitudinal pour les systèmes de vol à $\pm 30$ s pour les procédures en-route, et $\pm 10$ s pour les procédures d'approche.

Nous utiliserons ces spécifications comme hypothèses de travail. Étant donné le pas d'échantillonnage choisi pour les trajectoires ($\Delta t = 15$ s), cela correspond à un suivi quasi parfait des trajectoires 4D.

Dans un cadre plus réaliste, ces précisions ne sont pas atteignables et on peut observer plusieurs types d'incertitudes : certaines sont relativement faibles (incertitudes sur les vitesses horizontales notamment), d'autres sont plus importantes (*e.g.* incertitudes sur les vitesses verticales, contraintes opérationnelles non prévues, conditions météorologiques). Il est également difficile de connaître avec précision l'heure de décollage d'un avion tant que celui-ci n'a pas décollé. Afin de prendre en compte cette contrainte, nous proposons dans la section 4.6 une méthode dynamique permettant de mettre à jour le problème au cours de sa résolution. Nous proposons également une modification du modèle permettant de prendre en compte une partie des incertitudes sur le suivi des trajectoires.

## 4.2 Variables

Les variables de décision du modèle sont les retards des vols. Pour un vol $i$ donné, le retard $\delta_i$ prend sa valeur dans l'intervalle $[0, \delta_{\max}]$, où $\delta_{\max}$ est la valeur du retard maximal autorisé lors de l'allocation de créneau de décollage. $\delta_{\max}$ est un paramètre du modèle.

Comme l'indique le modèle de con it décrit dans la section 3.2, nous utiliserons des variables auxiliaires $d_{ij}$ définies par : $d_{ij} = \delta_j - \delta_i$. Ces variables prendront donc leurs valeurs dans l'intervalle $[-\delta_{\max}, \delta_{\max}]$ et hors des $\mathcal{C}_{ij}$ définis par l'équation 3.5.

À partir de ces ensembles de variables et du modèle de con it donné par l'équation 3.5, on peut décrire formellement le modèle retenu pour le problème de résolution de con its par allocation de créneaux de décollage pour une journée de trafic dans un espace aérien donné :

**Dé nition 6 (Problème d'allocation de créneaux sans con it)** *Pour un ensemble de $n$ vols, si on autorise un retard maximal de $\delta_{max}$, le problème d allocation de créneau de*

2. Ce groupe de travail réunit des constructeurs aéronautiques, fournisseurs de systèmes de vol et organismes de contrôle aérien européens a n de spéci er les standards de navigation en 4D permettant le support des futurs besoins des systèmes ATM.

*décollage peut être décrit par le couple* $(\Delta, \mathcal{C})$*, où :*

$$\begin{aligned} \Delta &= \{\delta_i, i \in [1, n]\} \\ \mathcal{C} &= \bigwedge_{i,j \in [1,n]^2} (\delta_j - \delta_i) \notin \mathcal{C}_{ij} \end{aligned}$$

*où* $\Delta$ *est l ensemble des variables de décision, avec* $\forall \delta \in \Delta$*,* $\delta$ *prend ses valeurs dans l intervalle* $[0, \delta_{\max}]$ *et* $\mathcal{C}$ *est l ensemble des contraintes issues de la détection de con it.*

Comme mentionné dans la section 3.2.2, la matrice des con its est antisymétrique et on peut donc se contenter de :

$$\bigwedge_{i<j \in [1,n]^2} d_{ij} \notin \mathcal{C}_{ij}$$

## 4.3 Traitement des données

La granularité fine du modèle proposé nécessite le traitement d'une grande quantité de données. Certains filtrages peuvent être mis en uvre afin de réduire la taille des instances et donc le temps d'exécution et la consommation mémoire du programme qui résoud le problème. On peut principalement filtrer le trafic par niveaux de vol, ce qui correspond à filtrer par type de vol (vols commerciaux ou vols de tourisme, qui évoluent à des altitudes différentes) ou par tranche horaire, afin de ne traiter que les périodes les plus denses par exemple. Cette section présente les différents filtres mis en uvre.

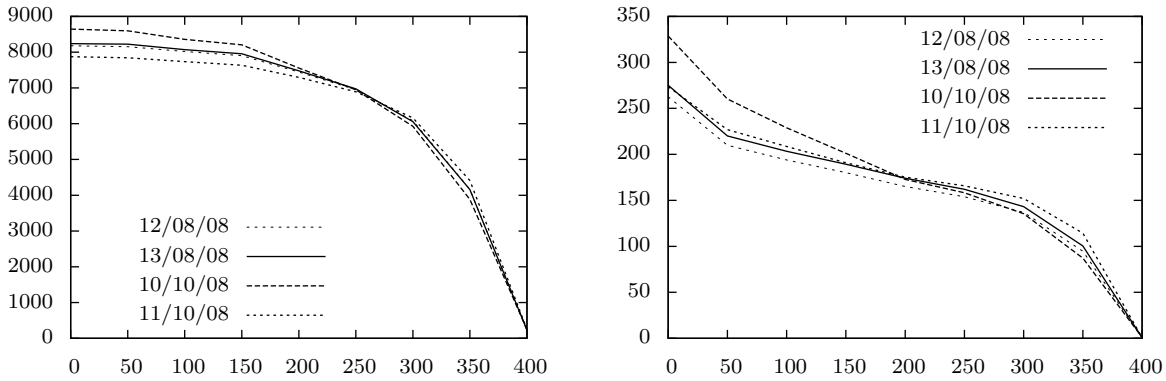
### 4.3.1 Filtrage des trajectoires

Comme nous l'avons déjà évoqué, les trajectoires de départ et d'arrivée dans les zones proches des aéroports suivent des procédures particulières qui ne sont pas implantées dans le simulateur de trafic utilisé. D'autre part, les méthodes de contrôle dans ces zones diffèrent des méthodes utilisées lors de la phase en-route. Pour ces raisons, nous avons décidé de ne pas prendre en compte ces portions de trajectoire. Toutes les trajectoires sont donc tronquées à l'intérieur d'un cylindre de 10 NM autour de leurs aéroports de départ et de destination, sinon on aurait artificiellement de nombreux con its pour les vols qui partent du même aéroport.

D'autre part, afin de ne considérer que les vols commerciaux, il est possible de filtrer les données dans une certaine « tranche » de l'espace, c'est-à-dire uniquement les points dont le niveau de vol est dans un intervalle $[FL_{\min}, FL_{\max}]$ donné. Dans ce cas, tous les con its situés entièrement au-dessous du niveau de vol $FL_{\min}$ ou entièrement au-dessus du niveau de vol $FL_{\max}$ sont supprimés de l'instance. Dans la pratique, ce type de filtrage est essentiellement appliqué sur un intervalle du type $[FL_{\min}, +\infty]$ pour prendre en compte la majeure partie du trafic commercial, notamment au-dessus du FL 290.

La figure 4.1 montre l'in uence d'un tel filtrage[3] sur le nombre de vols et sur le nombre de con its potentiels pour quelques journées de trafic dans l'espace aérien français. On

3. Les vols sans aucun con it sont également supprimés de l'instance (cf. section 4.3.3).




(a) Nombre de vols en fonction de $FL_{\min}$.

(b) Nombre de con its (milliers) en fonction de $FL_{\min}$.

FIGURE 4.1 – In uence du filtrage en niveau de vol sur la taille du problème (filtrage en fonction du niveau de vol minimum uniquement, *i.e.* $FL_{\max} = +\infty$) pour quatre journées de trafic dans l'espace aérien français.

remarque notamment que la plupart des vols se situent au-dessus du FL 250. Cela est dû au fait que la plupart des avions utilisés pour les vols réguliers évoluent à des altitudes élevées (*e.g.* FL 330 pour un Airbus A320). Ce qui se passe en-dessous du FL 250 sera de moindre importance pour notre étude, cette tranche sera filtrée si nécessaire pour réduire la taille de l'instance.

Il est également possible d'effectuer un filtrage temporel, en ne considérant que les portions de trajectoires dont les temps de passage sont dans un intervalle $[t_{\min}, t_{\max} + \delta_{\max}]$. Nous utiliserons notamment ce filtrage afin de mettre en uvre une résolution par fenêtre glissante (voir section 4.6.2).

### 4.3.2 Filtrage des con its

Le retard maximal autorisé $\delta_{\max}$ permet de supprimer de l'instance une partie des con its. En effet, en reprenant la définition des con its (voir équation 3.5), on a : $\mathcal{C}_{ij} = [\underline{lb}_1, \overline{ub}_1] \cup \ldots \cup [\underline{lb}_\sigma, \overline{ub}_\sigma]$ et les contraintes sur les variables $d_{ij}$ imposent : $d_{ij} \in [-\delta_{max}, \delta_{max}]$ et $d_{ij} \notin \mathcal{C}_{ij}$, c'est-à-dire :

$$d_{ij} \notin \mathcal{C}_{ij} \cap [-\delta_{\max}, \delta_{\max}]$$

Tous les con its tels que $\overline{ub}_k < -\delta_{\max}$ ou $\underline{lb}_k > \delta_{\max}$, $k \in [1, \sigma]$, peuvent donc être supprimés de l'instance du problème.

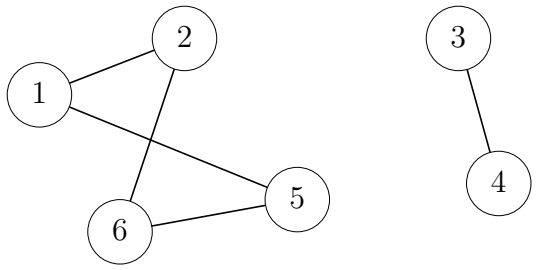



FIGURE 4.2 – Découpage en clusters : les problèmes contenant les avions $\{1, 2, 5, 6\}$ et $\{3, 4\}$ respectivement, peuvent être résolus de manière indépendante.

### 4.3.3 Filtrage des vols

Une fois le filtrage des con its effectué, tous les vols sans con it sont retirés de l'instance. Un vol $i$ est sans con it pour une instance donnée si :

$$\forall j \in [1, n] \setminus \{i\},\ \mathcal{C}_{ij} \cap [-\delta_{\max}, \delta_{\max}] = \emptyset$$

D'autre part, les vols dont l'aéroport de départ est situé hors de la zone CEAC (voir figure 1.5) ne sont pas régulés par la CFMU. Afin de prendre en compte cette réalité opérationnelle, nous avons fait le choix de ne pas allouer de retard à ces vols. Ceux-ci sont donc des contraintes pour les autres vols, qui devront les éviter. Si des con its sont détectés entre deux tels vols, ils sont comptés mais ne sont pas intégrés à l'instance du problème. Ces con its devront être résolus par d'autres méthodes, manuelles ou automatiques, mais ces résolutions ne rentrent pas dans le cadre de cette étude.

### 4.3.4 Découpage du problème

Lors de la résolution de con its aériens, il est possible de découper le problème en *clusters* de vols (définition 7), qui sont autant de sous-problèmes indépendants, afin de faciliter le calcul d'une solution.

**Dé nition 7 (Cluster)** *On appelle* cluster *de vols en con it la clôture transitive de la relation est en con it avec . L ensemble des clusters correspond donc aux classes d équivalence de cette relation.*

Si on représente les données sous la forme d'un graphe dont les n uds représentent les vols et les arêtes les con its entre ces vols (voir figure 4.2), alors les clusters de vols en con it sont les composantes connexes de ce graphe. Il est possible de détecter ces composantes connexes, puis de séparer le problème principal en autant de sous-problèmes indépendants, plus faciles à résoudre.

En pratique, cependant, chaque vol est en con it avec de nombreux autres vols, comme illustré sur la figure 4.3, avec certains vols pouvant avoir des con its avec plus de 600 autres vols (pour $\delta_{max} = 90\,\text{min}$ sur la journée entière). La clôture transitive donne alors lieu à

une composante connexe principale, regroupant la quasi-totalité des vols de la journée de trafic considérée. Le découpage en clusters n'a donc pas été mis en uvre pour les résultats présentés dans la suite.

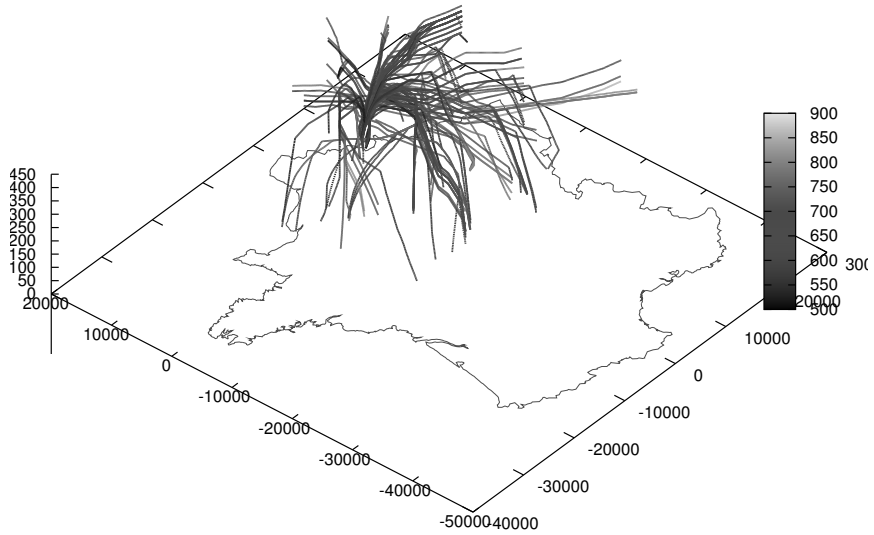



FIGURE 4.3 – Un vol en con it avec plusieurs autres. Le temps (en minutes) est représenté par un dégradé de gris.

## 4.4 Optimisation

Lors du processus d'allocation de créneaux de décollage, nous définissons le coût d'une solution en fonction de la quantité de retard allouée par rapport aux heures de décollage prévues. Nous utiliserons deux fonctions de coût différentes, qui correspondent à deux objectifs distincts : équité de l'allocation de créneaux vis-à-vis des compagnies aériennes ou performance globale du système.

### 4.4.1 Équité entre utilisateurs de l'espace

Afin de favoriser l'équité entre les compagnies aériennes et de permettre une calibration du système d'allocation de créneaux de décollage (*i.e.* en phase de faisabilité de la méthode, pour définir les retards maximaux nécessaires à la résolution de tous les con its potentiels dans un espace donné), nous définissons une première fonction de coût qui correspond au retard maximum alloué lors de la résolution de con its :

$$\text{cost}_{\max} = \max_{i \in [1,n]} \delta_i$$

Cette fonction de coût sera utilisée pour la résolution en programmation par contraintes car elle permet une meilleure propagation et donc de meilleures performances (voir chapitre 7) qu'avec un critère comme la moyenne des retards (voir section suivante). Cette fonction nous a également permis d'atteindre des preuves d'optimalité (ou d'échec) pour la quasi totalité des instances traitées.

L'inconvénient de cette fonction de coût est qu'elle donne peu d'indications sur la quantité totale de retard allouée : on sait seulement que celle-ci est bornée par $n \times \text{cost}_{\text{max}}$. Il faudra donc tenter de maintenir cette quantité à une valeur faible par d'autre moyens (*e.g.* utilisation d'heuristiques adaptées).

### 4.4.2 Performance globale du système

Afin de mesurer la performance du système, la fonction de coût doit prendre en compte la totalité des retards alloués lors du processus. La fonction de coût retenue pour cet objectif est la somme des retards :

$$\text{cost}_{\text{sum}} = \sum_{i \in [1,n]} \delta_i$$

C'est cette fonction de coût qui a été retenue pour la résolution à l'aide de métaheuristiques. D'autres fonctions ont été testées avec moins de succès (mauvaise convergence de l'algorithme) et ont donc été écartées. Ce fut notamment le cas de la somme des carrés des retards, qui permet non seulement de mesurer la performance globale du système, mais également de pénaliser les retards importants.

Il serait également possible de prendre en compte le coût en utilisant les chiffres de [Cook 04] (si on note $c_m = 72\,€/\text{min}$) : $\text{cost}_€ = c_m \times \text{cost}_{\text{sum}}$. Cependant, cela n'a aucune in uence sur l'optimisation et il faudrait un modèle économique plus précis (analyse par type d'appareil, par type de mission...) pour avoir une fonction de coût pertinente.

## 4.5 In uence du retard maximal autorisé

Le retard maximal autorisé n'est pas seulement un paramètre de l'algorithme de recherche, il a également une in uence sur la détection de con its. En effet, l'augmentation du retard maximal autorisé entraîne une augmentation du nombre de con its potentiels et a donc une in uence directe sur la taille du problème. En poursuivant ce raisonnement : si un retard de 24 h est autorisé, alors la détection de con it peut se faire sans prendre en compte la dimension temporelle, puisque le croisement dans l'espace de deux trajectoires produirait une contrainte.

De ce fait, si on obtient une preuve d'inconsistance sur une instance donnée du problème, il est nécessaire de générer une nouvelle instance avec un retard maximal autorisé plus important, détectant ainsi les con its potentiels avec un horizon temporel plus grand, donc en augmentant la taille du problème.

## 4.6 Modélisation des incertitudes

Les hypothèses de précision présentées dans la section 4.1 ne sont pas réalistes dans le contexte actuel et restent ambitieuses à l'horizon 2020 retenu par le projet SESAR. Afin de rendre notre modèle plus réaliste, nous avons mis en uvre deux méthodes permettant de prendre en compte des incertitudes, principalement sur le suivi longitudinal des trajectoires 4D.

### 4.6.1 Prise en compte des incertitudes sur les heures de décollage

Afin de rendre robustes les solutions de l'allocation de créneaux de décollage, nous proposons d'étendre chaque intervalle de valeurs interdites d'une quantité *ext* donnée, cette extension étant faite de manière symétrique sur chacun de ces intervalles. Cette approche est similaire à celle proposée par [Lauderdale 11] où la norme de séparation horizontale est augmentée artificiellement de manière à créer une « zone tampon » prenant en compte les incertitudes sur le suivi de trajectoire. La représentation des con its donnée par l'équation 3.5 est alors modifiée pour devenir :

$$\mathcal{C}_{ij} = \left[\underline{lb}_1 - \frac{ext}{2}, \overline{ub}_1 + \frac{ext}{2}\right] \cup \cdots \cup \left[\underline{lb}_\sigma - \frac{ext}{2}, \overline{ub}_\sigma + \frac{ext}{2}\right]$$

Une telle extension permet donc de prendre en compte une incertitude de $\pm\frac{ext}{2}$ sur les heures de décollage. Elle peut également permettre de réduire les effets d'autres sources d'incertitudes, notamment sur les vitesses horizontales des avions. Cependant, cette extension se fait au détriment du coût des solutions : le problème étant plus contraint, les solutions obtenues ont un coût bien plus élevé (voir section 8.2.5). Afin de réduire le coût de la prise en compte des incertitudes, nous proposons dans la section suivante une méthode de résolution dynamique.

### 4.6.2 Résolution dynamique avec fenêtre glissante

L'heure de décollage d'un avion ne peut être connue avec exactitude qu'une fois que celui-ci a effectivement décollé. Bien entendu, plus on se rapproche de l'heure prévue de décollage, plus l'incertitude sur celle-ci est faible. Compte tenu de ce phénomène, il paraît inutile d'effectuer la prévision de con its et l'allocation de créneaux de décollage avec un horizon d'une journée, puisqu'on traiterait alors des con its qui n'auront probablement pas lieu.

Afin de prendre en compte de manière dynamique les incertitudes sur les heures de décollage, nous avons mis en place un mécanisme de résolution par *fenêtre glissante*. Ce mécanisme consiste à effectuer la résolution avec un horizon $H_r$ donné, puis à décaler cet horizon d'une quantité $\gamma \ll H_r$ pour la résolution suivante tout en conservant les décisions prises pour les vols qui doivent décoller dans les $\gamma$ prochaines minutes, et ce jusqu'à atteindre la fin de la journée considérée. La figure 4.4 illustre ce principe. Le vol $i$ a une heure de décollage prévue $\tau_i$. Lors de la résolution de la première fenêtre, il se voit attribuer un

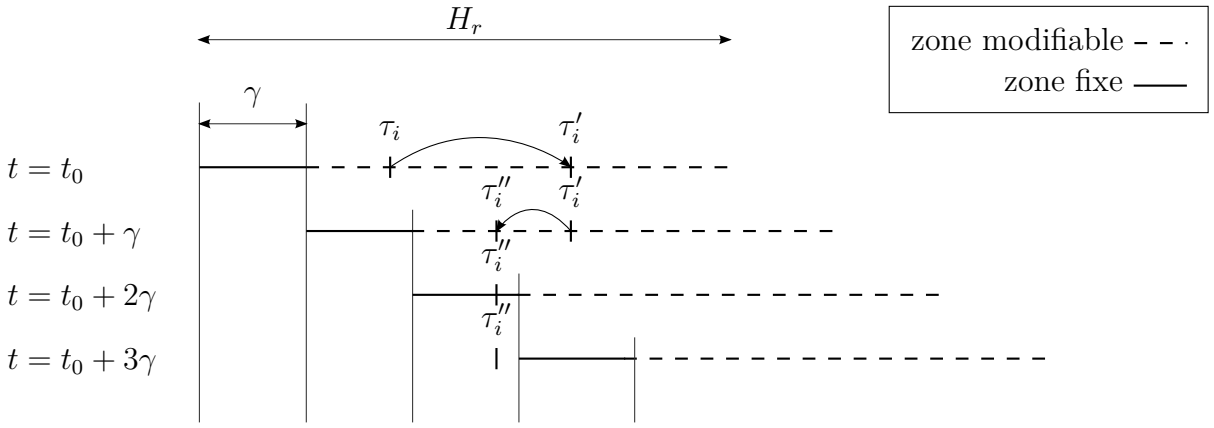



FIGURE 4.4 – Résolution par fenêtre glissante. $\tau_i$ est l'heure prévue pour le décollage du vol $i$, $\tau_i'$ et $\tau_i''$ sont les heures de décollage modifiées du vol $i$ après résolutions de la première et de la deuxième fenêtres respectivement.

retard de sorte que son décollage est désormais prévu à $\tau_i'$. Lors de la résolution de la fenêtre suivante, son retard peut être diminué, donnant lieu à un décollage à $\tau_i''$. Lorsqu'on résoud la troisième fenêtre, l'heure de décollage de ce vol ne peut plus être modifiée, c'est donc cette heure qui doit être transmise à l'avion.

Cette méthode de résolution par fenêtre glissante permet également de traiter en plusieurs « morceaux » un problème dont la taille est trop grande pour être traité en une fois.

### Résolution par fenêtre glissante

À chaque étape de la résolution, on considère le sous-problème constitué des vols dont le décollage est prévu dans les $H_r$ prochaines minutes (en comptant un possible retard de $\delta_{\max}$). On effectue la résolution de ce sous-problème et on fixe les retards pour les vols dont le décollage (éventuellement retardé) doit s'effectuer dans les $\gamma$ prochaines minutes. Puis on décale la fenêtre d'observation de $\gamma$ minutes et on réitère le procédé. L'algorithme 4.1 donne une description plus formelle de la résolution par fenêtre glissante.

L'ensemble $\mathcal{P}_t$ est le sous-problème à considérer au pas de temps $t$ et $\mathcal{S}_t$ est une solution de ce sous-problème. $\mathcal{F}_t$ est l'ensemble des vols dont le retard sera fixé pour la suite de la résolution : ces vols deviennent des contraintes pour les résolutions suivantes. Les retards alloués à ces vols sont fixés et donc intégrés à la solution $\mathcal{S}$ du problème.

La valeur de l'horizon de résolution a une in uence directe sur la taille du sous-problème à résoudre à chaque pas de temps : plus $H_r$ est grand, plus le sous-problème sera difficile à traiter. En revanche, un horizon de résolution important permet de trouver une meilleure solution globale, puisqu'à chaque pas de temps on a une plus grande connaissance du problème complet. Un horizon important permet également d'éviter « l'effet d'horizon », c'est-à-dire tomber dans une instance sans solution car tous les problèmes de la fenêtre précédente ont été repoussés au-delà de la fenêtre. Cela se fait au prix d'un temps d'exécution plus élevé pour la résolution. Il est par ailleurs inutile, dans un cadre plus réaliste, de choisir

**Entrées :** $\mathcal{V}_0 = [1, n]$ un ensemble de vols, avec $\tau_i$ l'heure de décollage prévue pour le vol $i$, *alloc* une fonction d'allocation de créneaux de décollage sans con it qui prend en paramètres les vols à considérer ainsi que la solution partielle comme contrainte.
**Sorties :** solution au problème d'allocation de créneaux de décollage sur la journée.

1: $\mathcal{V} \leftarrow \mathcal{V}_0$
2: $\mathcal{S} \leftarrow \emptyset$
3: $t \leftarrow 0$
4: **tant que** $\mathcal{V} \neq \emptyset$ **faire**
5: $\quad \mathcal{P}_t = \{i \in \mathcal{V} \mid \tau_i \in [t, t + H_r + \delta_{\max}]\}$
6: $\quad \mathcal{S}_t = alloc(\mathcal{P}_t, \mathcal{S})$
7: $\quad \mathcal{F}_t = \{i, i \in \mathcal{V} \mid \tau_i + \delta_i \in [t, t + \gamma]\}$
8: $\quad \mathcal{S} \leftarrow \mathcal{S} \cup (\mathcal{S}_t \cap \{(i, \delta_i), i \in \mathcal{F}_t\})$
9: $\quad \mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{F}_t$
10: $\quad t \leftarrow t + \gamma$
11: **n tant que**
12: **renvoyer** $\mathcal{S}$

**Algorithme 4.1:** Résolution par fenêtre glissante

un horizon de résolution important, car les heures de décollage prévues sont trop incertaines à quelques heures d'avance, et beaucoup de résolutions risquent d'être remises en cause à plusieurs reprises. Nous avons choisi un horizon de résolution de deux heures pour nos résolutions en fenêtre glissante ; cette valeur fournit un bon compromis entre le temps de calcul pour chaque situation et l'anticipation des futurs con its.

Le paramètre $\gamma$ correspond à la réactivité du système : plus sa valeur est faible, plus le système sera réactif, au prix d'un nombre de résolutions plus élevé. Une valeur trop grande de $\gamma$ nuit à la résolution. En effet, dans le cas extrême où $\gamma = H_r$, on risque de retarder une grande partie des vols de telle manière qu'ils se retrouvent dans la « tranche » suivante, rendant ainsi sa résolution bien plus difficile, voire impossible. La valeur de $\gamma$ ne doit pas non plus être trop faible : si on sort du cadre des simulations pour se placer dans un cadre opérationnel, il faut que la résolution puisse s'effectuer dans un temps inférieur à $\gamma$, afin que les retards alloués puissent être communiqués aux avions à temps. Pour les simulations présentées dans ce document, nous avons utilisé des valeurs pour $\gamma$ comprises entre 5 et 15 minutes pour des temps de résolution de l'ordre d'une minute.

### Fenêtre glissante et incertitudes

Afin de prendre en compte les incertitudes sur les heures de décollage de manière dynamique, nous modifions légèrement la procédure précédente. Lors du traitement de la fenêtre temporelle commencant à l'instant $t$, nous introduisons une incertitude $\varepsilon_i$ sur l'heure de décollage des vols $i$ qui auraient dû décoller dans l'intervalle de temps $[t - \gamma, t]$. Plusieurs cas sont possibles :

- $\tau_i + \delta_i + \varepsilon_i < t$ : l'avion a décollé et devient une contrainte pour la suite de la résolution,

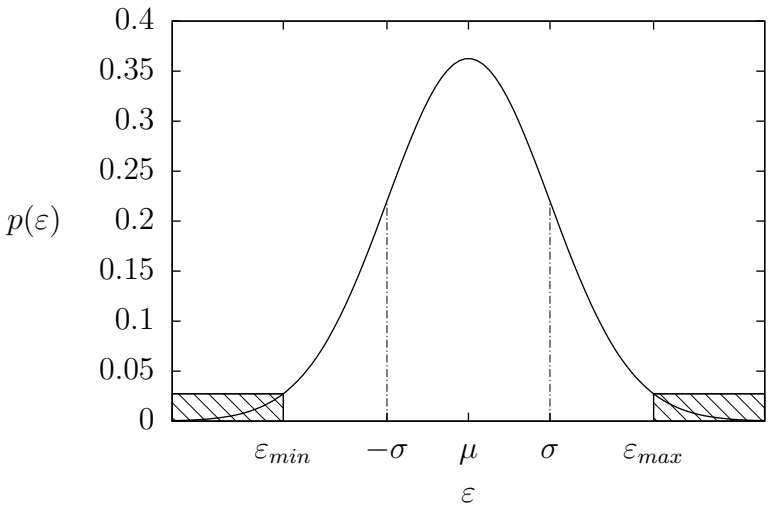



FIGURE 4.5 – Densité de probabilité de l'incertitude $\varepsilon$ sur les heures de décollage.

comme dans le cas de la fenêtre glissante sans incertitude.

- $\tau_i + \delta_i + \varepsilon_i \in [t, t+\gamma]$ : l'avion n'a pas encore décollé, mais se retrouve dans la zone non modifiable de la fenêtre et devient donc également une contrainte pour la suite de la résolution. On estime que la probabilité pour que l'avion décolle dans l'intervalle $[t, t+\gamma]$ est élevée.
- $\tau_i + \delta_i + \varepsilon_i > t + \gamma$ : le vol se retrouve à nouveau dans la zone modifiable de la fenêtre temporelle, il est donc repris en compte pour la résolution en cours.

Avec cette méthode, il est tout à fait possible que deux vols qui ont été séparés se retrouvent à nouveau en con it l'un avec l'autre, puisqu'une fois l'incertitude appliquée sur l'heure de décollage, l'algorithme de résolution n'a plus la possibilité de modifier celle-ci. Ces con its seront comptés lors des résolutions ; ils devront être pris en compte par les extensions des con its présentées dans la section 4.6.1, par allocation de niveaux de vol (voir chapitre 5) ou résolus par d'autres méthodes, de la même manière que les con its entre vols non européens (voir section 4.3.3).

À chaque étape, un vol $i$ donné a une probabilité $p_r$ d'avoir une incertitude sur son heure de décollage. Si c'est le cas, celle-ci, notée $\varepsilon_i$, est tirée aléatoirement selon une distribution de probabilité de type gaussienne tronquée, de moyenne $\mu$ et d'écart type $\sigma$ et bornée dans un intervalle $[\varepsilon_{\min}, \varepsilon_{\max}]$ afin de ne pas tirer un retard trop important (supérieur à 24 heures par exemple). La figure 4.5 montre la densité de probabilité de l'incertitude $\varepsilon_i$. L'intérêt de ce modèle est que l'on peut passer continûment d'une instance sans incertitude ($p_r = 0$) à une instance où tous les vols ont une incertitude, tirée selon une distribution gaussienne tronquée paramétrable ($p_r = 1$).

L'algorithme 4.2 est une modification de l'algorithme 4.1 de résolution par fenêtre glissante prenant en compte les incertitudes décrites dans cette section. À chaque étape et pour chaque vol $i$, on note $p_i$ la variable aléatoire qui indique si le vol $i$ doit être bruité ($p_i = 1$) ou non ($p_i = 0$), avec : $P(p_i = 1) = p_r$ et $P(p_i = 0) = 1 - p_r$.

**Entrées :** $\mathcal{V}_0 = [1, n]$ un ensemble de vols, avec $\tau_i$ l'heure de décollage prévue pour le vol $i$, *alloc* une fonction d'allocation de créneaux de décollage sans con it qui prend en paramètres les vols à considérer, la solution partielle comme contrainte ainsi que les incertitudes sur les heures de décollage de vols précédents et $\mathcal{GT}$ une fonction qui renvoie une valeur selon une loi gaussienne tronquée.

**Sorties :** solution au problème d'allocation de créneaux de décollage sur la journée.

1: $\mathcal{V} \leftarrow \mathcal{V}_0$
2: $\mathcal{S} \leftarrow \emptyset$
3: $\mathcal{I} \leftarrow \emptyset$
4: $t \leftarrow 0$
5: **tant que** $\mathcal{V} \neq \emptyset$ **faire**
6: $\quad \mathcal{P}_t = \{i \in \mathcal{V} \mid \tau_i \in [t, t + H_r + \delta_{\max}]\}$
7: $\quad \mathcal{S}_t = alloc(\mathcal{P}_t, \mathcal{S}, \mathcal{I})$
8: $\quad \mathcal{F}_t = \{i, i \in \mathcal{V} \mid \tau_i + \delta_i \in [t, t + \gamma]\}$
9: $\quad$ **pour tout** $i \in \mathcal{F}_t$ **faire**
10: $\quad\quad p_i$ t.q. $P(p_i = 1) = p_r$ // probabilité que le vol $i$ soit bruité
11: $\quad\quad$ **si** $p_i = 1$ **alors**
12: $\quad\quad\quad \varepsilon_i = \mathcal{GT}(\mu, \sigma, \varepsilon_{\min}, \varepsilon_{\max})$
13: $\quad\quad\quad$ **si** $\tau_i + \delta_i + \varepsilon_i > t + \gamma$ **alors**
14: $\quad\quad\quad\quad \mathcal{F}_t \leftarrow \mathcal{F}_t \setminus \{i\}$
15: $\quad\quad\quad$ **sinon**
16: $\quad\quad\quad\quad \mathcal{I} \leftarrow \mathcal{I} \cup \{(i, \varepsilon_i)\}$
17: $\quad\quad\quad$ **n si**
18: $\quad\quad$ **n si**
19: $\quad$ **n pour**
20: $\quad \mathcal{S} \leftarrow \mathcal{S} \cup (\mathcal{S}_t \cap \{(i, \delta_i), i \in \mathcal{F}_t\})$
21: $\quad \mathcal{V} \leftarrow \mathcal{V} \setminus \mathcal{F}_t$
22: $\quad t \leftarrow t + \gamma$
23: **n tant que**
24: **renvoyer** $\mathcal{S}$

**Algorithme 4.2:** Résolution par fenêtre glissante avec prise en compte des incertitudes sur les heures de décollage

Outre les incertitudes sur les heures de décollage, cette technique permet de prendre en compte d'autres contraintes opérationnelles, notamment les annulations de vols.

# Chapitre 5

# Allocation de niveaux de vol

Dans le système de régulation tactique actuel, quasiment aucune optimisation n'est effectuée sur les niveaux de vol [1]. Un niveau de vol, correspondant à une consommation minimale de carburant pour la trajectoire considérée, est demandé par la compagnie aérienne pour chacun de ses vols, et celui-ci est accepté par le contrôle. Cependant, du fait des similitudes entre les différents avions qui constituent la  otte des compagnies aériennes, certains niveaux de vol sont particulièrement demandés, ce qui génère une charge de trafic particulièrement dense. Or les régulations tactiques actuelles (gérées par la CFMU) ou futures (comme celle présentée dans le chapitre précédent) prennent en compte la complexité du trafic afin d'attribuer les créneaux de décollage. Un trafic dense mènera ainsi à des retards importants.

Dans le cadre de l'affectation de créneaux de décollage présentée dans le chapitre 4, la mesure importante est le nombre de con  its potentiels entre les vols. L'intérêt d'une meilleure distribution des vols sur les différents niveaux de vol possibles réside dans le fait que deux avions volant sur deux niveaux de vol différents sont séparés au sens de la définition d'un con  it aérien [2], ramenant ainsi la situation à plusieurs « tranches » de trafic à gérer, indépendantes les unes des autres si on exclut les avions en phase évolutive. La figure 5.1 montre la répartition des con  its sur une journée de trafic dans l'espace aérien français. On remarque notamment que seuls 30 à 35 % des con  its ont lieu en phase de croisière, les autres con  its ne pourront pas être résolus par allocation de niveaux de vol.

Cependant, une allocation de niveaux de vol doit tenir compte de contraintes de coût : un avion évoluant à une altitude différente de son altitude optimale consomme plus de carburant et rejette plus de $CO_2$ dans l'atmosphère. Il convient donc de distribuer les niveaux de vol de telle manière qu'un vol se voie attribuer un niveau proche de son altitude optimale. La section 5.1 donne une idée de ce coût supplémentaire. La section 5.2 présente le modèle retenu pour l'allocation de niveaux de vol. Nous proposons dans la section 5.3 une première approche pour laquelle les vols sont regroupés en  ux avant l'allocation, puis dans la section 5.4 une approche pour laquelle chaque vol est considéré de manière indépendante.

1. Seule la règle semi-circulaire évoquée dans le chapitre 3 a une incidence sur les niveaux de vol.
2. Les niveaux de vol utilisables sont séparés les uns des autres de 1 000 ft, soit la valeur $N_v$ de la norme de séparation verticale.

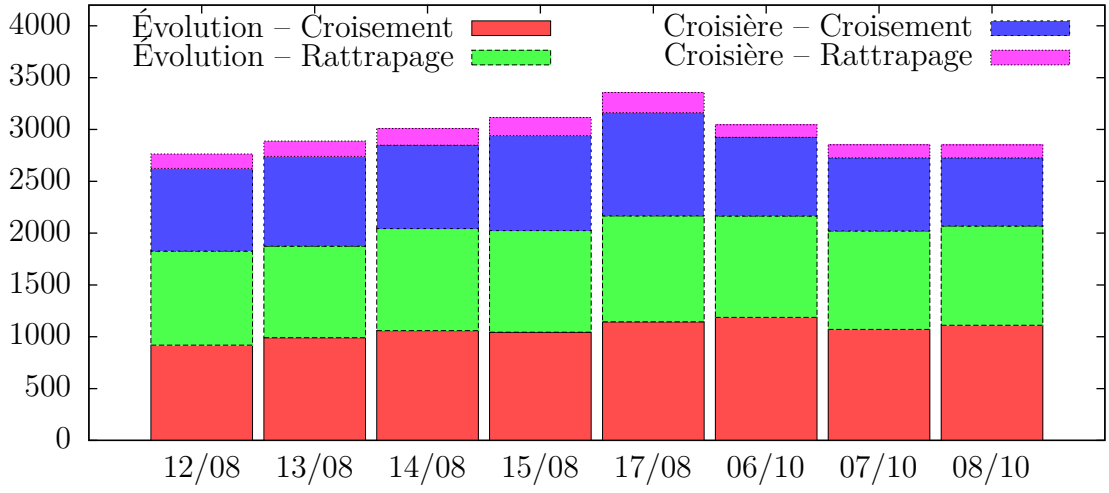



FIGURE 5.1 – Répartition des différents types de con its pour quelques journées de trafic en août et octobre 2008 dans l'espace aérien français.

## 5.1 Coût d'un changement de niveau de vol

Lors du dépôt d'un plan de vol, le demandeur (*e.g.* une compagnie aérienne) doit préciser à quel niveau de vol il souhaite évoluer lors de la phase de croisière. Ce niveau de vol souhaité – le *Requested Flight Level (RFL)* – correspond la plupart du temps à l'altitude optimale de croisière. En effet, pour chaque type d'appareil et pour chaque configuration (nombre de passagers, distance à parcourir...), on peut déterminer l'altitude à laquelle la consommation de carburant sera minimale. Le tableau 5.1, dont les données sont issues de [Airbus 04], donne la réduction du rayon d'action de différents appareils en fonction de l'écart à l'altitude optimale. Cette réduction du rayon d'action correspond grossièrement à l'augmentation de la consommation de carburant.

TABLE 5.1 – Réduction du rayon d'action en fonction de la différence d'altitude par rapport à l'altitude optimale.

| Avion | +2 000 ft | −2 000 ft | −4 000 ft | −6 000 ft |
|---|---|---|---|---|
| A300-600 | 2,0 % | 0,9 % | 3,4 % | 9,3 % |
| A310 | 1,9 % | 1,4 % | 4,4 % | 9,3 % |
| A320 | 1,4 % | 2,1 % | 6,2 % | 12,0 % |
| A330-200 | 1,8 % | 1,3 % | 4,2 % | 8,4 % |
| A330-300 | 3,0 % | 1,0 % | 3,2 % | 7,2 % |
| A340-200 | 1,4 % | 1,5 % | 4,0 % | 8,0 % |
| A340-300 | 1,5 % | 1,6 % | 5,2 % | 9,5 % |

Source : Airbus

Les impacts observés sont très dépendants du type d'avion et il semble difficile de

déduire de ces données une tendance générale. De manière simpliste, on observe qu'une différence inférieure à 2 000 ft, dans un sens ou dans l'autre, dégrade peu les performances (moins de 2 % dans la plupart des cas). En revanche, des écarts plus importants impliquent des dégradations significatives. Il est donc nécessaire de maintenir l'écart entre le RFL et le niveau de vol alloué à une valeur la plus faible possible, afin de diminuer les coûts induits par un vol à une altitude non optimale.

## 5.2 Modèle

L'allocation de niveau de vol consiste à attribuer à chaque avion un niveau de vol pour sa phase de croisière, de telle manière que :

- le nombre de croisements entre avions évoluant au même niveau de vol soit minimal ;
- pour chaque vol, l'écart entre le niveau de vol alloué et le RFL soit le plus faible possible.

### 5.2.1 Variables

L'ensemble $\mathcal{L}$ des variables du problème d'allocation de niveaux de vol pour un ensemble $\mathcal{V}$ de vols se définit par :

$$\mathcal{L} = \{FL_i, i \in \mathcal{V}\}$$

où $FL_i$ correspond au niveau de vol du vol $i$. Les valeurs des variables $FL_i$ correspondent aux niveaux de vol utilisables (tous les 1 000 ft) et sont donc des entiers [3] multiples de 10.

Comme nous l'avons signalé dans la section précédente, il y a un coût significatif à ne pas attribuer son niveau de vol optimal à un vol. Afin de tenir compte de cette contrainte, nous introduisons un paramètre $\Delta_{\max}$ qui correspond à l'écart maximum par rapport au RFL. Si on note $RFL_i$ le RFL du vol $i$, alors :

$$\forall i \in \mathcal{V}, FL_i \in [RFL_i - \Delta_{\max}, RFL_i + \Delta_{\max}]$$

Dans la réalité, tous les niveaux de vol ne sont pas atteignables par tous les types d'avions. En effet, chaque avion a une altitude maximale qui dépend de ses performances et il est donc impossible d'attribuer un niveau de vol au-dessus de cette altitude. Il est très facile d'intégrer ce type de contraintes au modèle, en restreignant les valeurs possibles des variables :

$$\forall i \in \mathcal{V}, FL_i \in [RFL_i - \Delta_{\max}, \min(FL_i^{\max}, RFL_i + \Delta_{\max})]$$

où $FL_i^{\max}$ est le niveau de vol maximum que l'on puisse attribuer au vol $i$ (*i.e.* le niveau de vol le plus élevé au-dessous de son altitude maximale), mais leur définition requiert des données de performances et de masse difficiles à obtenir auprès des constructeurs et des compagnies aériennes.

3. On pourrait également imposer le respect de la règle semi-circulaire en contraignant les variables à être de la forme $2n$ pour les vols évoluant vers l'ouest et $2n+1$ pour les vols évoluant vers l'est. Cependant, il n'est pas particulièrement intéressant de maintenir cette règle dans le cadre d'une plani cation sans con it telle que présentée dans cette thèse.

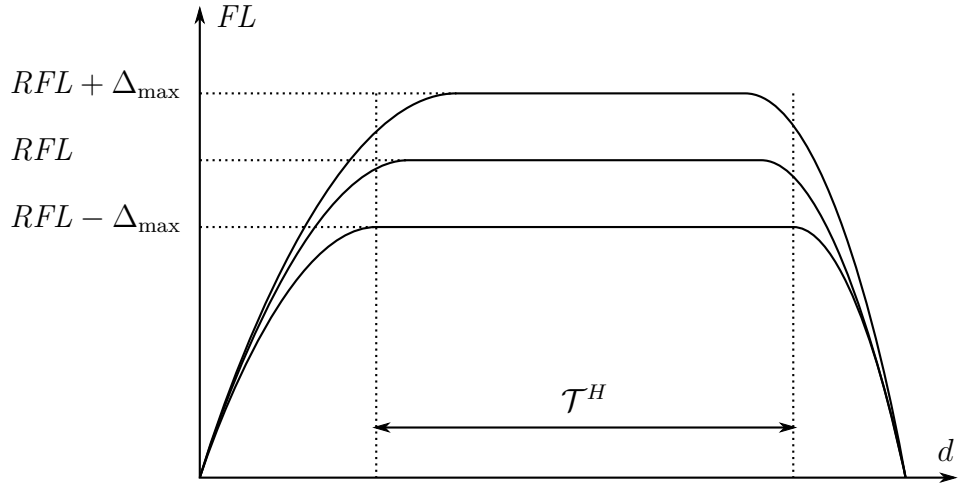



FIGURE 5.2 – Plus longue phase de croisière possible pour un vol. Les trajectoires représentées correspondent à trois niveaux de vol différents pour un même vol. $d$ est la distance (dans le plan horizontal) parcourue depuis le départ.

### 5.2.2 Filtrage

L'allocation de niveaux de vol ne sépare les vols que pour leur phase de croisière. De ce fait, toutes les portions de montée et de descente des trajectoires peuvent être tronquées, puisque les croisements que l'on pourrait détecter dans ces zones ne pourraient de toute façon pas être évités par cette méthode.

Pour un vol $i$, on peut ainsi définir une trajectoire restreinte $\mathcal{T}_i^H \subset \mathcal{T}_i$ pour l'allocation de niveaux de vol :

$$\mathcal{T}_i^H = \left\{(p_i^k, t_i^k) \in \mathcal{T}_i, altitude(p_i^k) \geq RFL_i - \Delta_{\max}\right\}$$

Cette trajectoire correspond à la plus longue phase de croisière possible, c'est-à-dire la phase de croisière dans le cas où le niveau de vol attribué est le plus bas possible, comme illustré sur la figure 5.2.

Partant de ces trajectoires, deux modèles ont été envisagés : l'un avec agrégation des vols en ux et attribution de niveaux de vol à ces ux, l'autre avec attribution des niveaux de vol directement aux vols. Le premier modèle se justifie d'un point de vue opérationnel, en se basant sur le principe d'allouer un « tube » pour chaque paire origine-destination. Le second modèle ne prend pas en compte ce type de contrainte, ce qui lui donne une plus grande exibilité. Ces deux modèles sont présentés dans les sections suivantes.

## 5.3 Allocation par ux

Notre première approche du problème d'allocation de niveaux de vol est similaire à celles de [Letrouit 98] et [Barnier 02b] : les vols évoluant sur une même route et à la même

altitude se voient attribuer le même niveau de vol, cette attribution étant faite de manière à ce que deux ux de vols sécants évoluent à des niveaux différents.

### 5.3.1 Agrégation des vols en ux

La première étape de cette méthode consiste à agréger les vols en ux. Pour cela, on définit tout d'abord l'ensemble $\mathcal{R}$ des différentes routes et niveaux de vols demandés :

$$\mathcal{R} = \{(r_i, RFL_i),\ i \in \mathcal{V}\}$$

On note $K = |\mathcal{R}|$ le cardinal de cet ensemble et on remarque que $K \leq n$. On peut donc également décrire cet ensemble sous la forme :

$$\mathcal{R} = \{\rho_1, \rho_2, ..., \rho_K\}$$

où $\rho_k$ est constitué d'une route et d'un RFL. On définit alors un ux de la manière suivante :

**Dé nition 8 (Flux de vols)** *Soit $\mathcal{V}$ un ensemble de vols. Un* ux de vols $\mathcal{F}_k$ *est un sous-ensemble de $\mathcal{V}$ contenant tous les vols qui suivent une même route (séquence de points de report) et demandent le même niveau de vol, c est-à-dire :*

$$\mathcal{F}_k = \{i \in \mathcal{V},\ (r_i, RFL_i) = \rho_k\}$$

Notons que l'ensemble $\mathcal{F}$ des ux constitue une partition de l'ensemble des vols $\mathcal{V}$. Par extension de la notation, on notera pour un ux de vols $\mathcal{F}_k$ :

- $r_k$ sa route, *i.e.* la route commune des vols qu'il contient ;
- $FL_k$ le niveau de vol attribué au ux, et donc à tous les avions qu'il contient ;
- $n_k = |\mathcal{F}_k|$ le nombre de vols dans $\mathcal{F}_k$.

L'agrégation des vols en ux pour l'allocation de niveaux de vol ne permet pas de résoudre les con its en rattrapage au sein d'un même ux. Ces con its devront donc être résolus par une autre méthode : par l'allocation de créneaux de décollage dans notre cas, mais des méthodes de type train d'avions [4] peuvent également être utilisées si les vitesses sont compatibles.

Il est également possible de constituer ces ux en regroupant les vols suivant la même route et ayant des RFLs proches mais pas forcément identiques. Ceci permet notamment de réduire le nombre de ux et correspond, d'un point de vue opérationnel, à regrouper les vols effectuant le même trajet origine-destination sur un nombre prédéterminé de niveaux de vol : plus les RFLs des ux à regrouper sont distants, plus le nombre de niveaux nécessaires sera petit. Il est possible, pour un écart de RFLs donné, de réaliser une optimisation afin de construire un nombre minimal de regroupements de ux. Cette méthode a été implémentée en utilisant la programmation par contraintes pour construire des ux d'un « diamètre » de trois niveaux de vol. Cependant, les solutions obtenues sont coûteuses : en effet, le graphe de contraintes associé est dense et le décalage d'un seul niveau de vol pour un tel ux peut entraîner des décalages de plusieurs niveaux pour certains vols contenus dans celui-ci.

4. Cette méthode consiste à réguler un ensemble de vols suivant la même trajectoire en donnant comme consigne aux pilotes de suivre l'avion précédent en respectant une distance minimale.

### 5.3.2 Séparation des ux en altitude

Une fois les vols agrégés en ux, il faut allouer des niveaux de vol différents aux ux qui se croisent. Pour cela, on détecte les points de croisement des différentes routes et on dira qu'il y a un con it entre deux ux $\mathcal{F}_k$ et $\mathcal{F}_\ell$ si leurs routes respectives $r_k$ et $r_\ell$ s'intersectent (dans le plan horizontal). On notera $conflit_{2D}(\mathcal{F}_k, \mathcal{F}_\ell)$ la relation de con it ainsi définie. La séparation en altitude de deux ux sécants s'exprime donc par la contrainte :

$$conflit_{2D}(\mathcal{F}_k, \mathcal{F}_\ell) \Rightarrow FL_k \neq FL_\ell$$

Ce modèle engendre un graphe de contraintes (où un n ud représente un ux et une arête entre deux n uds représente un con it entre les ux correspondants) très dense, malgré le nombre de n uds limité. Il est possible de ne traiter que les plus gros ux ($\{\mathcal{F}_k \in \mathcal{F}, n_k > n_{\min}\}$) afin de réduire la taille du problème, mais malgré ce filtrage un nombre trop faible de contraintes a pu être résolu avec une distance au RFL raisonnable.

### 5.3.3 Prise en compte de l'aspect temporel

Il est possible de réduire la densité du graphe de contraintes issu du modèle précédent en prenant en compte le temps d'occupation de chaque ux. En effet, un ux n'est pas occupé pendant la totalité de la journée considérée, et notamment les ux contenant peu de vols sont souvent occupés sur une très courte période. Le principe consiste alors à ne compter un con it que s'il y a une intersection dans le plan horizontal et que les périodes d'occupation des ux sécants se chevauchent.

Pour un vol $i$ donné, on définit $\omega_i$ sa période d'occupation du ux auquel il appartient par :

$$\omega_i = \left[\tau_i^d, \tau_i^a + \delta_{\max}\right]$$

où $\tau_i^d$ et $\tau_i^a$ sont les heures prévues de décollage et d'atterrissage respectivement. La période d'occupation d'un ux $\mathcal{F}_k$ est alors l'union des périodes d'occupation des vols qu'il contient. On note :

$$\Omega_k = \bigcup_{i \in \mathcal{F}_k} \omega_i$$

On impose alors une contrainte de séparation verticale entre deux ux s'il y a intersection dans le plan horizontal et intersection temporelle, c'est-à-dire :

$$conflit_{2D}(\mathcal{F}_k, \mathcal{F}_\ell) \ \wedge \ \Omega_k \cap \Omega_\ell \neq \emptyset \ \Rightarrow \ FL_k \neq FL_\ell$$

Ce raffinement du modèle précédent permet une réduction de la densité du graphe de contraintes. Cependant, la période d'occupation pour les plus gros ux couvre une grande partie de la journée. Si on ne considère que ceux-ci, on se retrouve alors quasiment dans la même configuration qu'avec le modèle précédent, la prise en compte de l'aspect temporel de cette manière n'ayant une in uence significative que pour les plus petits ux. D'autre part, on considère ici que le con it concerne la totalité de la route, alors qu'il est en réalité limité à une zone précise. La section suivante propose une approche plus efficace pour la prise en compte de l'aspect temporel de la détection de con it entre ux de vols.

## 5.4 Allocation par vol

Le modèle de con it présenté dans le chapitre 3 peut être exploité dans la détection de con it entre ux de vol pour l'allocation de niveaux, afin d'améliorer la prise en compte de l'aspect temporel des con its. Le principe est de ne compter un con it entre deux ux que s'il existe au moins un con it entre des vols de chaque ux. La section 5.4.1 propose une adaptation du modèle de con it pour la détection en deux dimensions (2D), et la section 5.4.2 présente une amélioration du modèle d'allocation par ux utilisant ces con its.

Cependant, cette extension du modèle nous a conduit vers un modèle plus fin où l'allocation de niveau de vol est effectuée individuellement pour chaque vol . Les sections 5.4.3 et 5.4.4 présentent ce modèle.

### 5.4.1 Con its 2D + temps

La méthode de détection décrite dans le chapitre 3 donne une caractérisation en 4D des con its aériens : il y a con it si et seulement si $d_{ij} \in \mathcal{C}_{ij}$ où $\mathcal{C}_{ij} = \left[\underline{lb}_1, \overline{ub}_1\right] \cup \cdots \cup \left[\underline{lb}_\sigma, \overline{ub}_\sigma\right]$.

Afin de pouvoir utiliser cette représentation des con its pour la phase de séparation en altitude, il est nécessaire de restreindre ce modèle à une détection dans le plan horizontal. Cela signifie que la méthode reste identique, à ceci près que la distance verticale entre les points n'est pas testée. Nous utiliserons la notation $\mathcal{C}_{ij}^H$ pour désigner le con it ainsi obtenu entre les avions $i$ et $j$, et il y aura dans ce cas un con it si $d_{ij} \in \mathcal{C}_{ij}^H$. On a en particulier la propriété : $\mathcal{C}_{ij} \subseteq \mathcal{C}_{ij}^H$, c'est-à-dire que les intervalles de con its détectés en 4D sont inclus dans ceux détectés en « 2D + temps ». On observe en outre que :

$$\left| \bigcup_{i,j \in [1,n]^2} \mathcal{C}_{ij}^H \right| \gg \left| \bigcup_{i,j \in [1,n]^2} \mathcal{C}_{ij} \right|$$

avec des con its « 2D + temps » d'une taille 4 à 5 fois supérieure à celles des con its 4D en moyenne.

Dans le cadre de l'allocation de niveaux de vol, aucun choix n'est fait sur les variables $d_{ij}$, et il faut donc effectuer la résolution en prenant en compte toutes les valeurs possibles pour celles-ci, c'est-à-dire l'intervalle $[-\delta_{\max}, \delta_{\max}]$. On définit alors un con it dans le plan horizontal entre deux vols $i$ et $j$ par :

$$conflit^H(i,j) \;\Leftrightarrow\; [-\delta_{\max}, \delta_{\max}] \cap \mathcal{C}_{ij}^H \neq \emptyset \tag{5.1}$$

ce qui signifie qu'il y a un con it entre les vols $i$ et $j$ s'il y a une valeur possible pour $d_{ij}$ qui soit contenue dans les intervalles de valeurs interdites $\mathcal{C}_{ij}^H$.

### 5.4.2 Ra nement du modèle en ux

Il est possible de modifier le modèle d'allocation présenté dans la section 5.3 en utilisant la définition de con it précédente. L'idée de ce raffinement du modèle est de ne compter un con it entre deux ux que s'il existe deux avions (un dans chaque ux) ayant un con it au

sens de la définition 5.1. Ce principe se traduit par la définition suivante d'un con it entre deux ux :

$$conflit(\mathcal{F}_k, \mathcal{F}_\ell) \Leftrightarrow \exists (i,j) \in \mathcal{F}_k \times \mathcal{F}_\ell \text{ tels que } conflit^H(i,j)$$

et, comme pour les modèles précédents, on impose la contrainte :

$$conflit(\mathcal{F}_k, \mathcal{F}_\ell) \Rightarrow FL_k \neq FL_\ell$$

Utiliser les con its entre vols pour déterminer les con its entre ux de vols nous a conduit à envisager un modèle où l'attribution de niveaux de vol ne serait plus faite sur des ux, mais directement sur les vols. En effet, les graphes de contraintes dérivés de ce modèle sont ici encore très denses et les solutions obtenues impliquent des écarts au RFL trop importants (voir section 8.1.1). L'idée est donc d'utiliser ce modèle comme base pour un modèle non agrégé, que nous décrivons dans la section suivante.

### 5.4.3 Séparation des vols en altitude

Réaliser l'allocation de niveaux de vol en considérant directement l'ensemble des vols plutôt que les ux de vols permet une plus grande liberté lors de la résolution. Notamment, des vols suivant la même route à différents moments de la journée pourront se voir attribuer des niveaux de vol différents si le trafic le nécessite, ceci n'étant pas possible avec les modèles précédents. L'instance créée est de plus grande taille puisque les contraintes portent sur quelques milliers de vols, contre quelques centaines de ux pour les modèles précédents. Cependant, le graphe de contraintes associé au problème a une densité beaucoup plus faible (environ quatre fois plus faible en moyenne), la densité étant définie par :

$$\frac{|E|}{|V|(|V|-1)/2}$$

où $E$ est l'ensemble des arêtes (les con its) et $V$ est l'ensemble des n uds (les vols), c'est-à-dire le nombre d'arêtes divisé par le nombre d'arêtes possibles.

Encore une fois, seules les contraintes du modèle sont modifiées. Celles-ci consistent désormais à séparer en altitude les vols en con it au sens de l'équation 5.1 :

$$conflit^H(i,j) \Rightarrow FL_i \neq FL_j \tag{5.2}$$

Les ux de vols mis en uvre dans le précédent modèle sont en fait en partie reconstitués, dans la mesure où deux vols suivant la même route, ayant le même RFL et n'étant pas en con it au sens de la relation 5.1 se retrouveront probablement affectés au même FL. Le train d'avions évoqué dans la section précédente serait constitué grâce à l'allocation de créneaux de décollage décrite dans le chapitre 4.

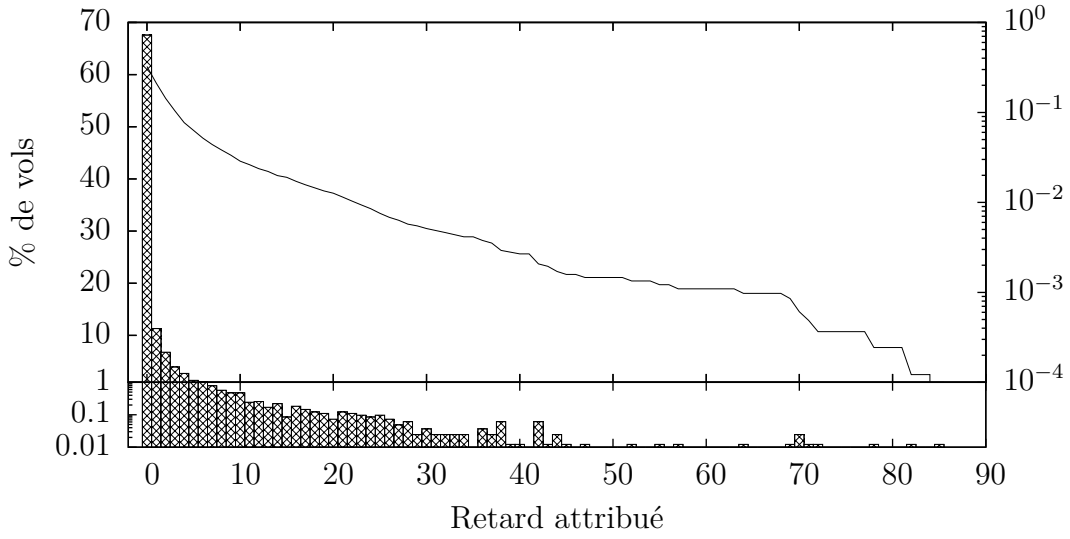



FIGURE 5.3 – Distribution des retards alloués pour une journée de trafic sur l'espace français. L'histogramme (échelle de gauche) donne pour chaque retard $\delta$ le pourcentage de vols retardés de $\delta$ (l'échelle est logarithmique sur l'intervalle $[0, 1]$). La courbe (échelle de droite, logarithmique) donne l'inverse de la fonction de répartition des retards.

### 5.4.4 Relaxation temporelle

Les expérimentations avec ce modèle ont montré que le problème généré était malgré tout très dur, menant à des écarts au RFL importants ou à un grand nombre de contraintes non satisfaites, notamment pour les valeurs couramment utilisées pour $\delta_{\max}$ (*e.g.* 90 min). Or l'observation des solutions de l'allocation de créneaux de décollage indique que la plupart des vols ne sont pas retardés ou le sont avec un retard très faible (quelques minutes), comme illustré sur la figure 5.3. On constate notamment que 95 % des vols ont un retard inférieur à 5 min et 99 % ont un retard inférieur à 20 min. En tenant compte de ces observations, une relaxation temporelle des contraintes est mise en uvre, en fixant simplement $\delta_{\max}$ à 0 pour l'allocation de niveaux de vol.

Les contraintes du modèle deviennent :

$$0 \notin \mathcal{C}_{ij}^{H} \Rightarrow FL_i \neq FL_j \tag{5.3}$$

Il est toutefois possible d'être un peu moins restrictif et de fixer $\delta_{\max}$ à une valeur faible non nulle ($\leq 5$ min). Ceci permettrait notamment de prendre en compte de l'incertitude sur le suivi longitudinal des trajectoires lors de la phase de croisière.

## 5.5 Optimisation

Il est important, lors de l'allocation de niveaux de vol, de respecter au mieux la demande des compagnies aériennes. En effet, comme signalé dans la section précédente, celles-ci

demandent à obtenir un niveau de vol le plus proche possible de l'altitude optimale de croisière de l'avion effectuant le vol. Ce critère est pris en compte dans la fonction de coût utilisée pour l'optimisation, et définie par :

$$\text{cost}_{FL} = \sum_{i=1}^{n} |RFL_i - FL_i|$$

chaque variable de niveau de vol étant par ailleurs bornée dans un intervalle restreint autour du RFL.

Cette fonction de coût ne représente que grossièrement le coût réel de l'allocation de niveaux de vol pour les compagnies aériennes. Il serait possible, si suffisamment de données de constructeurs étaient disponibles, de définir une fonction prenant en compte le coût réel en termes de consommation de carburant, du type :

$$\text{cost}_{conso} = \sum_{i=1}^{n} (conso_i(FL_i) - conso_i(RFL_i))$$

où $conso_i$ est une fonction qui donne le coût du vol $i$ (dépendant notamment du type d'avion utilisé pour le vol) en fonction de son niveau de vol de croisière.

# Troisième partie

# Méthodes de résolution

# Chapitre 6

# Résolution par algorithmes évolutionnaires

Les problèmes de résolution de con its aériens, tels que celui que nous avons présenté dans le chapitre 4, sont des problèmes de grande taille et fortement combinatoires. La fonction à optimiser possède une multitude d'optima locaux et n'est pas convexe. D'autre part, l'aspect combinatoire du problème rend les fonctions de coût difficiles à optimiser. Pour ces raisons, les méthodes du type Newton ou programmation linéaire mixte ont été écartées au profit de deux méthodes : une métaheuristique (algorithme génétique) décrite dans ce chapitre et une méthode d'optimisation globale explorant exhaustivement l'espace de recherche, la programmation par contraintes, présentée dans le chapitre 7.

Les algorithmes génétiques ont été choisis notamment pour leur robustesse et pour le peu de « bonnes propriétés » (continuité, dérivabilité...) qu'ils requièrent à propos de la fonction à optimiser. D'autre part, ce type d'algorithme a été utilisé avec succès pour la résolution de problèmes liés au trafic aérien, tels que la résolution de con its aériens [Granger 02], l'optimisation du trafic au sol sur un grand aéroport [Gotteland 04], l'optimisation des ux de trafic [Gianazza 04a] ou encore l'allocation de niveaux de vol [Constans 05b].

Les principes des algorithmes génétiques sont rappelés dans la section 6.1. La section 6.2 montre l'application des algorithmes génétiques au problème d'allocation de créneaux de décollage décrit dans le chapitre 4.

## 6.1 Principe des algorithmes génétiques

Les algorithmes génétiques (AG) s'inspirent de la théorie de l'évolution naturelle pour rechercher les optima d'un critère défini sur un espace de recherche (ou espace d'états). Dans ce cadre, un *individu* représente un point dans cet espace et l'algorithme génétique consiste alors à faire évoluer une *population* d'individus en utilisant des processus de *sélection*, de *mutation* et de *croisement* inspirés des processus naturels du même nom. Les principes fondamentaux en sont développés dans les années 60 et 70 et présentés dans [Holland 75], mais c'est [Goldberg 89] qui a suscité de l'intérêt pour ce type d'algorithme en présentant

des applications à des problèmes concrets. Des résultats théoriques sur les algorithmes génétiques existent, notamment une démonstration complète de convergence stochastique [Cerf 94]. Cependant, ces résultats ne sont pas exploitables pour la mise en pratique des algorithmes génétiques, qui fait plutôt appel à l'expertise de l'utilisateur. [Eiben 03] propose un descriptif des développements plus récents basés sur le paradigme de l'évolution naturelle.

La section 6.1.1 présente le fonctionnement général d'un algorithme génétique ainsi que les différents paramètres qui permettent de l'adapter au problème à optimiser. La section 6.1.2 présente deux améliorations classiques contribuant aux performances de l'algorithme.

### 6.1.1 Algorithme génétique élémentaire

L'algorithme 6.1 propose un cadre générique d'algorithme génétique faisant évoluer une population $P$ d'individus. Après initialisation de la population, l'algorithme génère de manière successive de nouvelles populations (on parle souvent de plusieurs *générations* de la population) en appliquant les principes de sélection (ou intensification, c'est-à-dire qu'on cherche plus autour des points prometteurs en dupliquant les individus) et de croisement et mutation (ou diversification) définis par l'utilisateur en adéquation avec le problème à résoudre, jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint. Le meilleur élément de la population est alors renvoyé.

**Entrées :** $F$ une fonction à optimiser
1: $P \leftarrow$ init()
2: **tant que** $\neg$arrêt **faire**
3: $e \leftarrow$ évaluation$(P, F)$
4: $P' \leftarrow$ sélection$(P, e)$
5: $P_c \leftarrow$ croisement$(P')$
6: $P_m \leftarrow$ mutation$(P')$
7: choisir $P_r \subset P'$ t.q. $|P_r| = |P'| - |P_c| - |P_m|$ // on complète la population
8: $P \leftarrow P_c \cup P_m \cup P_r$
9: **n tant que**
10: $e \leftarrow$ évaluation$(P, F)$
11: **renvoyer** meilleur$(P, e)$

**Algorithme 6.1:** Algorithme génétique

Les différentes étapes de cet algorithme sont décrites de manière détaillée dans les sections suivantes.

#### Codage des données

Historiquement, le codage des individus prenait la forme d'une chaîne de bits constituée de la concaténation des représentations binaires des variables de décision. Cette représenta-

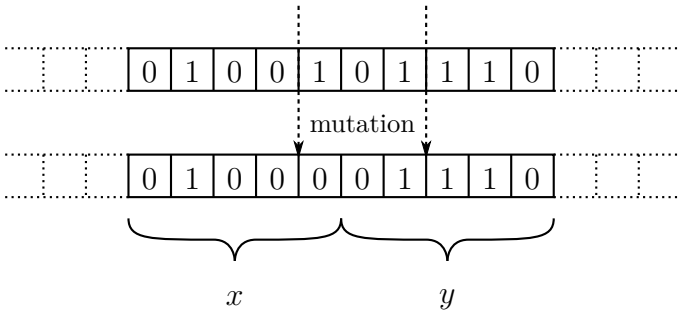



FIGURE 6.1 – Limites du codage binaire des individus : deux variables « proches » dans la représentation ne le sont pas forcément dans le problème.

tion est la plus proche du modèle naturel : elle re ète la différence entre le génotype [1] et le phénotype [2] dans la mesure où les caractéristiques « observables » (*i.e.* la qualité de la solution) sont déterminés par des sous-ensembles non disjoints de gènes. Ce codage permet également de définir des opérateurs de croisement et de mutation simples et génériques et a été le support des premiers résultats de convergence des algorithmes génétiques utilisant la théorie des schémas [Vose 91].

Cependant, pour les problèmes de grande dimension, ce codage binaire n'est pas adapté car il est trop « aveugle » : chaque variable est représentée par une portion de la chaîne et leur proximité au sein de celle-ci ne re ète pas la structure du problème. La figure 6.1 illustre ce problème : les variables $x$ et $y$ sont consécutives dans la chaîne de codage et une mutation [3] peut entraîner une modification des deux variables simultanément, sans que celles-ci aient un lien dans le problème. Afin de conserver la structure du problème et de faciliter les opérations de codage et de décodage des données, les individus peuvent être codés de manière plus spécialisée, mieux adaptée au problème à résoudre [Michalewicz 92], par exemple sous la forme d'une chaîne d'entiers ou de réels. Il sera alors plus facile de mettre en uvre des opérateurs de mutation et de croisement adaptés au problème (voir sections suivantes). On distingue souvent algorithmes génétiques et algorithmes évolutionnaires vis-à-vis de leurs codages respectifs (binaire pour les premiers, entiers ou réels pour les seconds). Nous utiliserons indifféremment ces deux appellations dans la suite du document pour désigner les algorithmes utilisés.

### Population initiale

La population initiale peut avoir une in uence sur la rapidité de convergence de l'algorithme. Si la « forme » de la fonction à optimiser n'est pas connue, la population sera générée de manière aléatoire. Ce tirage aléatoire est fait de manière uniforme sur chacune

1. Le génotype est la séquence des gènes (ici les valeurs des bits) contenus dans un chromosome (ici la chaîne de codage).
2. Le phénotype est l'ensemble des caractères observables d'un individu.
3. Une mutation (voir page 88) consiste en une perturbation locale de la solution, comme par exemple le changement de la valeur d'un bit ou d'une sous-chaîne de bits.

des composantes de l'espace d'états. Dans la mesure du possible, les individus générés doivent respecter les contraintes du problème. Si ce n'est pas possible, leur évaluation sera pénalisée en fonction du nombre des contraintes violées. Il est également possible, si des informations sont connues à propos de la fonction à optimiser, de restreindre l'initialisation de la population à certaines portions de l'espace d'états.

La taille de la population est en général constante lors de l'exécution de l'algorithme génétique (comme c'est la cas dans l'algorithme 6.1). Le choix du nombre d'individus $K$ est un compromis entre le temps de calcul et la vitesse de convergence.

### Condition d'arrêt

La condition d'arrêt de l'algorithme génétique (ligne 2 de l'algorithme 6.1) peut revêtir des formes différentes : valeur cible atteinte pour l'évaluation, stabilité du meilleur élément de la population, nombre de générations maximal $N$ atteint... Le choix est dépendant du problème ainsi que du temps de calcul disponible.

### Évaluation

Si $F$ est la *fonction d évaluation* (*i.e.* la fonction à optimiser), la phase d'évaluation consiste simplement à appliquer $F$ à chacun des $K$ individus :

1: $e$ séquence des évaluations
2: **pour tout** $i \in [1, K]$ **faire**
3: $\quad e_i \leftarrow F(P_i)$
4: **n pour**
5: **renvoyer** $e$

**Algorithme 6.2:** évaluation$(P, F)$

Dans le cas d'un problème sous contraintes (comme c'est le cas pour la résolution de con its décrite dans le chapitre 4), la fonction d'évaluation $F$ peut prendre celles-ci en compte en attribuant une pénalité aux individus qui violent une (ou plusieurs) contrainte(s). En effet, il est parfois difficile de construire des individus admissibles et de maintenir la population dans le domaine admissible lors des opérations de croisement et de mutation. D'autre part, il est parfois intéressant de garder des individus non admissibles au sein de la population, car un croisement entre deux individus non admissibles peut générer des individus admissibles de bonne qualité.

La pénalité associée à une violation de contrainte ne devra pas être trop importante, car l'algorithme génétique risque alors de privilégier la recherche de solutions admissibles plutôt que celle de l'optimum. Elle ne devra pas être trop faible pour la raison inverse, sauf dans le cas où on cherche un compromis entre le nombre de contraintes résolues et le coût de la solution.

### Processus de sélection

La sélection permet de retenir avec une certaine probabilité les meilleurs éléments de la génération courante et d'éliminer les moins bons, tout en conservant une proportion d'individus moins bons qui peuvent éventuellement contenir des morceaux de très bonne solution. De nombreux principes de sélection existent, comme la sélection par rang [Baker 85] ou la sélection par tournoi [Brindle 81]. Nous avons choisi d'implémenter (voir section 6.2.6) un schéma classique de sélection appelé *stochastic remainder without replacement* [Goldberg 89], dérivé du principe *roulette wheel selection*.

Le *roulette wheel selection* [De Jong 75] consiste à tirer $K$ éléments dans la population de telle manière qu'à chaque tirage, la probabilité $p_i$ de tirer un individu $i$ donné soit proportionnelle à son évaluation $e_i$ :

$$p_i = \frac{e_i}{\sum_{j=1}^{K} e_j}$$

Lorsque la taille de la population est faible, un biais de sélection peut se produire en raison du faible nombre de tirages effectués, l'espérance mathématique n'étant alors pas atteinte.

Le *stochastic remainder without replacement* permet d'éviter ce problème, le principe étant de systématiquement choisir les très bons individus puis de compléter la population en utilisant le *roulette wheel selection*. Pour chaque individu $i$, on calcule le rapport $r_i$ de son évaluation sur la moyenne des évaluations :

$$r_i = \frac{e_i}{\bar{e}} \quad \text{où } \bar{e} = \frac{1}{K}\sum_{j=1}^{K} e_j$$

Chaque individu $i$ est choisi $\lfloor r_i \rfloor$ fois, puis la population est complétée en utilisant le *roulette wheel selection* sur l'ensemble des individus, en utilisant pour chacun la valeur $r_i - \lfloor r_i \rfloor$ à la place de leur évaluation $e_i$. L'algorithme 6.3 donne la procédure de sélection utilisant le *stochastic remainder without replacement*.

Lors de l'exécution du processus de sélection sur une population $P$, il existe un risque de « perdre » le meilleur individu. Afin d'améliorer l'efficacité des algorithmes génétiques, il est possible de conserver, lors de la sélection, le (ou les) meilleur(s) chromosome(s) avant d'appliquer la méthode de sélection choisie pour compléter la population. Ce raffinement est appelé *élitisme*.

Le processus de sélection permet la convergence de l'algorithme génétique vers les meilleurs individus. Cependant, il est nécessaire de maintenir la diversité au sein de la population afin d'explorer l'espace de recherche. Le croisement et la mutation assurent cette diversification.

### Croisement

L'opération de croisement permet d'améliorer la diversité de la population par une recomposition des gènes de plusieurs individus. Il consiste à générer des individus « enfants » à partir d'individus « parents » (en général, on choisit deux parents pour générer deux

1: $P' \leftarrow \emptyset$
2: $s \leftarrow \sum_{j=1}^{K}(r_j - \lfloor r_j \rfloor)$
3: **pour tout** $i \in [1, K]$ **faire**
4: ajouter $\lfloor r_i \rfloor$ fois $P_i$ dans $P'$
5: **n pour**
6: **tant que** $|P'| < K$ **faire**
7: $\zeta \in [0, s]$ // distribution uniforme pour le *roulette wheel selection*
8: $j \leftarrow 0$
9: **tant que** $\zeta \geq 0$ **faire**
10: $j \leftarrow j + 1$
11: $\zeta \leftarrow \zeta - (r_j - \lfloor r_j \rfloor)$
12: **n tant que**
13: ajouter $P_j$ dans $P'$
14: **n tant que**
15: **renvoyer** $P'$

**Algorithme 6.3:** sélection$(P, e)$

enfants). Le croisement s'effectue sur un pourcentage $p_{cross}$ de la population, $p_{cross}$ étant un paramètre de l'algorithme génétique, selon la procédure décrite dans l'algorithme 6.4.

1: $P_c \leftarrow \emptyset$
2: $n_{cross} \leftarrow K.p_{cross}/2$ // nombre de croisements à réaliser
3: **pour** $k = 1$ jusqu'à $n_{cross}$ **faire**
4: $(P_i, P_j) \in P^2$, $i \neq j$ // choix des parents
5: $(P'_i, P'_j) \leftarrow$ croise$(P_i, P_j)$
6: $P_c \leftarrow P_c \cup \{P'_i, P'_j\}$
7: **n pour**
8: **renvoyer** $P_c$

**Algorithme 6.4:** croisement$(P)$

La recomposition des gènes (ligne 5 de l'algorithme) peut se faire de multiples façons et peut parfois être adaptée au problème et au codage des individus. Pour des individus dont le codage est discret, le croisement consiste souvent en un échange d'un ou plusieurs « gènes » (*i.e.* portions de chromosomes) entre les parents, comme illustré sur la figure 6.2.

Pour les problèmes continus, il est plus courant de créer les gènes des enfants comme barycentres des gènes de parents :

$$\begin{aligned} \vec{e_a} &= \alpha\vec{p_a} + (1-\alpha)\vec{p_b} \\ \vec{e_b} &= (1-\alpha)\vec{p_a} + \alpha\vec{p_b} \end{aligned}$$

où $\vec{p_a}, \vec{p_b}, \vec{e_a}$ et $\vec{e_b}$ représentent les codages respectifs des parents $a$ et $b$ et des enfants $a$ et $b$ et $\alpha$ est un coefficient de pondération aléatoire en général choisi dans $[0; 1]$.

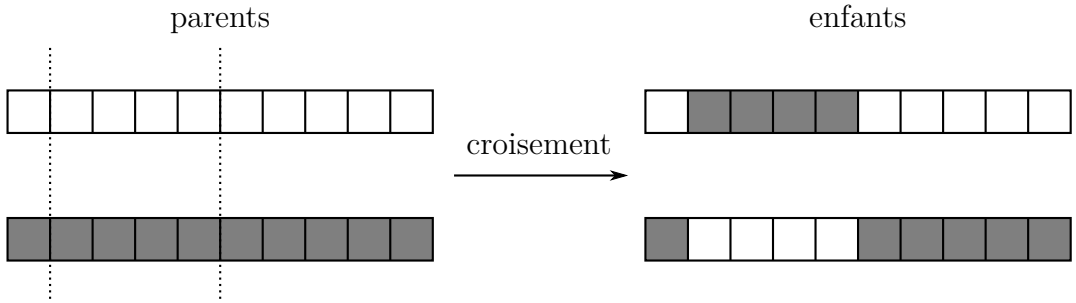



FIGURE 6.2 – Opérateur de croisement pour un codage discret. Les chromosomes des parents sont coupés en deux endroits, les gènes compris entre les points de coupure sont échangés.

**Fonctions partiellement séparables** Lorsque la fonction à maximiser est *partiellement séparable* [Griewank 82], c'est-à-dire qu'elle s'exprime comme la somme de termes positifs ne faisant intervenir qu'une partie des variables, alors il est possible de définir des opérateurs de croisement permettant de diriger le croisement vers les zones intéressantes de l'espace de recherche [Durand 98]. On s'intéresse dans ce cas aux fonctions d'évaluation qui peuvent s'écrire sous la forme :

$$F(x_1, ..., x_n) = \sum_{j=1}^{m} F_i(x_{j_1}, ..., x_{j_{n_i}})$$

où $\forall i \in [1, m], F_i$ est une des composantes positives de la fonction $F$, dépendant des variables $x_{j_1}, ..., x_{j_{n_i}}$.

On définit alors l'ensemble $E_k$ comme l'ensemble des indices des fonctions $F_i$ qui portent sur la variable $x_k$ :

$$E_k = \left\{ i \,|\, x_k \in \{x_{j_1}, ..., x_{j_{n_i}}\} \right\}$$

qui permet d'introduire pour chaque variable $x_k$ son *évaluation locale* :

$$G_k(x_1, ..., x_n) = \sum_{i \in E_k} F_i(x_{j_1}, ..., x_{j_{n_i}})$$

Cette évaluation locale correspond à la contribution de la variable $x_k$ dans l'évaluation de l'individu.

On peut alors définir un opérateur de croisement qui tient compte de cette évaluation locale en choisissant pour chaque variable celle du parent qui a la meilleure évaluation locale, à un facteur $\nu$ près. Cela se traduit (dans le cas d'une maximisation) par le choix, pour la variable $x_k$ du premier enfant, de :

- la variable $x_k$ du parent $a$ si $G_k(\vec{p_a}) > G_k(\vec{p_b}) + \nu$ ;
- la variable $x_k$ du parent $b$ si $G_k(\vec{p_a}) < G_k(\vec{p_b}) - \nu$ ;
- une valeur aléatoire (ou barycentrique) pour la variable $x_k$ si $|G_k(\vec{p_a}) - G_k(\vec{p_b})| \leq \nu$

Pour obtenir deux enfants différents, on peut construire le deuxième enfant soit de la même manière que le premier mais avec une valeur différente pour $\nu$, soit lui affecter les valeurs non retenues pour le premier, soit encore utiliser un opérateur de croisement plus classique.

Nous utiliserons cette méthode de croisement sur fonction séparable pour la résolution du problème d'allocation de créneaux de décollage (voir section 6.2.5).

**Mutation**

La procédure de mutation consiste à modifier de manière aléatoire un ou plusieurs gènes chez un pourcentage $p_{mut}$ donné d'individus, $p_{mut}$ étant un paramètre de l'algorithme génétique, comme décrit dans l'algorithme 6.5.

1: $P_m \leftarrow \emptyset$
2: $n_{mut} \leftarrow K.p_{mut}$ // nombre de mutations à réaliser
3: **pour** $k = 1$ jusqu'à $n_{mut}$ **faire**
4: $P_i \in P$ // choix d'un individu
5: $P'_i \leftarrow \text{mute}(P_i)$
6: $P_m \leftarrow P_m \cup \{P'_i\}$
7: **n pour**
8: **renvoyer** $P_m$

**Algorithme 6.5:** mutation($P$)

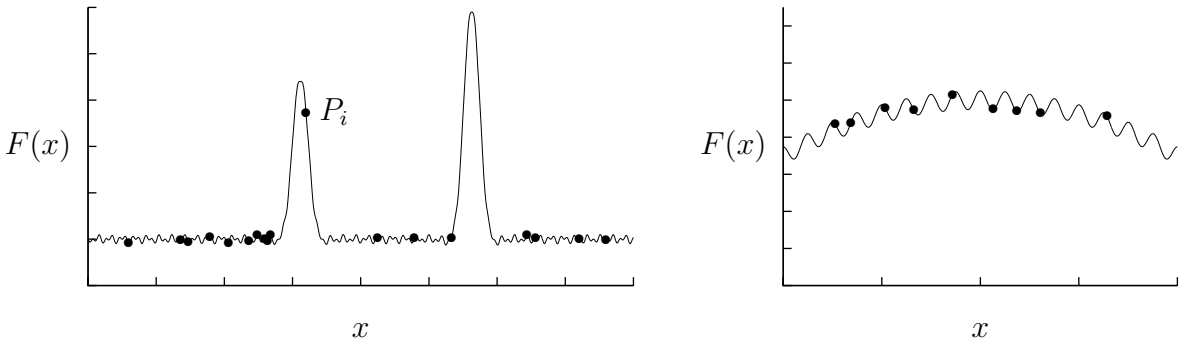
La mutation d'un individu (ligne 5 de l'algorithme) est souvent réalisée en tirant un ou plusieurs gènes de manière aléatoire, puis en modifiant ces gènes. La modification des gènes peut consister en l'addition d'un bruit aléatoire (*e.g.* un bruit gaussien) à la valeur originale. Dans le cas des problèmes discrets, elle prend plutôt la forme d'un changement de valeur (avec une certaine distance maximale si la notion de distance est définie).

Comme dans le cas du croisement, il est également possible de définir un opérateur de mutation prenant en compte les spécificités du problème à résoudre (voir section 6.2.4 pour l'application de ce principe dans le cadre de l'allocation de créneaux de décollage). Il faut toutefois garder à l'esprit que le croisement et la mutation permettent l'exploration de l'espace d'états. Il est donc nécessaire que ces opérateurs gardent une part d'aléatoire dans leur implémentation afin de ne pas converger trop vite vers des optima locaux.

### 6.1.2 Ra nements

Au début de l'exécution d'un algorithme génétique, l'initialisation de la population mène souvent à des situations où quelques individus ont une excellente adaptation tandis que le reste de la population a une adaptation médiocre (voir figure 6.3(a)). En laissant opérer les processus de sélection décrits précédemment, ces quelques individus risquent d'occuper une proportion importante de la population dès la génération suivante, ce qui conduit généralement à une convergence prématurée. En fin d'exécution, il se produit le phénomène inverse, c'est-à-dire que la moyenne $\bar{e}$ des évaluations est proche de l'évaluation du meilleur individu observé (voir figure 6.3(b)). Dans ce cas, la sélection devient quasiment uniforme (*i.e.* équiprobable) et donc inefficace pour assurer le choix des meilleurs individus.

Afin de prévenir ce type de situation, il existe des procédés qui permettent d'éviter que les individus les mieux adaptés éliminent les individus les moins adaptés. Nous détaillerons en particulier le principe de mise à l'échelle, ou *scaling*, et le principe de *sharing*, qui seront utilisés pour la résolution du problème d'allocation de créneaux.




(a) L'individu $P_i$ risque d'être reproduit un très grand nombre de fois lors de la sélection et la convergence risque de se faire prématurément vers le sommet secondaire.

(b) Les individus ont des évaluations proches et la sélection risque de reproduire les moins bons et les meilleurs avec la même probabilité.

FIGURE 6.3 – Exemples de cas où la sélection risque d'être inefficace.

### Scaling

La mise à l'échelle ou *scaling* consiste à modifier les évaluations afin d'amplifier ou de réduire les écarts entre les individus. Le processus de sélection est alors appliqué sur les évaluations mises à l'échelle pour en améliorer l'efficacité.

Il existe dans la littérature de nombreuses méthodes pour la mise à l'échelle. Cette section présente les plus répandues.

Le *scaling linéaire*, déjà décrit dans [Bagley 67], modifie les évaluations selon la relation :

$$e_i^s = a.e_i + b$$

où $e_i^s$ est l'évaluation mise à l'échelle de l'individu $i$ et $a > 0$ et $b$ sont les paramètres du *scaling*. Lorsque $a < 1$, les écarts entre individus sont réduits, favorisant l'exploration, tandis que si $a > 1$, les écarts sont renforcés et c'est la convergence qui est privilégiée. Cette mise à l'échelle est statique (dans le sens où le facteur de « compression » ou de « dilatation » de l'échelle n'évolue pas au fil des générations) et peut donc pénaliser certaines phases de l'algorithme.

Le *scaling exponentiel* [Gillies 85] se définit par la relation :

$$e_i^s = (e_i)^k$$

Lorsque $k \in ]0, 1[$, les écarts entre individus sont réduits. Ils sont augmentés lorsque $k > 1$. Ces propriétés sont souvent exploitées en augmentant la valeur du paramètre $k$ au fur et à mesure des générations, afin de favoriser l'exploration au démarrage de l'algorithme et la convergence à la fin. Cette croissance de $k$ est souvent directement reliée au numéro de la génération en cours. La figure 6.4 illustre l'effet d'un *scaling* exponentiel.

La mise à l'échelle par *sigma truncation*, attribuée à [Forrest 85], est une méthode qui fonctionne de manière dynamique en prenant en compte la répartition des évaluations. Elle

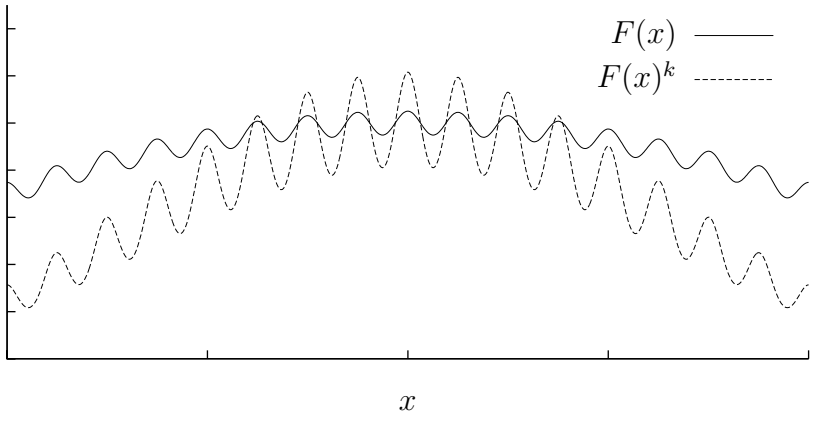



FIGURE 6.4 – Effet d'un *scaling* exponentiel de facteur $k = 4$.

se définit par :

$$e_i^s = \max(0, e_i - (\bar{e} - m.\sigma_e))$$

où $\bar{e}$ est la moyenne des évaluations de la population, $\sigma_e$ leur écart type et $m$ un paramètre de la mise à l'échelle (en général, on choisit $m \in [1, 3]$). Les évaluations les moins bonnes (inférieures à $\bar{e} - \sigma_e$) sont tronquées à 0, ce qui a pour effet de favoriser les meilleurs éléments, notamment lorsque l'écart type est faible. C'est cet opérateur de mise à l'échelle qui a été retenu dans notre application car il permet de favoriser les solutions sans con it si leur coût en termes de retard n'est pas trop élevé (voir section 6.2.2).

### Sharing

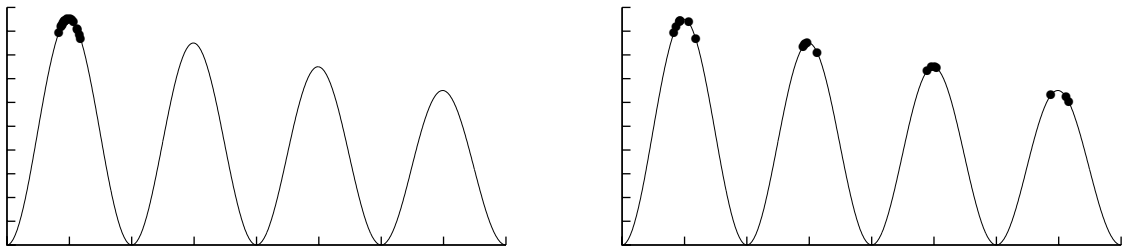
Le *sharing* [Goldberg 87] cherche à favoriser l'exploration en prenant en compte la proximité géographique des individus dans l'espace d'états. Le principe est de modifier l'évaluation des individus ayant un voisinage « dense » (*i.e.* contenant de nombreux autres individus). Il est donc nécessaire de définir une notion de distance $d$ entre individus. Celle-ci est utilisée pour calculer l'évaluation modifiée de la manière suivante :

$$e_i^{sh} = \frac{e_i}{\sum_{j=1}^{K} S(d(P_i, P_j))}$$

avec

$$S(d) = \begin{cases} 1 - \left(\frac{d}{d_{sh}}\right)^{\alpha} & \text{si } d < d_{sh} \\ 0 \text{ sinon} \end{cases}$$

Le paramètre $d_{sh}$ définit la distance maximale d'in uence du *sharing* : deux individus séparés d'une distance supérieure à $d_{sh}$ ne se pénalisent pas. Sa valeur doit être choisie en adéquation avec le problème et la fonction de distance définie. Le paramètre $\alpha$ détermine la « force » du *sharing*. Plus $\alpha$ est grand, plus les groupes d'individus agglomérés sont pénalisés. La figure 6.5 illustre les effets du *sharing* sur la répartition de la population.



(a) Sans *sharing*, la population a tendance à ne former qu'un groupe autour de la meilleure valeur.

(b) Avec *sharing*, les individus sont répartis sur les sommets secondaires, proportionnellement à la valeur de ceux-ci.

FIGURE 6.5 – Illustration des effets du *sharing*. Les points (+) représentent la population à la fin de la convergence de l'algorithme.

Cette procédure de *sharing* donne généralement de bons résultats, mais sa complexité temporelle est en $\mathcal{O}(K^2)$, puisqu'il faut calculer la distance entre toutes les paires possibles d'individus, alors que les autres étapes de l'algorithme génétique ont une complexité en $\mathcal{O}(K)$. [Yin 93] présente une variante du *sharing* fondée sur la séparation des individus en groupes, ou *clusters*. Ce procédé nécessite l'introduction d'une notion de barycentre et est paramétré par deux valeurs $d_{\min} < d_{\max}$ qui permettent le découpage en *clusters*. L'algorithme parcourt la population en appliquant les étapes suivantes jusqu'à obtention d'un point fixe :

- chaque individu est initialement le centre d'un *cluster* ;
- si l'individu considéré se trouve à une distance inférieure à $d_{\max}$ du centre d'un *cluster* existant, il est ajouté à celui-ci (le centre en est recalculé), sinon, il devient le centre d'un nouveau *cluster* ;
- si les centres de deux *clusters* sont situés à une distance inférieure à $d_{\min}$ l'un de l'autre, alors les *clusters* correspondant sont réunis en un *cluster* dont le centre est le barycentre des deux centres initiaux.

Une fois les *clusters* déterminés, l'évaluation de chaque individu est modifiée de la manière suivante :

$$e_i^{sh} = \frac{e_i}{f_i}, \quad \text{avec } f_i = |\mathrm{cl}_i|.\left(1 - \left(\frac{d(P_i, C_{\mathrm{cl}_i})}{2d_{\max}}\right)^{\alpha}\right)$$

où $|\mathrm{cl}_i|$ est le nombre d'éléments du *cluster* contenant l'individu $P_i$, $C_{\mathrm{cl}_i}$ le centre de ce *cluster* et $\alpha$ un paramètre déterminant l'intensité du *sharing*.

Ce processus de *sharing clusterisé* peut être implémenté par un algorithme de complexité temporelle en $\mathcal{O}(K \log K)$, avec une efficacité souvent équivalente au processus de *sharing* standard. Il est mis en uvre dans le cadre du problème d'allocation de créneaux de décollage afin d'améliorer l'exploration de l'espace de recherche tout en permettant des temps d'exécution acceptables. Il est par ailleurs intéressant de faire de l'élitisme lorsqu'un processus de *sharing* est utilisé, afin de ne pas « perdre » le meilleur élément dans le cas où

celui-ci se retrouve pénalisé.

## 6.2 Allocation de créneaux de décollage par AG

Un des principaux intérêts des algorithmes génétiques est que ceux-ci ne requièrent que peu de connaissances sur la fonction à optimiser et ne nécessitent notamment aucune hypothèse sur la régularité de celle-ci. Ce dernier point constitue un atout important dans notre étude, étant donné que le problème d'allocation de créneaux de décollage sans con it est un problème d'optimisation combinatoire possédant de nombreux optima locaux.

Cette section détaille les structures de données, les opérateurs de croisement et de mutation ainsi que les différents paramètres utilisés pour la résolution de ce problème à l'aide d'un algorithme génétique.

### 6.2.1 Variables et structures de données

Comme indiqué dans la description du modèle (voir chapitre 4), les variables de l'algorithme génétique sont les retards alloués aux vols :

$$\Delta = \{\delta_i, \forall i \in [1, n]\}$$

où chaque $\delta_i \in [0, \delta_{\max}]$. Un élément de population est donc représenté par un tableau contenant ces $n$ variables, soit une chaîne d'entiers.

### 6.2.2 Fonction d'évaluation

La fonction de coût retenue pour la résolution par algorithme génétique est la somme des retards alloués $\text{cost}_{\text{sum}}$. C'est en effet ce critère qui a permis d'obtenir la meilleure convergence de l'algorithme (compromis entre temps de calcul et coût des solutions). D'un point de vue opérationnel, ce critère est intéressant car il re ète la performance globale du système d'allocation de créneaux.

Cependant, il est particulièrement difficile de construire des solutions qui respectent les contraintes liées aux con its. Il est donc nécessaire de prendre en compte ces contraintes dans la fonction d'évaluation, afin de pénaliser les solutions qui violent des contraintes. Nous introduisons à cet effet les variables booléennes suivantes, pour une paire de vols $i$ et $j$ donnée :

$$c_{ij} = \begin{cases} 1 & \text{si les vols } i \text{ et } j \text{ sont en con it} \\ 0 & \text{sinon} \end{cases}$$

c'est-à-dire, en reprenant l'expression des con its de l'équation 3.5 :

$$c_{ij} = \begin{cases} 1 & \text{si } d_{ij} \in \mathcal{C}_{ij} \\ 0 & \text{sinon} \end{cases} \tag{6.1}$$

Le nombre de con its non résolus s'exprime donc par :

$$c = \sum_{i<j} c_{ij}$$

On définit alors la fonction d'évaluation $F$ par :

$$F = \frac{1}{1+c}\left(n - \sum_{i=1}^{n} \frac{\delta_i}{\delta_{\max}}\right)$$

ou encore :

$$F = \frac{n - \frac{\text{cost}_{\text{sum}}}{\delta_{\max}}}{1+c}$$

Cette fonction d'évaluation croît lorsque le nombre de con its ou la somme des retards diminuent. D'autre part, cette formulation permet d'obtenir une évaluation bornée dans l'intervalle $[0, n]$, qui donne de meilleurs résultats lors du processus de sélection. L'algorithme génétique va chercher à maximiser cette fonction.

### 6.2.3 Population initiale

La population initiale est générée aléatoirement de la manière suivante : pour chaque variable $\delta_i$, une valeur est tirée aléatoirement (selon une loi uniforme) dans l'intervalle $[0, \delta'_{\max}]$, où $\delta'_{\max} \leq \delta_{\max}$ est un paramètre de l'initialisation. Une valeur élevée pour $\delta'_{\max}$ permet de trouver plus rapidement des solutions admissibles, mais la convergence vers un optimum est ensuite plus longue. Une valeur de 10 min a été retenue pour nos expérimentations, car cette valeur permet un bon compromis entre la recherche de solutions admissibles et le temps de convergence.

### 6.2.4 Opérateur de mutation

Pour chaque chromosome candidat à la mutation, le retard d'un des vols impliqués dans des con its est modifié. Dans le cas où tous les con its sont résolus, un vol est choisi au hasard et ses paramètres sont modifiés. Pour mettre en uvre cette opération, il est donc nécessaire de connaître le nombre de con its $c_i$ dans lesquels un vol $i$ donné est impliqué. Ce nombre s'exprime comme suit :

$$c_i = \sum_{\substack{j \in [1,n] \\ j \neq i}} c_{ij}$$

Un vol $i$ est dit *plus contraint* qu'un vol $j$ (on notera $i \succ j$) s'il est impliqué dans un plus grand nombre de con its ou s'il a un retard plus important en cas d'égalité :

$$i \succ j \quad \Leftrightarrow \quad c_i > c_j \vee (c_i = c_j \wedge \delta_i > \delta_j) \tag{6.2}$$

Pour chaque mutation, un nombre $m$ est pris aléatoirement dans l'intervalle $[1, \frac{n}{2}]$, et $m$ vols sont pris aléatoirement parmi les $n$. Parmi ces $m$ vols, le plus contraint au sens de la relation 6.2 est soit optimisé localement (résolution des con its dans lesquels il est impliqué ou diminution de son retard), soit modifié de manière aléatoire, avec une probabilité de 50 % pour chaque possibilité. L'algorithme 6.6 décrit la procédure de mutation utilisée.

**Entrées :** $\chi = \{\delta_1^\chi, ..., \delta_n^\chi\}$ un chromosome de la génération $g$
**Sorties :** chromosome $\chi'$ obtenu par mutation de $\chi$

1: $m \in [1, \frac{n}{2}]$
2: $\{i_1, ..., i_m\}$ // choix de $m$ vols parmi les $n$
3: $i_p \leftarrow i_1$
4: **pour** $k \in [2, m]$ **faire**
5: **si** $i_k \succ i_p$ **alors**
6: $i_p \leftarrow i_k$ // recherche du vol le plus contraint
7: **n si**
8: **n pour**
9: **si** $rand_{\mathbb{R}}([0; 1]) < 0,5$ **alors**
10: $\delta'_{i_p} \leftarrow$ résoudre$(i_p)$ // on tente de résoudre tous les con its impliquant le vol $i_p$
11: **sinon**
12: $\delta_{i_p} \leftarrow rand_{\mathbb{N}}([0, \delta_{\max}])$
13: **n si**
14: **renvoyer** $\chi' = \{\delta_1^\chi, ..., \delta_{i_p-1}^\chi, \delta'_{i_p}, \delta_{i_p+1}^\chi, ..., \delta_n^\chi\}$

**Algorithme 6.6:** Opération de mutation

Il serait possible de réaliser l'optimisation locale de manière systématique, afin d'obtenir de meilleurs éléments de population. Cependant, l'algorithme deviendrait trop déterministe et peu de place serait laissée à l'exploration de l'espace de recherche, menant ainsi à une convergence prématurée vers un optimum local.

Cet opérateur de mutation a été établi de manière empirique, afin d'accélérer la convergence tout en maintenant une part d'aléatoire suffisante à l'exploration de l'espace de recherche. L'optimisation fine de l'algorithme génétique ne rentre pas dans le cadre de cette étude.

### 6.2.5 Opérateur de croisement

Ce problème de résolution de con its est partiellement séparable (voir section 6.1). Nous utilisons cette propriété pour construire un opérateur de croisement qui permet d'augmenter la probabilité de construire des enfants « meilleurs », au sens de l'évaluation, que leurs parents.

Pour cela, nous définissons pour chaque variable du problème, c'est-à-dire les vols $i$, une fonction d'évaluation locale $F_i$ :

$$F_i = \frac{1 - \frac{\delta_i}{\delta_{\max}}}{1 + c_i} \tag{6.3}$$

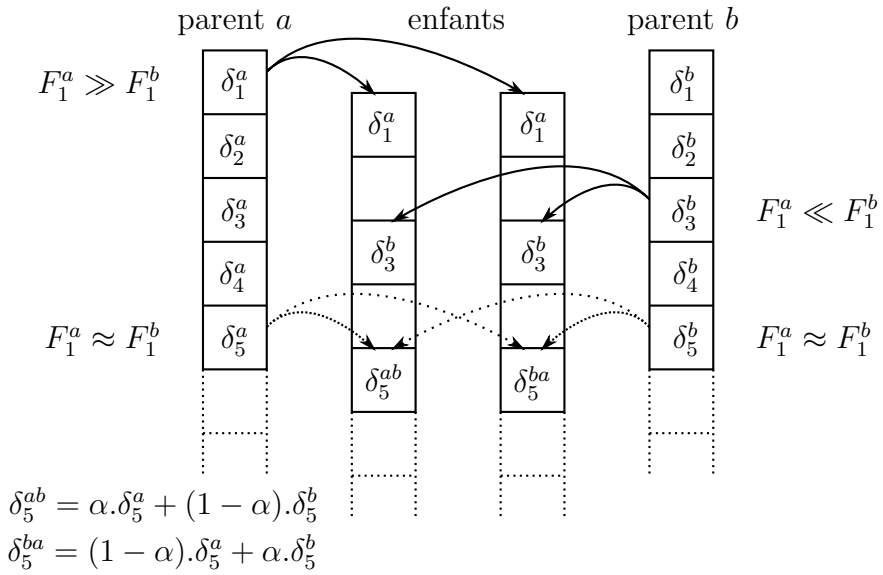



FIGURE 6.6 – Opérateur de croisement.

où :

$$c_i = \sum_{j \neq i} c_{ij}$$

correspond au nombre de con its non résolus impliquant le vol $i$. Ces évaluations locales sont utilisées pour construire l'opérateur de croisement illustré sur la figure 6.6. Pour un couple de parents $a$ et $b$, les évaluations locales $F_i^a$ et $F_i^b$ de chaque variable, c'est-à-dire de chaque vol $i$, sont comparées de telle manière que :

- si $F_i^a \ll F_i^b$ (resp. $F_i^b \ll F_i^a$), alors les deux enfants sont construits avec la valeur $\delta_i^b$ (resp. $\delta_i^a$) pour la variable $i$ ;
- si $F_i^a \approx F_i^b$, la valeur de la variable $i$ sera une combinaison linéaire des valeurs $\delta_i^a$ et $\delta_i^b$ pour chacun des enfants.

Ces opérateurs de mutation et de croisement sont moins aléatoires lors des premières générations de l'algorithme car il y a un nombre important de con its à résoudre, de telle manière que les solutions avec moins de con its auront une évaluation beaucoup plus élevée. Ces opérateurs se focalisent donc dans un premier temps sur la recherche de solutions admissibles. Dès que des situations sans con it apparaissent dans la population, mutation et croisement deviennent moins déterministes et laissent la place à la recherche d'un optimum.

### 6.2.6 Paramètres de l'algorithme génétique

Étant donné le temps d'exécution de notre algorithme génétique (voir section 8.2.1), il n'était pas envisageable d'utiliser des techniques d'optimisation des paramètres de celui-ci telles que décrites dans [Grefenstette 86] (paramètres statiques) ou [Eiben 99] (contrôle dynamique des paramètres pendant l'exécution). Les paramètres ont donc été choisis de

manière empirique, à l'aide de tests sur des problèmes de taille réduite et en se basant sur des études précédentes de problèmes de résolution de con its [Durand 96, Granger 02].

Les paramètres suivants ont été retenus :

- la taille de la population est fixée à $K = 100$ individus ;
- les populations seront générées avec un taux de croisement de 20 % et un taux de mutation de 60 % ;
- l'élitisme est utilisé, ce qui signifie que le meilleur individu d'une génération est systématiquement reproduit dans la génération suivante ;
- le nombre de générations est fixé à 500 (afin de limiter le temps de calcul), ce qui constitue le critère d'arrêt de l'algorithme.

D'autre part, la sélection utilise le processus de *stochastic remainder without replacement*, qui est adapté à la faible taille de la population, et le *scaling* est réalisé selon la méthode *sigma truncation*.

Le problème de résolution de con its par allocation de créneaux de décollage est fortement combinatoire et possède un grand nombre d'optima. Afin d'éviter une convergence prématurée, nous utilisons le procédé de *sharing clusterisé* décrit dans la section 6.1.2. La fonction de distance $d$ entre deux individus est définie par :

$$D(a, b) = \frac{1}{n} \sum_{i=1}^{n} |\delta_i^a - \delta_i^b|$$

Les distances $d_{\min}$ et $d_{\max}$ sont fixées à la valeur 1. Étant donnée la notion de distance définie et les valeurs retenues pour les paramètres $d_{\min}$ et $d_{\max}$, il n'est pas particulièrement intéressant de définir un barycentre qui ait un sens « géographique » relativement à l'espace d'état. Le centre d'un *cluster* pour le *sharing* sera arbitrairement choisi comme le premier individu du *cluster*.

Enfin, nous avons essayé d'améliorer les solutions en sortie de cet algorithme à l'aide d'une méthode de recherche locale stochastique de type recuit simulé [Kirkpatrick 83]. Cependant, les améliorations obtenues sont très faibles (moins de 1 %) malgré un nombre important d'itérations, ce qui nous a conduit à ne pas retenir cette possibilité.

Cet algorithme génétique a permis d'obtenir des résultats satisfaisants pour la résolution de con its dans l'espace aérien français (voir chapitre 8). Ces résultats ont par ailleurs été valorisés dans le cadre du projet « Episode 3 » de la Commission Européenne [Allignol 09]. Cependant, les temps d'exécution très importants nous ont conduit à envisager un autre paradigme, la programmation par contraintes, pour la résolution de ce problème. Les détails de cette méthode sont présentés dans le chapitre suivant.

# Chapitre 7

# Résolution en programmation par contraintes

La Programmation par Contraintes (PPC) est une technologie qui permet la modélisation et la résolution des problèmes de recherche et d'optimisation combinatoires de façon complète. Elle utilise des méthodes de résolution variées, issues notamment des domaines de la recherche opérationnelle et de l'intelligence artificielle, et est utilisée dans les domaines scientifique et industriel. Les domaines d'application de la PPC sont très variés, notamment à cause de l'expressivité des langages de modélisation associés : problèmes d'ordonnancement, de placement, d'affectation de ressources...

De précédentes études ont montré l'intérêt de la PPC pour la résolution de problèmes liés au trafic aérien. [Tran Dac 03] l'utilise pour modifier le découpage de l'espace aérien en secteurs de contrôles afin de minimiser la charge de travail des contrôleurs. [Dalichampt 97] propose différentes stratégies d'allocation de créneaux de décollage utilisant la PPC. [Flener 07] minimise la complexité du trafic dans un ensemble de secteurs où la complexité est exprimée en fonction du nombre d'avions dans le secteur ou près de ses frontières, en retardant les avions au sol ou en modifiant la vitesse ou l'altitude des avions en vol. [Barnier 02a] modélise plusieurs problèmes à l'aide de la PPC : respect des capacités des secteurs par allocation de créneaux de décollage, affectation de niveaux de vol dans un réseau de routes directes et optimisation des schémas d'ouverture des secteurs de contrôle.

Ce chapitre donne les notions nécessaires à la modélisation et la résolution des problèmes en PPC (section 7.1), puis présente les modèles en contraintes pour les problèmes d'allocation de créneaux de décollage sans con it (section 7.2) et d'allocation de niveaux de vol (section 7.3).

## 7.1 Principes de la programmation par contraintes

La PPC est un paradigme de programmation assez récent [Van Hentenryck 92], qui permet de rendre indépendants, d'une part la modélisation du problème à l'aide d'un langage de haut niveau, et d'autre part sa résolution à l'aide de « solveurs de contraintes ». Cette

section présente le cadre de la PPC ainsi que les méthodes de résolution des problèmes de satisfaction de contraintes. Le formalisme proposé ici reprend ceux présentés dans [Schiex 92, Van Hentenryck 95].

### 7.1.1 Principes généraux

La PPC est une technologie qui permet de résoudre des problèmes de recherche combinatoires exprimés sous la forme de Problèmes de Satisfaction de Contraintes (CSP). Les CSP sont exprimés en termes de *variables* (les inconnues du problème) et de *contraintes* qui limitent les combinaisons de valeurs que peuvent prendre ces variables. La résolution d'un CSP consiste à affecter une valeur aux différentes variables de telle manière que les contraintes soient respectées.

Un système de PPC consiste en un langage de modélisation expressif et un (ou plusieurs) solveur(s). L'intérêt de ce type de système réside dans l'indépendance entre le problème décrit à l'aide du langage et sa résolution. Il est ainsi possible de tester plusieurs stratégies de recherche à moindre coût, sans avoir à modifier la description du modèle. Certains systèmes (*e.g.* IBM ILOG CP) proposent en outre plusieurs types de solveurs et permettent donc de tester d'autres paradigmes en utilisant toujours la même description. Nous définissons dans cette section les notions principales de spécification d'un CSP utiles pour leur résolution.

#### Contraintes

Les contraintes sont les relations qui lient les variables du problème en restreignant les combinaisons de valeurs qu'elles peuvent prendre simultanément. Comme pour les problèmes de programmation mathématique, ces contraintes sont :

- déclaratives : elles ne décrivent pas de procédure chargée de les faire respecter ;
- non directionnelles : on ne distingue pas de variables d'entrée et de variables de sortie ;
- additives : toutes les contraintes d'un CSP doivent être satisfaites simultanément ; la propriété à vérifier est donc implicitement la conjonction de toutes les contraintes.

Ce qui fait l'intérêt de la modélisation des problèmes en CSP est la généralité de la notion de contrainte. Celle-ci permet, dans la plupart des solveurs PPC, de fournir à l'utilisateur un langage riche et de haut niveau, ainsi que des connecteurs logiques permettant de combiner les contraintes et d'en fabriquer de nouvelles. L'expression d'un problème se fait ainsi avec un langage proche de la formalisation du problème.

#### Domaines

La généricité de l'expression des contraintes permet d'avoir le même formalisme pour des problèmes portant sur des variables ayant des domaines de types variés. En effet, toutes les structures mathématiques dotées d'une théorie équationnelle décidable peuvent intervenir dans l'expression d'un CSP : booléens, entiers, réels, ensembles...

Pour la modélisation de notre problème, nous nous intéresserons exclusivement aux CSP sur les domaines finis. Pour chaque variable devra être précisé un domaine initial, *i.e.* les

valeurs que l'on peut donner à la variable lors de la recherche de solution. Il est également possible de donner le même domaine à toutes les variables et de représenter les restrictions initiales par des contraintes unaires.

**FaCiLe**

Les modèles présentés dans la suite de ce chapitre ont été implantés à l'aide de la librairie de programmation par contraintes FaCiLe (Functional Constraint Library) [Barnier 01a]. C'est une librairie *open source* écrite en Objective Caml, qui a été développée au LOG (Laboratoire d'Optimisation Globale [1]). Elle fournit la plupart des contraintes usuelles sur les domaines finis : contraintes arithmétiques linéaires et non linéaires, contraintes globales (« tous différents », contrainte de cardinalité), réification (ou « méta-contraintes » voir section suivante). Elle permet également à l'utilisateur de créer de nouvelles contraintes et de nouveaux buts de recherche grâce à une interface de haut niveau.

### 7.1.2 Problèmes de satisfaction de contraintes

Le cadre de la PPC est très générique et fait intervenir des domaines quelconques ainsi que des relations quelconques entre les variables. Nous donnons dans cette section le formalisme des CSP sur les domaines finis, ainsi que des méthodes constructives de résolution.

**Dé nition 9 (CSP sur les domaines nis)** *Un* CSP *sur les domaines nis se dé nit comme un triplet* $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$*, où :*

$\mathcal{X} = \{x_1, ..., x_n\}$ *est un ensemble ni de variables.*

$\mathcal{D} = (\mathcal{D}_1, ..., \mathcal{D}_n)$ *est un ensemble ni de domaines de cardinal ni. Le domaine* $\mathcal{D}_i$ *de la variable* $x_i$ *est l ensemble des valeurs que celle-ci peut prendre.*

$\mathcal{C} = \{c_1, ..., c_m\}$ *est l ensemble des contraintes. Chaque contrainte* $c_i = (\mathcal{X}_i, \mathcal{R}_i)$ *est dé nie par l ensemble* $\mathcal{X}_i \subseteq \mathcal{X}$ *des variables sur lesquelles elle porte et par une relation* $\mathcal{R}_i$ *dé nissant l ensemble des valeurs que peuvent prendre simultanément les variables de* $\mathcal{X}_i$*.*

Pour une contrainte $i$ donnée, le cardinal de l'ensemble $\mathcal{X}_i = \{x_{i_1}, ..., x_{i_{k_i}}\}$, $|\mathcal{X}_i| = k_i$, sera appelé l'*arité* de la contrainte (en particulier, on appelera contrainte *unaire* une contrainte d'arité un et contrainte *binaire* une contrainte d'arité deux). L'ensemble $\mathcal{R}_i$ définissant la relation entre les variables de $\mathcal{X}_i$ est un sous-ensemble du produit cartésien des domaines de chaque variable de $\mathcal{X}_i$ : $\mathcal{R}_i \subseteq \mathcal{D}_{i_1} \times ... \times \mathcal{D}_{i_{k_i}}$.

**Exemple 3 (CSP :** $x = y, y < z$**)** *Soient trois variables* $x \in \{1, 2\}, y \in \{1, 2, 4\}, z \in \{2, 3, 4\}$*. On cherche à instancier ces variables de telle manière que :* $x = y, y < z$*.*

1. Laboratoire commun à l'École Nationale de l'Aviation Civile (ÉNAC) et au CENA.

*Ce problème se modélise par le CSP suivant :*

$$\mathcal{P} \left\{ \begin{array}{lcl} \mathcal{X} & = & \{x, y, z\} \\ \mathcal{D} & = & \{\{1,2\},\{1,2,4\},\{2,3,4\}\} \\ \mathcal{C} & = & \left\{ \begin{array}{l} (\{x,y\},\{(1,1),(2,2)\}) \\ (\{y,z\},\{(1,2),(1,3),(1,4),(2,3),(2,4)\}) \end{array} \right\} \end{array} \right.$$

Dans la définition 9, ainsi que dans l'exemple précédent, les contraintes sont définies *en extension*, c'est-à-dire sous la forme d'une liste exhaustive de toutes les combinaisons possibles de valeurs. Cependant, la plupart des problèmes applicatifs utilisent des contraintes exprimées à l'aide de formules arithmétiques, d'équations avec des opérateurs d'ensembles, d'opérateurs logiques ou même de spécifications de plus haut niveau (*e.g.* « toutes ces variables sont différentes », souvent appelée *alldi* ). Ces dernières contraintes sont exprimées *en intention.* Ceci n'affecte pas la définition 9, puisque toute contrainte en intention peut s'exprimer sous la forme d'une contrainte en extension. Dans la notation d'une contrainte en intention, on pourra omettre l'ensemble des variables sur lesquelles elle porte, puisque celles-ci apparaissent dans l'expression de la contrainte. Les solveurs PPC proposent généralement ces contraintes en intention en tant que primitives, ainsi que des connecteurs logiques permettant de les combiner. Le langage ainsi obtenu est proche des notations algébriques et ensemblistes de haut niveau utilisées dans les premières étapes de la formalisation des problèmes et facilite donc leur modélisation.

Afin de s'abstraire du type des contraintes utilisées dans la définition d'un CSP, on associe à chaque contrainte une fonction qui indique si celle-ci est satisfaite pour une instanciation donnée.

**Dé nition 10 (Fonction de satisfaction)** *Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un CSP. On associe à chaque contrainte $c_i = (\mathcal{X}_i, \mathcal{R}_i) \in \mathcal{C}$ une* fonction booléenne de satisfaction *dé nie par :*

$$\begin{array}{rcl} \tilde{c}_i : \mathcal{D}_{i_1} \times ... \times \mathcal{D}_{i_{k_i}} & \longrightarrow & \{vrai, faux\} \\ (v_{i_1}, ..., v_{i_{k_i}}) & \longmapsto & \tilde{c}_i(v_{i_1}, ..., v_{i_{k_i}}) = \left\{ \begin{array}{ll} vrai & si(v_{i_1}, ..., v_{i_{k_i}}) \in \mathcal{R}_i \\ faux & sinon \end{array} \right. \end{array}$$

Nous verrons dans la section 7.1.3 que l'on n'a pas toujours à attendre que toutes les variables d'une contrainte $c_i$ soient instanciées pour savoir si celle-ci est satisfaite ou non, c'est-à-dire si $\tilde{c}_i = 0$ ou $\tilde{c}_i = 1$, notamment grâce au maintien de l'arc-consistance du CSP.

**Dé nition 11 (Instanciation)** *Soit $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un CSP.*

*On appelle* littéral *une a ectation de la forme $x_i = v_i$ avec $x_i \in \mathcal{X}, v_i \in \mathcal{D}_i$.*

*Une* instanciation totale *est un ensemble de littéraux, un pour chaque variable de $\mathcal{X}$.*

*Une* instanciation partielle *est un sous-ensemble d une instanciation totale. Pour une instanciation partielle $\mathcal{I}$, on note $\mathcal{X}_\mathcal{I}$ l ensemble des variables sur lesquelles elle porte.*

On utilisera la notation $\mathcal{I}(x_i) = v_i$ pour indiquer que la valeur $v_i$ est affectée à la variable $x_i$ par l'instanciation $\mathcal{I}$. On notera également $\tilde{c}_i(\mathcal{I})$ l'application de la fonction

de satisfaction de la contrainte $c_i$ à la restriction de $\mathcal{I}$ sur les variables de $\mathcal{X}_i$, si $\mathcal{X}_i \subseteq \mathcal{X}_\mathcal{I}$. On dira alors que l'instanciation $\mathcal{I}$ *satisfait* la contrainte $c_i$ si et seulement si $\tilde{c}_i(\mathcal{I}) = vrai$. Dans le cas contraire, on dira que $\mathcal{I}$ *viole* la contrainte.

Pour qu'une instanciation partielle $\mathcal{I}$ puisse mener à une solution, il faut que toutes les contraintes concernées par cette instanciation soient satisfaites. On dira alors que $\mathcal{I}$ est une *instanciation consistante*. Une solution d'un CSP est une instanciation totale et consistante.

**Réi cation** Il est possible de *relaxer* certaines contraintes lors de la modélisation d'un problème en CSP en utilisant des contraintes *réi ées*. La *réi cation* d'une contrainte $c$ consiste à ajouter une variable booléenne $b(c)$ contrainte à 1 si $c$ est satisfaite et à 0 si $c$ est violée. Dans ce cas, la contrainte $c$ n'est pas ajoutée à l'ensemble $\mathcal{C}$ des contraintes du CSP ; elle pourra l'être dynamiquement (ou sa négation) lors de la recherche. Il est en revanche possible d'utiliser les variables $b(c)$ dans de nouvelles contraintes.

**Exemple 4 (Réi cation)** *Soit $\{x_1, ..., x_n\}$ un ensemble de variables de domaines $\mathcal{D}_i = [1, n], \forall i \in [1, n]$. Pour une valeur $v$ donnée, la variable booléenne $b(x_i = v)$, correspondant à la réi cation de la contrainte $(x_i = v)$, vaut 1 si $x_i = v$, 0 sinon.*

*Cette réi cation permet par exemple de construire une contrainte communément appelée* contrainte de cardinalité*, qui compte le nombre de variables d un ensemble égales à une valeur $v$ :*

$$card(c, v, \{x_1, ..., x_n\}) \Leftrightarrow c = \sum_{i=1}^{n} b(x_i = v)$$

*Il existe d ailleurs un algorithme ad-hoc e cace pour cette contrainte.*

La réification permet également de relaxer des contraintes lors de la résolution de problèmes surcontraints et peut donc permettre de trouver des « bonnes » solutions là où la résolution aurait échoué. Elle augmente également l'expressivité d'un langage de contraintes, en permettant notamment d'écrire des disjonctions (voir exemple 5).

**Exemple 5 (Disjonction)** *Soient $c_1$ et $c_2$ deux contraintes. On peut par exemple exprimer la disjonction $c_1 \vee c_2$ grâce à la réi cation des contraintes $c_1$ et $c_2$ de la manière suivante :*

$$c_1 \vee c_2 \;\Leftrightarrow\; b(c_1) + b(c_2) \geq 1$$

### 7.1.3 Résolution de CSP

Il existe plusieurs types de méthodes pour résoudre des CSP : méthodes complètes, recherche locale, métaheuristiques... La méthode la plus communément adoptée en PPC consiste en une exploration de l'espace de recherche sous la forme d'un arbre de recherche, dont chaque n ud correspond à une instanciation partielle, et ses fils à l'extension de cette instanciation avec une nouvelle affectation, chaque fils correspondant à une valeur dans le domaine de la variable concernée. Il est également possible de voir un n ud comme un sous-problème du CSP, c'est-à-dire une restriction de celui-ci avec des contraintes supplémentaires.

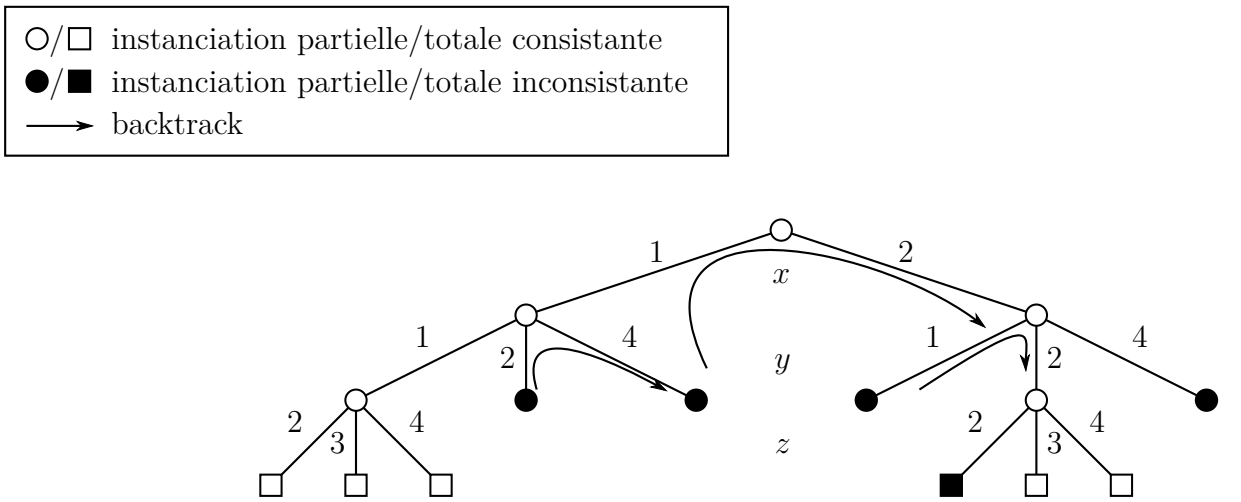



FIGURE 7.1 – Parcours d'un arbre de recherche par algorithme de *backtrack*.

### Backtrack

L'algorithme de *backtrack* est une sophistication de l'énumération systématique de toutes les combinaisons possibles. Le principe est de vérifier la satisfaction des contraintes à chaque n ud de l'arbre. Dès qu'une variable $x$ est instanciée, on vérifie la satisfaction de toutes les contraintes $c = (\mathcal{X}_c, \mathcal{R}_c)$ telles que $x \in \mathcal{X}_c$. Si une instanciation partielle $\mathcal{I}$ est inconsistante avec une contrainte, alors toutes ses extensions le seront également et $\mathcal{I}$ ne mènera donc à aucune solution. On peut alors élaguer le sous-arbre correspondant à cette instanciation en effectuant un *backtrack* (retour arrière) jusqu'à la première instanciation qui offre une autre branche non encore explorée. La figure 7.1 montre le parcours de l'arbre de recherche de solution pour le CSP de l'exemple 3 avec l'algorithme de *backtrack*.

**Entrées :** $\mathcal{N}$ ensemble des variables non instanciées, $\mathcal{I}$ instanciation partielle.
**Sorties :** une solution du CSP.

1: **si** $\mathcal{N} = \emptyset$ **alors**
2: **renvoyer** $\mathcal{I}$ // une solution
3: **sinon**
4: $x \in \mathcal{N}$ // choix d'une variable à instancier
5: **pour tout** $a \in \mathcal{D}_x$ **faire**
6: $\mathcal{I}' := \mathcal{I} \cup \{(x, a)\}$ // choix d'une valeur pour l'instanciation
7: **si** $\bigwedge_{c \in \mathcal{C} \wedge x \in \mathcal{X}_c \wedge \mathcal{X}_c \subseteq \mathcal{X}_{\mathcal{I}'}} \tilde{c}(\mathcal{I})$ **alors**
8: $BT(\mathcal{N} \setminus \{x\}, \mathcal{I}')$
9: **n si**
10: **n pour**
11: **n si**

**Algorithme 7.1:** Backtrack : $BT(\mathcal{N}, \mathcal{I})$

L'algorithme 7.1 décrit cette méthode de manière formelle en se basant sur le principe de l'énumération par force brute, mais en testant la consistance des contraintes dont les variables sont instanciées à chaque n ud. Afin de ne pas tester plusieurs fois des contraintes dont on sait qu'elles sont satisfaites à chaque nouvelle valeur essayée, on se contente de vérifier celles qui concernent la variable $x$ en cours d'instanciation, c'est-à-dire les contraintes $c$ qui vérifient :

$$c \in \mathcal{C} \wedge x \in \mathcal{X}_c \wedge \mathcal{X}_c \subseteq \mathcal{X}_{\mathcal{I}'}$$

On impose ici que toutes les variables de la contrainte soient instanciées avant de tester la satisfaction de celle-ci. Le problème sous-jacent à cette remarque est le suivant : on risque de se trouver dans le cas où un échec dû à une affectation d'une valeur à une variable est répété plusieurs fois dans plusieurs branches de l'algorithme de *backtrack*. Notamment :

1. Pour une variable $x_i$ de domaine $\mathcal{D}_i$ et soumise aux contraintes unaires (d'arité 1) $c_i^1, ..., c_i^k$, s'il existe une valeur $v \in \mathcal{D}_i$ qui ne satisfasse pas la conjonction $c_i^1 \wedge ... \wedge c_i^k$, alors il y a un risque de répéter des instanciations de $x_i$ à $v$ suivies d'échecs systématiques.
2. Pour deux variables $x_i$ de domaine $\mathcal{D}_i$ et $x_j$ de domaine $\mathcal{D}_j$, soumises aux contraintes $c_{ij}^1, ..., c_{ij}^l$ et telles qu'il existe une valeur $v \in \mathcal{D}_i$ telle que l'instanciation de $x_i$ à $v$ ne satisfasse la conjonction $c_{ij}^1 \wedge ... \wedge c_{ij}^l$ pour aucune valeur de $\mathcal{D}_j$, l'algorithme de *backtrack* va tester toutes les valeurs possibles pour $x_j$ avant de déterminer que $v$ n'est pas une valeur convenable pour $x_i$. Ces essais-erreurs se répèteront à chaque nouvelle tentative d'instanciation de variables instanciées avant $x_i$.

La section suivante introduit la notion de *consistance locale* ainsi que des techniques permettant d'exploiter celle-ci afin d'anticiper les échecs décrits précédemment.

### Propagation de contrainte et consistance locale

La consistance globale d'un CSP correspond à sa résolution. On peut également définir des propriétés de *consistance locale* [Mackworth 77] qui vont permettre d'anticiper des échecs tels que décrits dans la section précédente en inférant des contraintes qui ne modifient pas l'ensemble des solutions, mais qui vont permettre de simplifier le CSP en filtrant les domaines des variables.

Le filtrage le plus simple consiste à maintenir les domaines des variables de manière à ce que toutes les contraintes unaires soient satisfaites. Dans l'exemple 3, lorsque $x$ est instanciée à la valeur 1, alors les valeurs 2 et 4 peuvent être retirées du domaine de $y$, puisqu'aucune de ces deux valeurs ne satisfait la contrainte $x = y$. Ce filtrage, appelé *consistance de n ud* ou *1-consistance*, revient à anticiper l'énumération d'une étape.

**Dé nition 12 (1-consistance)** *Un CSP* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ *est* 1-consistant *si* $\forall x \in \mathcal{X}$ *tel que* $\mathcal{D}_x \neq \emptyset, \forall v \in \mathcal{D}_x, \forall c \in \mathcal{C}$ *telle que* $\mathcal{X}_c = \{c\}, \tilde{c}(\{x = v\}) = vrai$.

La consistance de n ud s'établit en parcourant l'ensemble des $n$ contraintes unaires [2] $c$

2. Toutes les contraintes unaires portant sur une même variable peuvent être agrégées en une seule contrainte dé nie comme la conjonction des précédentes. Il reste alors au plus une contrainte unaire par variable du CSP.

et en remplaçant le domaine de chaque variable $x$ concernée par $\mathcal{D}_x \cap \mathcal{R}_c$. Cette opération a donc une complexité en $\mathcal{O}(nd_{\max})$ où $n$ est le nombre de variables du CSP et $d_{\max}$ est la taille du plus grand domaine. D'autre part, si aucune contrainte n'est rajoutée dynamiquement au CSP, la consistance de n ud est établie une fois pour toutes au moment de poser les contraintes.

Un filtrage plus fort permet une anticipation de deux étapes sur l'énumération. Ce filtrage consiste, pour une contrainte binaire (*i.e.* d'arité 2) $c$ portant sur les variables $x$ et $y$, à supprimer du domaine de $x$ (resp. $y$) les valeurs qui ne mènent à aucune instanciation consistante possible pour $y$ (resp. $x$) vis-à-vis de la contrainte $c$. Dans l'exemple 3, la valeur 4 du domaine de la variable $y$ implique une violation de la contrainte $y < z$, puisqu'il n'existe aucune valeur dans le domaine de $z$ qui soit strictement supérieure à 4. Cette valeur peut donc être supprimée du domaine de $y$. Ce filtrage est appelé *consistance d arc* ou *2-consistance.*

**Dé nition 13 (2-consistance)** *Un CSP* $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ *est* 2-consistant *si* $\forall x, y \in \mathcal{X}^2$ *tels que* $\mathcal{D}_x \neq \emptyset$ *et* $\mathcal{D}_y \neq \emptyset, \forall c \in \mathcal{C}$ *telle que* $\mathcal{X}_c = \{x, y\}$ *:*

$$\begin{cases} \forall v_x \in \mathcal{D}_x, \exists v_y \in \mathcal{D}_y, (v_x, v_y) \in \mathcal{R}_c \\ \forall v_y \in \mathcal{D}_y, \exists v_x \in \mathcal{D}_x, (v_x, v_y) \in \mathcal{R}_c \end{cases}$$

L'algorithme 7.2 [Mackworth 77] rend un arc $(x, y)$ (*i.e.* une moitié de contrainte binaire[3]) consistant et renvoie un booléen qui indique si le domaine de la variable $x$ a été modifié par cette opération.

```
1: modif ← faux
2: pour tout v_x ∈ D_x faire
3:   si ∄v_y ∈ D_y tel que (v_x, v_y) ∈ R_(x,y) alors
4:     D_x ← D_x \ {v_x}
5:     modif ← vrai
6:   n si
7: n pour
8: renvoyer modif
```

**Algorithme 7.2:** révise$(x, y)$

L'établissement de la consistance d'arc d'un CSP est plus coûteux que celui de la consistance de n ud. En effet, le filtrage de la procédure « révise » peut être répété plusieurs fois pour chaque contrainte puisque la réduction du domaine d'une variable peut entraîner de nouveaux filtrages sur d'autres contraintes. Il faut donc répéter ce filtrage jusqu'à ce qu'un point fixe soit atteint, c'est-à-dire jusqu'à ce que plus aucun domaine ne puisse être réduit. De nombreux algorithmes ont été proposés pour établir l'arc consistance. Ces algorithmes sont pour la plupart désignés par AC-$m$ (pour *arc consistency*, $m$ représentant les évolutions successives de l'algorithme). Nous donnons ici la description

3. Il est en e et nécessaire d'appliquer révise$(x, y)$ *et* révise$(y, x)$.

de l'algorithme AC-3 [Mackworth 77] qui maintient un ensemble $\mathcal{Q}$ des arcs à réviser à l'aide de l'algorithme 7.2.

**Entrées :** $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un CSP
**Sorties :** 2-consistance de $\mathcal{P}$
1: $\mathcal{Q} \leftarrow \{(x, y), \exists c = (\{x, y\}, \mathcal{R}_c) \in \mathcal{C}\}$
2: **tant que** $\mathcal{Q} \neq \emptyset$ **faire**
3: $(x, y) \in \mathcal{Q}$ // choix d'un arc
4: $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{(x, y)\}$
5: **si** révise$(x, y)$ **alors**
6: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(z, x), \exists c = (\{z, x\}, \mathcal{R}_c) \in \mathcal{C}, z \neq y\}$
7: **n si**
8: **n tant que**

**Algorithme 7.3:** AC-3

L'ensemble $\mathcal{Q}$ est initialisé avec l'ensemble des arcs du CSP. À chaque étape, on retire de $\mathcal{Q}$ un arc $(x, y)$ qui est rendu consistant et, si le domaine de $x$ a été modifié, on y replace les arcs de la forme $(z, x), z \neq y$.

L'algorithme 7.3, dont la complexité est en $\mathcal{O}(md^3_{\max})$ ($m = |\mathcal{C}|$ étant le nombre de contraintes), a été amélioré, notamment par [Bessière 93] qui propose l'algorithme AC-6 dont la complexité est en $\mathcal{O}(md^2_{\max})$. [Régin 05] présente les multiples améliorations des ces algorithmes (notamment des améliorations de AC-3 et AC-6) et propose un nouvel algorithme générique et adaptatif, AC-*, qui combine les différents algorithmes de consistance d'arc existants.

De la même manière que nous avons défini la consistance de n ud (1-consistance) et la consistance d'arc (2-consistance), il est possible de définir une propriété plus générale de $k$-consistance ($k \in [1, n]$) qui signifie que pour toute instanciation partielle consistante $\mathcal{I}$ de taille $k - 1$ et pour chaque variable $x$ non instanciée par $\mathcal{I}$, il existe au moins une valeur $v$ dans le domaine de $x$ qui mène à une instanciation $\mathcal{I} \cup \{(x, v)\}$ consistante. Cette propriété forte permet en général un filtrage efficace, mais l'établissement de la $k$-consistance pour $k > 2$ est très coûteuse à mettre en uvre.

La propagation de contrainte consiste à utiliser des algorithmes d'arc-consistance lors de la recherche de solution par *backtrack* : à chaque nouvelle instanciation, les domaines sont filtrés grâce aux contraintes induites par l'établissement de l'arc-consistance, comme illustré sur la figure 7.2 pour la résolution du problème de l'exemple 3.

De nombreux autres raffinements permettent d'améliorer l'algorithme de *backtrack* afin d'accélérer la recherche de solutions. On peut citer notamment les techniques de *backjumping* [Gaschnig 79, Dechter 90, Prosser 93], qui permettent d'exploiter la structure des contraintes pour « remonter » plus haut dans l'arbre de recherche lors d'un *backtrack* et éviter d'échouer de manière répétée pour la même raison (phénomène de *thrashing*), ou les méthodes d'apprentissage de contraintes telles que le *nogood recording* [Schiex 93], qui consistent à ajouter des contraintes de manière dynamique sur les échecs rencontrés lors de la recherche.

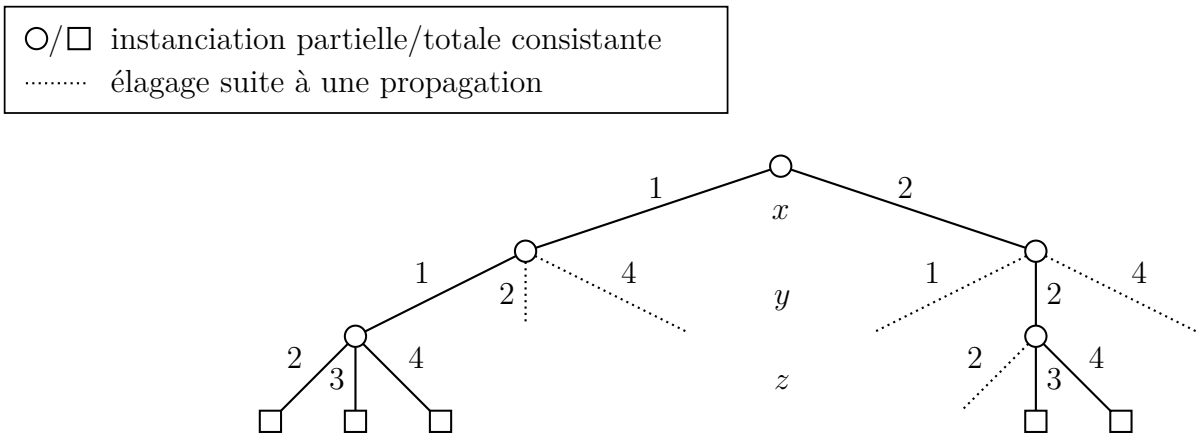



FIGURE 7.2 – *Backtrack* et propagation de contraintes pour la recherche de solutions.

### Heuristiques d'instanciation

L'algorithme de *backtrack* tel que nous l'avons décrit dans les sections précédentes n'indique aucune information quant à l'ordre d'instanciation des variables (ligne 4 de l'algorithme 7.1) ou des valeurs (ligne 6). L'ordre d'instanciation a pourtant une in uence déterminante sur l'efficacité de la recherche de solution, puisque la quantité de filtrage (*i.e.* de réductions de domaine) dépend de la décision prise. Il existe des stratégies standards pour choisir l'ordre d'instanciation des variables, notamment :

- `dom` [Haralick 80] : instanciation de la (des) variable(s) ayant le plus petit domaine en premier ;
- `deg` [Dechter 89] : instanciation de la (des) variable(s) impliquée(s) dans le plus grand nombre de contraintes en premier. Cette stratégie est rarement utilisée seule, mais plus souvent comme second critère (au début de la recherche, notamment, il est courant que toutes les variables aient le même domaine et un second critère est utilisé pour les départager).

Ces deux stratégies correspondent à l'idée générale d'instancier d'abord les variables les plus contraintes ou qui laissent le moins de « marge de man uvre » en ce qui concerne la taille des domaines. Elles sont souvent combinées afin de départager des variables pour lesquelles l'un des critères a la même valeur. L'heuristique `dom+deg` [Frost 95] choisit la variable de plus petit domaine en premier et celle de plus grand degré en cas d'égalité. L'heuristique `dom/deg` [Bessière 96] choisit en premier la variable $x$ qui minimise le rapport $|\mathcal{D}_x|/|\mathcal{V}_x|$ où $\mathcal{V}_x = \{y \in \mathcal{X}, \exists c \in \mathcal{C}, (x,y) \in \mathcal{X}_c\}$ est l'ensemble des voisins de $x$ dans le graphe représentant le CSP (*i.e.* le graphe dont les n uds sont les variables et les arcs les contraintes). Contrairement à `dom+deg` qui donne une priorité au critère de taille des domaines, l'heuristique `dom/deg` utilise les deux critères de façon égalitaire.

Il est également possible d'utiliser des heuristiques plus sophistiquées, prenant en compte certains aspects statiques ou dynamiques du problème à résoudre (*e.g.* ressource la plus critique dans un problème d'ordonnancement). L'expertise de l'utilisateur peut alors permettre de construire des heuristiques *ad hoc* très efficaces.

**Optimisation**

Les méthodes de résolution présentées précédemment permettent de trouver une solution ou toutes les solutions d'un CSP, ou bien de prouver que celui-ci n'a pas de solution. Il est toutefois possible de modifier ces algorithmes afin de trouver la meilleure solution au regard d'un critère donné en utilisant le même principe qu'un algorithme de *branch & bound*.

Une variable modélisant le coût est ajoutée au CSP, avec pour domaine initial l'intervalle $[inf, sup]$ où $inf$ (resp. $sup$) est la borne inférieure (resp. supérieure) du coût si elle est connue, ou l'ensemble des entiers représentables sinon (en général, la phase de propagation initiale permet de déduire un intervalle pour le coût quand celui-ci est défini par une fonction des variables de décision). À chaque solution de coût $c$ trouvée, on tronque la partie supérieure (dans le cas d'une minimisation) du domaine de la variable de coût :

- soit à $c - 1$ si on cherche les solutions strictement meilleures : c'est la seule méthode pour être sûr de trouver l'optimal ;
- soit à $c - step$ ou $c(1 - p)$ si on cherche les solutions meilleures avec un écart (absolu d'une valeur $step$ ou relatif avec un pourcentage $p$) minimum par rapport à la solution courante de coût $c$.

Puis l'algorithme de *backtrack* poursuit la recherche, en repartant du n ud le plus bas permettant de trouver une meilleure solution, ou alors en repartant à la racine de l'arbre de recherche. La recherche se poursuit alors jusqu'à ce que l'espace soit entièrement parcouru, ou qu'une solution de coût $inf$ soit trouvée.

## 7.2 Allocation de créneaux de décollage en PPC

La résolution du problème d'allocation de créneaux de décollage sans con it à l'aide de la PPC présente deux enjeux particulièrement intéressants. Du point de vue du problème, l'utilisation de la PPC peut permettre, selon les performances de l'implémentation réalisée, d'obtenir une preuve d'optimalité de la solution trouvée. Du point de vue de la PPC, il est intéressant de déterminer si ce paradigme permet la résolution d'un problème combinatoire de cette taille (plusieurs milliers de variables, plusieurs centaines de milliers de contraintes, voire plusieurs millions pour des instances sur l'Europe). Cette section propose une représentation du problème d'allocation de créneaux en CSP et des techniques utilisées pour sa résolution.

### 7.2.1 Expression du CSP

La PPC permet d'exprimer les problèmes avec un langage de haut niveau, proche des formulations mathématiques. De ce fait, le CSP correspondant au problème d'allocation de créneaux sans con it s'exprime simplement à partir de la définition du problème donnée dans le chapitre 4 (définition 6). Cette section décrit ce CSP pour un problème à $n$ vols.

### Variables et domaines

Les variables de décision du problème sont les retards liés aux vols :

$$\Delta = \{\delta_i, i \in [1, n]\}$$

où chaque retard $\delta_i$ a pour domaine initial :

$$\mathcal{D}_{\delta_i} = [0, \delta_{\max}]$$

Nous utilisons également les variables auxiliaires $d_{ij} = \delta_j - \delta_i$, sur lesquelles sont exprimées les contraintes du problème. Ces variables ont pour domaine initial :

$$\mathcal{D}_{d_{ij}} = \left[\inf \mathcal{D}_{\delta_j} - \sup \mathcal{D}_{\delta_i}, \sup \mathcal{D}_{\delta_j} - \inf \mathcal{D}_{\delta_i}\right] = [-\delta_{\max}, \delta_{\max}]$$

### Contraintes

Les contraintes du modèle sont directement utilisables pour l'écriture du CSP, et s'expriment donc :

$$\forall (i, j) \in [1, n]^2, i < j, \quad d_{ij} \notin \mathcal{C}_{ij}$$

Ces contraintes peuvent également être exprimées sous la forme de contraintes disjonctives sur les variables de décision :

$$d_{ij} \notin \mathcal{C}_{ij} \Leftrightarrow \left\{ \begin{array}{rll} (-\delta_{\max} \leq & \delta_j - \delta_i & < \underline{lb}_1) \vee \\ (\overline{ub}_1 < & \delta_j - \delta_i & < \underline{lb}_2) \vee \cdots \vee \\ (\overline{ub}_{\sigma-1} < & \delta_j - \delta_i & < \underline{lb}_\sigma) \vee \\ (\overline{ub}_\sigma < & \delta_j - \delta_i & \leq \delta_{\max}) \end{array} \right. \tag{7.1}$$

à condition que $\underline{lb}_1 > -\delta_{\max}$ et $\overline{ub}_\sigma < \delta_{\max}$ ; dans le cas contraire, la première partie ou la dernière partie de cette disjonction sont supprimées. De même, lorsque $\overline{ub} < -\delta_{\max}$ ou $\underline{lb} > \delta_{\max}$, le con it n'est pas pris en compte.

### CSP

En reprenant le formalisme de la définition 9, le CSP $\mathcal{P}$ représentant le problème d'allocation de créneaux de décollage sans con it peut donc être exprimé sous la forme :

$$\mathcal{P} = \left\{ \begin{array}{rcl} \mathcal{X} & = & \{\delta_i, i \in [1, n]\} \cup \{d_{ij}, (i, j) \in [1, n]^2, i < j\} \\ \mathcal{D} & = & [0, \delta_{\max}]^{(n)} \cup [-\delta_{\max}, \delta_{\max}]^{(n(n-1)/2)} \\ \mathcal{C} & = & \{(d_{ij} = \delta_j - \delta_i) \wedge (d_{ij} \notin \mathcal{C}_{ij}),\ (i, j) \in [1, n]^2, i < j\} \end{array} \right.$$

où on note $\mathcal{I}^{(n)} = \{\underbrace{\mathcal{I}, ..., \mathcal{I}}_{n \text{ fois}}\}$ pour un intervalle $\mathcal{I}$ donné.

Comme mentionné précédemment, les $d_{ij}$ sont des variables auxiliaires définies à partir des variables de décision $\delta_i$.

### 7.2.2 Optimisation

Le critère d'optimisation retenu pour la résolution en PPC est le maximum des retards alloués :

$$\text{cost}_{\max} = \max_{i \in [1,n]} \delta_i$$

À chaque solution trouvée de coût $c$, la recherche continue avec comme nouvelle contrainte : $\text{cost} \leq c - 1$ afin de trouver la solution optimale.

Ce critère a été retenu plutôt que la somme des retards car il permet une propagation efficace des contraintes pour l'optimisation ainsi qu'une preuve d'optimalité pour la quasi totalité des instances considérées. En effet, le problème étant fortement combinatoire, il existe de nombreuses solutions possibles pour chaque valeur de la somme des retards, et les contraintes ajoutées sur le coût à chaque nouvelle solution trouvée ne permettent pas une propagation efficace. Ceci est d'autant plus vrai que la somme des retards prend des valeurs élevées (dans l'intervalle $[0, n\delta_{\max}]$), de sorte qu'une diminution de valeur 1 sur le coût à chaque nouvelle solution n'élague pas suffisamment l'arbre de recherche. Le retard maximum alloué, au contraire, prend peu de valeurs (dans l'intervalle $[0, \delta_{\max}]$), et sa diminution d'une valeur 1 à chaque solution trouvée est bien plus significative. L'intérêt de ce critère d'un point de vue opérationnel (équité entre les compagnies aériennes et utilisation du résultat pour montrer la faisabilité de la méthode) a par ailleurs déjà été évoqué dans le chapitre 4.

### 7.2.3 Stratégies de recherche

Comme indiqué dans la section 7.1.3, l'ordre d'instanciation des variables et des valeurs a une in uence forte sur les performances de la recherche de solution du CSP. Cette section présente deux stratégies de recherche mises en uvre dans le cadre de l'allocation de créneaux de décollage.

#### Stratégie basique

La stratégie la plus simple pour la recherche de solution consiste à instancier directement les variables de décision. Pour cette stratégie, nous utilisons une heuristique d'instanciation `dom+deg` : la variable de plus petit domaine est choisie pour l'instanciation, et en cas d'égalité celle avec le plus grand nombre de contraintes. Cette heuristique permet de traiter en priorité les variables les plus contraintes.

Nous avons vu dans la section précédente que le critère d'optimisation est le maximum des retards alloués. Or la quantité totale de retard alloué sur la journée de trafic considérée constitue également une mesure importante de la qualité des solutions. Nous utilisons donc une heuristique simple d'instanciation des valeurs qui permet de maintenir la somme des retards à une valeur la plus faible possible : à chaque instanciation, la valeur la plus proche de 0 est essayée en premier.

**Stratégie inspirée des techniques d'ordonnancement**

Les contraintes du problème telles qu'exprimées dans l'équation 7.1 sont similaires à des contraintes disjonctives d'exclusion mutuelle, souvent utilisées pour modéliser les problèmes d'ordonnancement [Baptiste 01]. Nous proposons une heuristique inspirée des techniques de résolution de ce type de problème, qui consiste à instancier d'abord les variables $d_{ij}$, puis les variables $\delta_i$ comme décrit dans la suite.

Les secteurs de contrôle peuvent être vus comme des machines dotées d'une capacité telle que donnée par la définition 1. Nous détaillons ce modèle agrégé de manière à ce que chaque con it soit considéré comme une machine de capacité unitaire sur laquelle on doit effectuer deux tâches de durées différentes (*i.e.* le temps de passage dans la zone de con it, qui dépend de la vitesse de l'avion). Plusieurs con its sur une même trajectoire peuvent être interprétés comme les tâches ordonnées d'un même travail, de la même manière que dans un *Jobshop Scheduling Problem (JSP)*.

Cependant, la comparaison s'arrête à ces considérations. En effet, l'intervalle de temps qui sépare deux con its (donc deux tâches) sur une même trajectoire est fixe, puisqu'un seul retard est associé à chaque vol, alors que dans le JSP, les tâches ne sont soumises qu'à des contraintes de précédence. D'autre part, le fait de considérer un con it potentiel en trois dimensions comme la clôture transitive des segments en con it avec une durée de tâche proportionnelle au temps de parcours de la zone de con it mène à une modélisation beaucoup trop grossière (voir section 3.2.1). En effet, dans ce cadre, le con it associé à deux vols empruntant la même route serait la trajectoire toute entière, ce qui interdirait au deuxième vol de décoller avant que le premier ait atterri. Notre modèle est bien plus précis et permet à deux avions empruntant la même route d'être séparés de 5 NM au mieux. Enfin, le nombre de machines (*i.e.* nombre de con its), même s'il n'est pas quadratique en fonction du nombre de travaux (*i.e.* nombre de vols) comme il pourrait l'être pour une instance arbitraire du problème, est toutefois très grand, comme le montre la figure 4.1 (environ 8 000 vols et 300 000 con its).

Le schéma de branchement présenté dans cette section est malgré tout inspiré des techniques standards d'ordonnancement, car les contraintes disjonctives du problème présentent des similitudes avec celles des problèmes d'ordonnancement. Le principe est de prendre des décisions qui permettront de faciliter la propagation des contraintes arithmétiques. Ces décisions vont faire en sorte que le domaine des variables $d_{ij}$ soit un intervalle.

De manière similaire aux techniques de séparation utilisées pour les problèmes d'ordonnancement, où les tâches exécutées sur la même machine sont ordonnées par paire (soit la tâche $A$ est exécutée avant la tâche $B$, soit $B$ est exécutée avant $A$), nous ajoutons soit la contrainte $d_{ij} < \underline{lb}$, soit la contrainte $d_{ij} > \overline{ub}$ dans le cas d'un intervalle de con it unique. S'il y a plusieurs intervalles interdits dans le domaine de $d_{ij}$, ce schéma est répété avec les bornes des intervalles restants. Cette stratégie est appliquée récursivement sur toutes les variable $d_{ij}$ non instanciées, en choisissant à chaque étape celle qui a la plus faible densité, *i.e.* le plus petit rapport entre le cardinal du domaine et la différence des bornes du domaine.

Afin de compenser le fait que le coût soit défini comme le retard maximal alloué, sans

tenir compte de la somme des retards, nous choisissons d'instancier la variable $d_{ij}$ à une valeur dans l'intervalle qui induit potentiellement le moins d'augmentation pour les variables de retard $\delta_i$ et $\delta_j$ associées. Une valeur élevée de $d_{ij}$ implique nécessairement qu'au moins l'un des deux retards soit grand, donc nous sélectionnons en premier l'intervalle le plus proche de zéro, ce qui laisse l'opportunité de choisir des retards faibles (sans exclure que $\delta_i$ et $\delta_j$ soient grands simultanément). Lorsque la recherche revient en arrière dans l'arbre de recherche (*backtrack*) sur une décision de ce type, cet intervalle est écarté puis on sélectionne récursivement l'intervalle suivant (le plus proche de zéro).

Lorsque tous les con its sont ordonnés et qu'il n'y a plus de « trous » dans les domaines des $d_{ij}$, alors la stratégie présentée dans la section précédente est appliquée pour l'instanciation des variables de décision.

### 7.2.4 Contraintes supplémentaires

L'un des avantages de la PPC est qu'il est facile de rajouter des contraintes au modèle de base, afin de rendre celui-ci plus réaliste. Notamment, dans notre problème, il est possible de prendre en compte des contraintes liées au temps de rotation des avions ou à la correspondance des vols.

**Rotations** Le temps de rotation est le temps écoulé entre l'atterrissage d'un appareil et son décollage pour un autre vol. Cette période correspond notamment aux tâches suivantes : débarquement des passagers, ménage de l'avion, remplissage des réservoirs de carburant, changement d'équipages, embarquement des passagers du vol suivant. Si les vols $i$ et $j$ sont réalisés avec le même avion, où le vol $i$ précède le vol $j$, on peut ajouter une contrainte :

$$\begin{aligned} \tau_j^d + \delta_j &\geq \tau_i^a + \delta_i + TR_{ij} \\ \Leftrightarrow d_{ij} &\geq \tau_i^a - \tau_j^d + TR_{ij} \end{aligned}$$

où $TR_{ij}$ est le temps de rotation, et $\tau_i^d$ et $\tau_i^a$ sont les heures prévues de décollage et d'atterrissage de l'avion $i$ respectivement. Le temps de rotation peut dépendre du type d'avion (plus l'avion est grand, plus le temps de rotation est important), de la compagnie qui l'exploite (souvent, les compagnies *low cost* ont un temps de rotation plus faible) ou du type d'escale (carburant, changement d'équipage...).

**Correspondances** Les correspondances sont les liaisons entre vols dues à l'organisation en *hub* des réseaux des compagnies aériennes. Le *hub* est une sorte de point central du réseau, de telle sorte qu'un passager souhaitant se rendre d'un aéroport $A$ à un aéroport $B$ (tous deux différents de l'aéroport $H$ qui sert de *hub*) va souvent devoir faire une escale par l'aéroport $H$ pour ensuite prendre un vol entre $H$ et $B$. Il est en outre probable que plusieurs correspondances se fassent avec le même vol $HB$, pour des passagers provenant d'aéroports $A_1, ..., A_p$. En règle générale, les correspondances sont calculées de manière à ce que les passagers n'aient pas à attendre trop longtemps à l'escale. En revanche, si un des vols $A_xH$ est en retard, la compagnie va retarder le vol $HB$ afin d'attendre les

passagers[4]. On peut donc poser comme contraintes, pour tous les vols $i_1, ..., i_p$ ayant une correspondance avec le vol $j$ :

$$\begin{aligned} \forall x \in [1,p] \text{ tel que } \delta_{i_x} \leq \delta_{corr}, \quad \tau_j^d + \delta_j \geq & \quad \tau_{i_x}^a + \delta_{i_x} + T_{corr} \\ \Leftrightarrow d_{i_x j} \geq & \quad \tau_{i_x}^a - \tau_j^d + T_{corr} \end{aligned}$$

où $T_{corr}$ est le temps nécessaire à la correspondance et $\delta_{corr}$ est le retard au-delà duquel la compagnie aérienne n'attend plus les passagers du vol retardé. Encore une fois, ces valeurs sont dépendantes de la compagnie aérienne et du type et de la fréquence du vol en correspondance. Le temps de correspondance peut également dépendre de l'aéroport servant de *hub* à la compagnie aérienne.

Ces deux types de contraintes ont été implantés dans notre modèle, mais ne sont pas utilisées dans nos simulations. En effet, les données d'utilisation des appareils sur les différents vols, ainsi que celles concernant les retards liés aux correspondances sont particulièrement stratégiques pour les compagnies aériennes et sont donc confidentielles.

## 7.3 Allocation de niveaux de vol en PPC

Cette section présente la modélisation en CSP du problème d'allocation de niveaux de vol ainsi que les méthodes de résolution mises en uvre. Un des intérêts de la PPC dans le cadre de ce problème réside dans la possibilité de relaxation de certaines contraintes, permettant ainsi d'établir un compromis entre le nombre de con its résolus et l'écart maximal autorisé par rapport au RFL. Il est également intéressant de pouvoir utiliser des stratégies de recherche variées, notamment dans la phase de développement. Nous proposons une première représentation en CSP sur les bases du modèle présenté dans le chapitre 5, puis un modèle avec contraintes réifiées permettant de relaxer une partie d'entre elles.

### 7.3.1 Expression du CSP

Les CSP représentant le problème d'allocation de niveaux de vol s'exprime de manière très similaire au modèle décrit dans la section 5.2.

Les variables sont les niveaux de vol pour chaque vol $i$ :

$$\mathcal{X} = \{FL_i,\ i \in [1,n]\}$$

avec $FL_i \in [RFL_i - \Delta_{\max},\ RFL_i + \Delta_{\max}]$ pour chaque vol, *i.e.* le niveau de vol alloué est au plus distant de $\Delta_{\max}$ du RFL. Il est également possible d'effectuer un raffinement pour chaque vol en fonction des performances de l'avion notamment.

Les contraintes sont issues du modèle présenté dans la section 5.4 (voir équation 5.3) et s'expriment sous la forme :

$$[-\delta_{\max}, \delta_{\max}] \cap \mathcal{C}_{ij}^H \neq \emptyset \ \Rightarrow\ FL_i \neq FL_j$$

4. Ceci est vrai pour des retards relativement faibles et selon le type de vol.

c'est-à-dire que s'il existe une valeur faisable de $d_{ij}$ qui appartienne à l'ensemble des valeurs interdites (*i.e.* il y a un con it potentiel), alors les niveaux de vol de $i$ et $j$ doivent être différents.

On obtient donc le CSP suivant :

$$\mathcal{P} = \begin{cases} \mathcal{X} & = & \{FL_i,\ i \in [1,n]\} \\ \mathcal{D} & = & \{[RFL_i - \Delta_{\max},\ RFL_i + \Delta_{\max}],\ i \in [1,n]\} \\ \mathcal{C} & = & \left\{FL_i \neq FL_j,\ (i,j) \in [1,n]^2, i < j, [-\delta_{\max}, \delta_{\max}] \cap \mathcal{C}_{ij}^H \neq \emptyset\right\} \end{cases}$$

Ce modèle a été testé avec une heuristique `dom` pour l'ordre d'instanciation des variables et en choisissant d'instancier en premier la valeur la plus proche du RFL. Cependant, les expérimentations ont mené à une preuve de non existence de solution pour des valeurs « raisonnables » de $\Delta_{\max}$ ($\Delta_{\max} \leq 30$) sur toutes les instances traitées.

Afin d'être en mesure de proposer une allocation de niveaux de vol « intéressantes » nous proposons donc un modèle réifié de ce CSP qui permettra de faire un compromis entre le nombre de con its résolus et l'écart par rapport au RFL.

**Modèle réi é** Si on note $c_{ij}$ la contrainte entre deux vols $i$ et $j$ distincts tels que $[-\delta_{\max}, \delta_{\max}] \cap \mathcal{C}_{ij}^H \neq \emptyset$ :

$$c_{ij} = (FL_i \neq FL_j)$$

alors on rajoute au CSP précédent les variables booléennes $b(c_{ij}) \in \{0,1\},\ \forall (i,j) \in [1,n]^2, i < j$. Toutes les contraintes du modèle étant réifiées, l'ensemble $\mathcal{C}'$ des contraintes de ce second modèle est vide. Ce CSP réifié s'exprime donc sous la forme :

$$\mathcal{P}' = \begin{cases} \mathcal{X}' & = & \{FL_i,\ i \in [1,n]\} \cup \{b(c_{ij}),\ (i,j) \in [1,n]^2, i < j\} \\ \mathcal{D}' & = & \{[RFL_i - \Delta_{\max},\ RFL_i + \Delta_{\max}],\ i \in [1,n]\} \cup \{0,1\}^{(n(n-1)/2)} \\ \mathcal{C}' & = & \emptyset \end{cases}$$

Pour les raisons évoquées au chapitre 5, nous effectuons une relaxation temporelle des contraintes en imposant $\delta_{\max} = 0$ (ou $\delta_{\max} \leq 5\,\text{min}$) pour l'allocation de niveaux de vol. Cette relaxation permet d'obtenir des solutions beaucoup moins coûteuses, même si elles ne garantissent pas que tous les con its soient effectivement évités après allocation des créneaux de décollage.

### 7.3.2 Optimisation

L'utilisation d'un modèle réifié nous a conduit à modifier le critère d'optimisation présenté dans le chapitre 5 :

$$\text{cost}_{FL} = \sum_{i=1}^{n} |RFL_i - FL_i|$$

ou un coût plus précis prenant en compte la consommation des appareils (section 5.5).

En effet, afin d'orienter la recherche vers des solutions admissibles, notre critère doit prendre en compte les con its non résolus. À cet effet, nous définissons pour chaque vol $i$ un coût partiel $\text{pcost}_i$ correspondant au nombre de con its non résolus dans lesquels il est impliqué :

$$\text{pcost}_i = \sum_{j \neq i} b(c_{ij})$$

L'optimisation sera alors réalisée en minimisant le nombre de vols impliqués dans au moins un con it, c'est-à-dire selon le critère :

$$\text{cost}'_{FL} = \sum_{i=1}^{n} M(\text{pcost}_i)$$

avec :

$$\forall k \in \mathbb{N}, \begin{cases} M(k) = 0 \text{ si } k = 0 \\ M(k) = 1 \text{ sinon} \end{cases}$$

Lorsque $\text{cost}'_{FL} = 0$, alors tous les con its sont résolus. Les écarts au RFL seront maintenus à des valeurs faibles par l'heuristique d'instanciation des valeurs (voir section suivante). Ce critère repose sur l'idée qu'il est plus intéressant de séparer un vol du reste du trafic plutôt que de résoudre le même nombre de con its mais répartis sur plusieurs vols. On pourrait également envisager de résoudre une « tranche » (entre deux niveaux de vol donnés) de trafic.

### 7.3.3 Résolution

La stratégie de résolution de ce CSP consiste à choisir en premier la variable $FL_i$ de plus petit domaine (heuristique `dom`) et à « durcir » les contraintes associées au vol correspondant à cette variable, c'est-à-dire instancier toutes les variables $b(c_{ij}), j \neq i$ à la valeur 1.

Cette instanciation implique que les con its impliquant le vol $i$ soient résolus : si l'instanciation est possible, alors l'algorithme de recherche continue avec la variable suivante de plus petit domaine, sinon il y a un *backtrack*.

Une fois que toutes les variables $b(c_{ij}), j \neq i$ sont instanciées, la variable $FL_i$ sont instanciées à leur tour, en choisissant en premier les valeurs les plus proches possibles de $RFL_i$. Cette heuristique permet de maintenir le critère $\text{cost}_{FL}$ à une valeur faible.

Les résultats d'allocation de niveaux de vol et de créneaux de décollage obtenus avec les méthodes décrites dans les chapitres 6 et 7 sont présentés dans le chapitre suivant.

# Chapitre 8

# Résultats et analyses

Les modèles présentés dans les chapitres précédents ainsi que les techniques de résolution ont tous été implémentés en OCaml, en utilisant la librairie FaCiLe pour les modèles en contraintes. Les validations de ces modèles et de leur combinaison ont été faites sur plusieurs journées de trafic simulé dans l'espace aérien français en août et octobre 2008, avec des instances contenant jusqu'à 8700 vols. Des résultats préliminaires sur l'Europe sont également présentés et montrent que notre approche est susceptible d'être appliquée à l'échelle continentale. Environ 10 % des vols ont un aéroport de départ hors de la zone CEAC et se verront donc attribuer un retard nul (voir section 4.3.3).

La section 8.1 présente les résultats obtenus pour la phase d'allocation de niveaux de vol en montrant l'intérêt de ne pas agréger les vols en  ux pour cette allocation. La section 8.2 montre les résultats de la phase d'allocation de créneaux de décollage seule, puis de sa combinaison avec l'allocation de niveaux de vol. Des résultats préliminaires sur la prise en compte d'incertitudes sont exposés en fin de chapitre.

## 8.1 Phase d'allocation de niveaux de vol

Les performances de l'allocation de niveaux de vol se mesurent grâce à deux critères : le nombre de con  its qui ont pu être résolus et les écarts par rapport au RFL nécessaires à la résolution de ces con  its. Cette section présente les résultats obtenus pour les implémentations des deux modèles présentés dans le chapitre 5, c'est-à-dire avec ou sans agrégation des vols en  ux.

### 8.1.1 Allocation par  ux de vols

Cette section présente les résultats de l'allocation de niveaux de vol obtenus avec le modèle classique d'agrégation des vols en  ux (section 5.3), incluant les raffinements présentés dans la section 5.4.2.

Le tableau 8.1 donne quelques mesures des instances traitées : nombre de vols, nombre de  ux, nombre de contraintes et coût de la meilleure solution trouvée.

TABLE 8.1 – Taille des instances et coût de l'allocation de niveaux de vol avec agrégation en ux.

| Journée | 08/12 | 08/13 | 08/14 | 10/06 | 10/07 | 10/08 | 10/10 |
|---|---|---|---|---|---|---|---|
| Vols | 8130 | 8204 | 8384 | 8297 | 7941 | 8021 | 8624 |
| Flux | 2921 | 3065 | 3168 | 3146 | 3112 | 3156 | 3406 |
| Con its | 16579 | 17775 | 17641 | 18671 | 16613 | 16515 | 19717 |
| $\text{cost}_{FL}$ | 42550 | 47360 | 43910 | 41330 | 34770 | 36480 | 38530 |

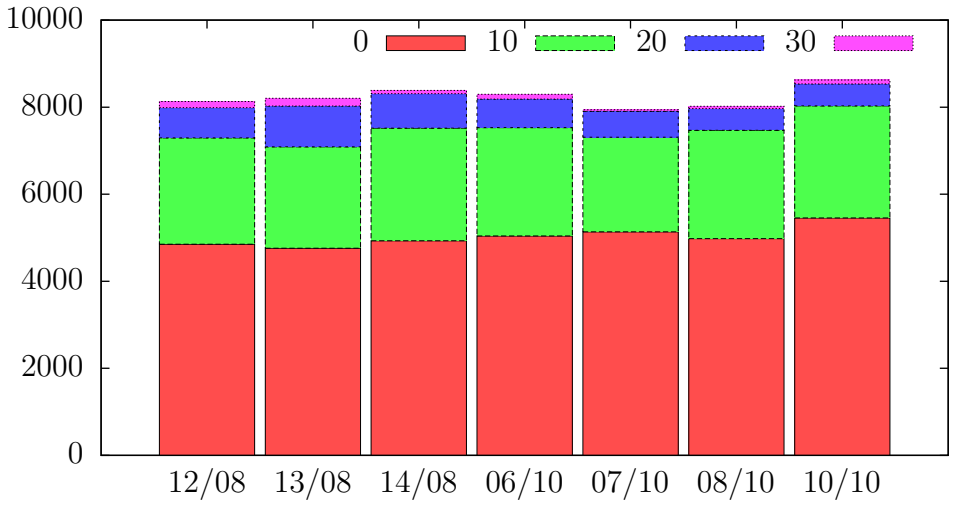



FIGURE 8.1 – Répartition des écarts par rapport au RFL après allocation avec agrégation en ux.

Ces résultats sont obtenus avec $\Delta_{\max} = 30$, ce qui signifie que chaque ux peut être affecté jusqu'à trois niveaux de vol au-dessus ou au-dessous de son RFL. Cette valeur est relativement faible, mais a néanmoins permis d'obtenir une allocation sans con it pour toutes les instances considérées. La figure 8.1 montre la répartition des écarts par rapport au RFL : environ 60 % des vols obtiennent leur RFL, 10 à 20 % sont un niveau au-dessus ou au-dessous et seulement 1 à 5 % se voient affecter un écart de trois niveaux de vol par rapport à leur RFL.

Le coût $\text{cost}_{FL}$ n'est pas directement optimisé par la résolution du CSP, puisque le modèle réifié est utilisé (voir chapitre 7). Cependant, l'heuristique d'instanciation retenue permet de maintenir ce coût à une valeur faible : un calcul approximatif en supposant que tous les avions sont des A320 montre une augmentation d'environ 1 % de la consommation globale de carburant.

Afin de valider l'allocation et produire les données de trajectoires nécessaires à la phase d'allocation de créneaux de décollage, les solutions obtenues sont fournies au simulateur de trafic CATS, simultanément aux plans de vols. La méthode de détection de con its présente dans ce simulateur permet de mesurer l'efficacité de l'allocation de niveaux de vol.

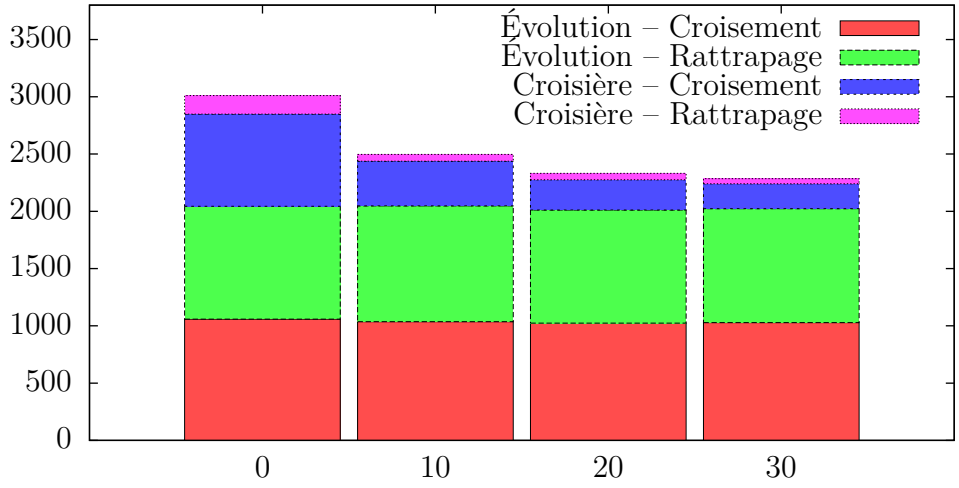



FIGURE 8.2 – Nombre de con its restants en fonction de $\Delta_{\max}$ le 14/08/2008 pour l'allocation avec agrégation en ux.

La figure 8.2 montre le nombre de con its détectés par CATS en fonction de la valeur de $\Delta_{\max}$ utilisée pour l'allocation. Ces con its sont catégorisés en plusieurs types, selon qu'ils ont lieu en croisière (les deux vols impliqués sont en croisière) ou en évolution (au moins l'un des deux vols impliqués est en phase ascendante ou descendante) et en rattrapage (les avions suivent le même tronçon de route) ou en convergence.

Comme prévu, le nombre total de con its diminue lorsque $\Delta_{\max}$ augmente, c'est-à-dire lorsqu'une plus grande marge de man uvre est laissée lors de l'optimisation. Le nombre de con its en évolution évolue très peu, puisque ceux-ci ne sont pas pris en compte par l'allocation de niveaux de vol. La diminution observée correspond à des légères modifications des phases de transition entre montée et croisière ou entre croisière et descente.

On observe néanmoins une diminution de 20 % du nombre total de con its lorsqu'on n'autorise qu'un seul niveau d'écart ($\Delta_{\max} = 10$). La résolution de la totalité des contraintes du modèle nécessite $\Delta_{\max} = 30$, induisant alors une réduction de 27 % du nombre total de con its observés.

Les résultats obtenus sont satisfaisants, mais la volonté de traiter les con its en rattrapage dès la phase d'allocation de niveaux de vol nous a conduit à envisager un modèle où les vols ne sont plus agrégés en ux. C'est l'objet du second modèle que nous avons mis en uvre et dont les résultats sont présentés dans la section suivante.

### 8.1.2 Allocation par vol

Cette section présente les résultats obtenus en utilisant notre deuxième modèle d'allocation de niveaux de vol (section 5.4), avec $\Delta_{\max} = 30$ et sans prendre en compte le retard maximal (sauf lorsqu'un autre paramétrage est précisé).

La figure 8.3 montre la répartition des écarts entre niveau de vol attribué et RFL. On

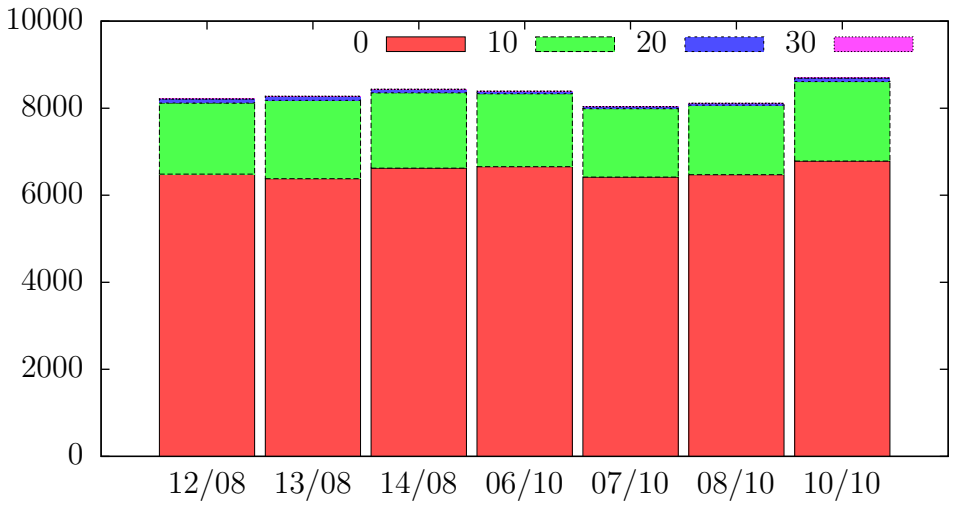



FIGURE 8.3 – Répartition des écarts par rapport au RFL après allocation par vol.

remarque que les résultats obtenus avec ce modèle sont meilleurs que ceux obtenus avec le modèle précédent, avec presque 80 % des vols qui obtiennent leur RFL et environ 20 % décalés d'un niveau seulement. Le coût des solutions ($\text{cost}_{FL}$) est en moyenne 55 % meilleur que celui du modèle précédent.

La figure 8.4 montre le nombre de con its détectés par le simulateur CATS pour différentes valeurs de $\Delta_{\max}$ utilisées pour l'affectation de niveaux. On remarque qu'autoriser un écart d'un seul niveau de vol permet d'éviter la quasi totalité des con its en croisière. D'autre part, et contrairement au modèle précédent, les con its en rattrapage sont bien pris en compte lors de ces résolutions. La résolution de l'ensemble des contraintes nécessite également un écart maximal de $\Delta_{\max} = 30$ pour la plupart des instances traitées (certaines sont résolues avec $\Delta_{\max} = 20$). On remarque une légère augmentation du nombre de con its en évolution pour $\Delta_{\max} = 30$ (par rapport à $\Delta_{\max} = 20$) ; ceci est dû au fait que les trajectoires sont modifiées avec l'affectation de niveaux de vol, ce qui peut induire de nouveaux con its, notamment des con its en phase évolutive non traités par l'affectation de niveaux. La figure 8.5 montre le nombre total de con its observés avec CATS, avec et sans allocation de niveaux de vol ($\Delta_{\max} = 30$) pour quelques journées de trafic dans l'espace aérien français. Bien sûr, comme pour le modèle avec agrégation en ux, seuls les con its en croisière sont évités grâce à la séparation en niveaux de vol. Les autres con its, en moyenne 75 % du nombre total de con its, devront être résolus par la seconde phase de notre méthode de planification, *i.e.* l'allocation de créneaux de décollage, dont les résultats sont présentés dans la section suivante.

Les résultats précédents utilisent la relaxation temporelle présentée dans la section 5.4.4. La figure 8.6 montre l'in uence de la prise en compte de l'aspect temporel des con its. Les coûts présentés sont associés à la résolution de l'ensemble des contraintes de l'instance. On observe que le coût de l'allocation de niveaux de vol croît de manière linéaire avec $\delta_{\max}$ pour des faibles valeurs de ce paramètre. D'un point de vue opérationnel, il ne serait pas

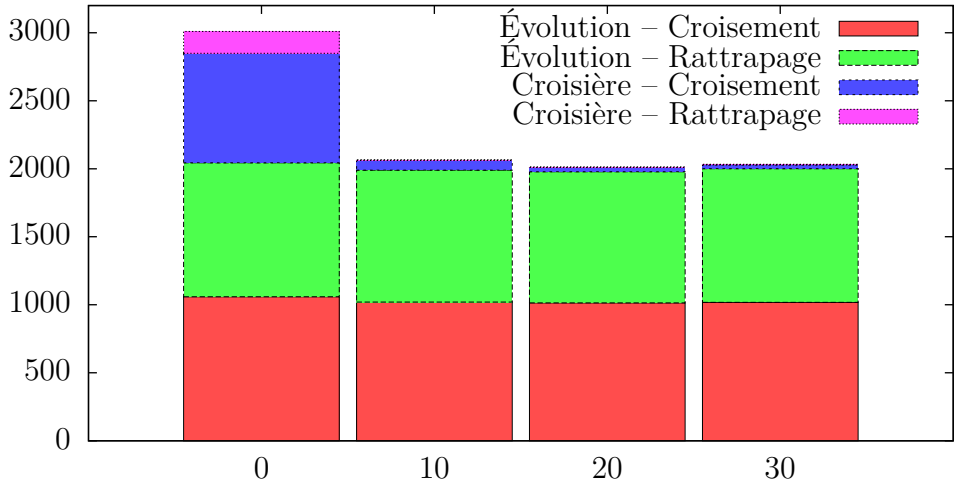



FIGURE 8.4 – Nombre de con its restants en fonction de $\Delta_{\max}$ le 14/08/2008 pour l'allocation par vol.

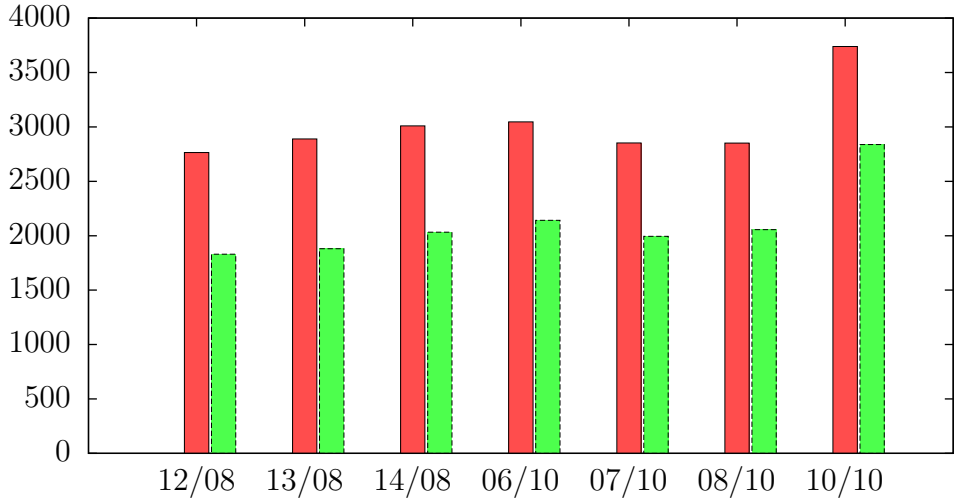



FIGURE 8.5 – Nombre de con its restants avant (en rouge) et après (en vert) l'allocation par vol.

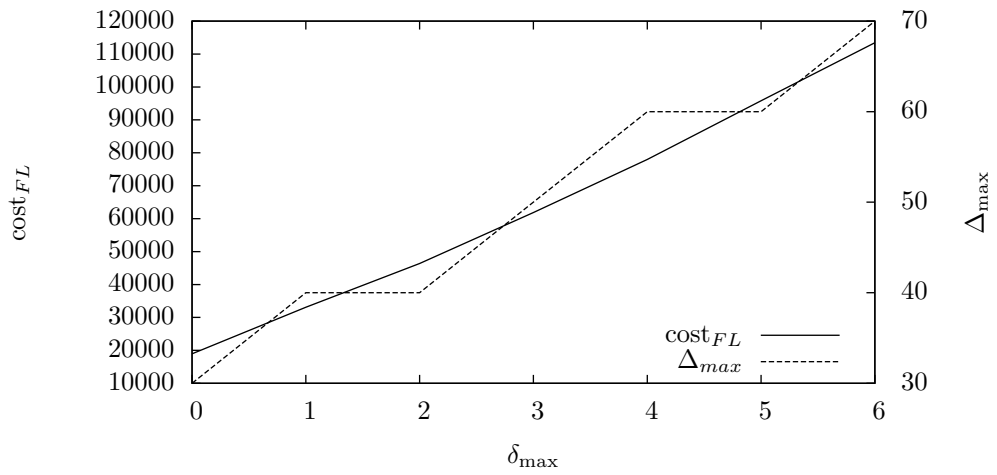


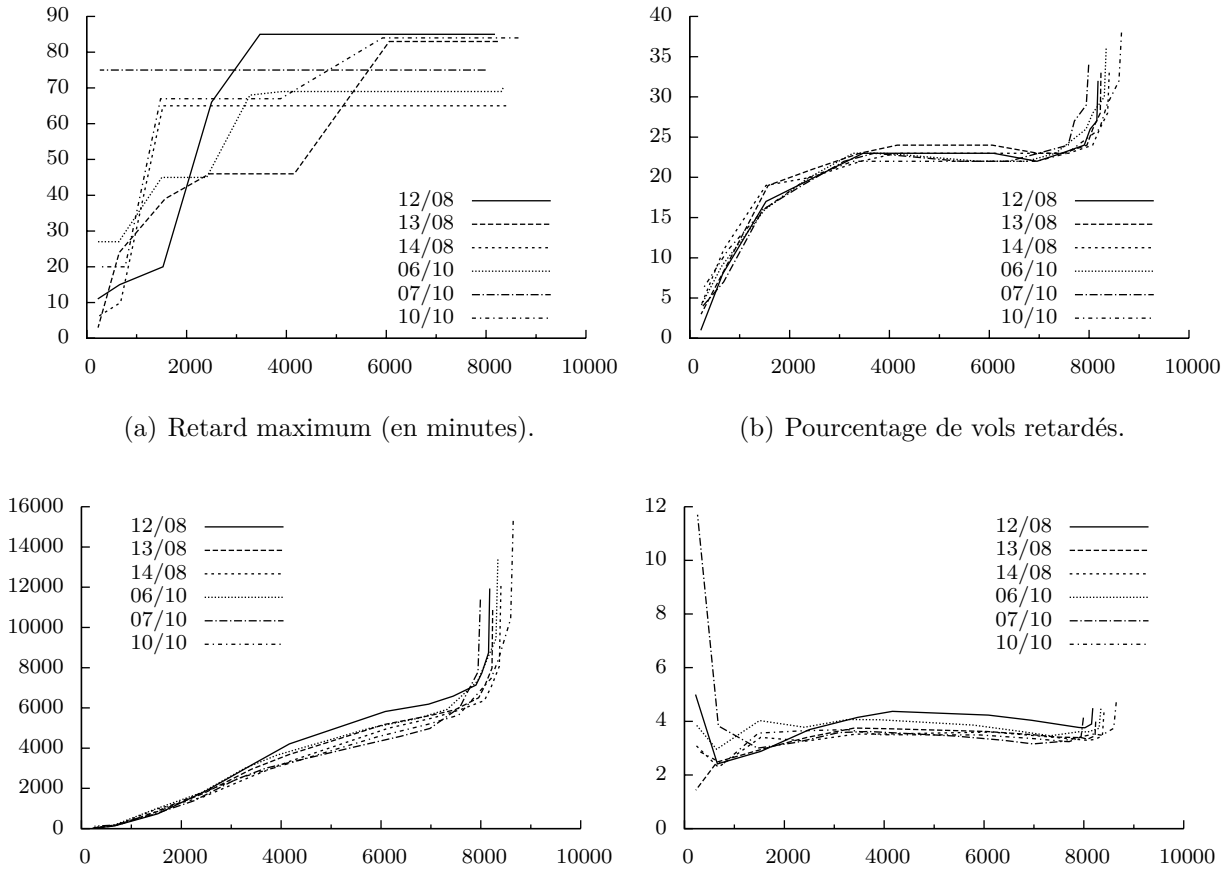

FIGURE 8.6 – Prise en compte de l'aspect temporel pour l'allocation de niveaux de vol : coût (cost$_{FL}$, échelle de gauche) et écart maximal au RFL ($\Delta_{\max}$, échelles de droite) en fonction de la valeur de $\delta_{\max}$.

envisageable de prendre en compte des valeurs de $\delta_{\max}$ supérieures à 3 minutes en raison de l'augmentation du coût de l'allocation.

## 8.2 Phase d'allocation de créneaux de décollage

L'allocation de créneaux de décollage a été testée avec succès sur de nombreuses instances, c'est-à-dire qu'elle a permis la résolution de la totalité des con its. La performance de cette allocation se mesure principalement par rapport au retard total alloué pour l'instance (cost$_{\text{sum}}$), même si le critère d'optimisation retenu pour le modèle CSP est le coût cost$_{\max}$ pour les raisons déjà évoquées dans le chapitre 4. En effet, le retard maximum alloué est corrélé à un sous problème particulièrement dense et « difficile », tandis que la somme des retards alloués mesure une performance globale.

La section 8.2.1 présente quelques résultats de l'allocation de créneaux seule et compare les différentes méthodes de résolution envisagées afin de justifier le choix du modèle en programmation par contraintes. La section 8.2.2 compare les résultats obtenus sur quelques instances traitées en routes standards et en routes directes afin de montrer l'in uence du choix du mode de navigation (routes standards ou directes, voir section 1.1.2) sur les retards alloués. La section 8.2.3 donne une indication de l'évolution du coût des solutions dans le cadre d'une modification de la valeur de la norme de séparation horizontale. La section 8.2.4 montre l'in uence de l'allocation de niveaux de vol sur les performances de l'allocation de créneaux de décollage. Enfin, la section 8.2.5 présente une analyse de sensibilité ainsi que des résultats préliminaires sur la prise en compte d'incertitudes.




(a) Retard maximum (en minutes).

(b) Pourcentage de vols retardés.

(c) Somme des retards (en minutes).

(d) Retard moyen (en minutes) par vol retardé.

FIGURE 8.7 – Résultats de l'allocation de créneaux de décollage en PPC en fonction de la taille de l'instance (nombre de vols) pour quelques journées de trafic de 2008.

### 8.2.1 Évaluation de l'allocation de créneaux

Nous avons testé la méthode d'allocation de créneaux de décollage sans con it sur plusieurs journées de trafic dans l'espace aérien français. Cette section présente les résultats obtenus sans allocation préalable de niveaux de vol, afin de pouvoir évaluer par la suite les gains apportés par cette dernière. Nous présentons dans un premier temps les résultats du modèle CSP que nous avons retenu pour la combinaison avec l'allocation de niveaux de vol. Nous donnerons par la suite des résultats comparatifs du modèle CSP et de l'algorithme génétique afin de justifier ce choix.

#### Résultats du modèle en contraintes

La figure 8.7(a) montre la valeur du critère d'optimisation, *i.e.* le retard maximum alloué. Cette valeur correspond au retard nécessaire à la résolution du sous problème le

plus « difficile » de l'instance. On remarque que cette valeur ne diminue significativement que pour les plus petites instances, ce qui signifie que les con its difficiles à résoudre ont lieu aux altitudes de croisière des vols réguliers (la taille de l'instance étant ici directement liée à l'altitude minimale de filtrage, voir figure 4.1 dans le chapitre 4).

Afin d'observer la performance globale de l'allocation de créneaux en termes de retards attribués, la figure 8.7(c) montre la quantité totale de retard allouée en fonction de la taille de l'instance. Cette quantité croît proportionnellement à la taille de l'instance, excepté pour les instances les plus grandes qui correspondent au traitement de la totalité du trafic (*i.e.* $FL_{\min} = 0$). On peut ainsi facilement remarquer que le retard moyen par vol est d'environ une minute pour les instances jusqu'à 7500 vols.

La figure 8.7(d) montre le retard moyen alloué *par vol retardé*. On observe que ce critère évolue peu avec la taille de l'instance [1], avec une valeur entre 3 et 5 min, car la quantité totale de retard allouée augmente de manière similaire au nombre de vols retardés (voir le pourcentage de vols retardés sur la figure 8.7(b)).

Ces chiffres sont compatibles avec les retards alloués actuellement par la CFMU. On rappelle en effet que les retards dus aux mesures de régulation ATFCM touchent actuellement 7,5 % des vols (ce pourcentage est en réalité plus élevé, puisque seuls sont comptés dans ce chiffre les vols retardés de plus de 5 min) pour un retard moyen de 21 minutes par vol retardé. La résolution des instances testées donne les résultats suivants : environ 23 % de vols retardés pour un retard moyen de 3 à 5 min par vol retardés. Le pourcentage de vols retardés serait encore bien plus faible, environ 12 %, en comptant celui-ci de la même manière que la CFMU (un vol est considéré comme retardé si son retard est supérieur à 5 min). On pourra s'en convaincre en se référant à la figure 5.3, page 77, qui montre la répartition des retards attribués par notre allocation sur une journée de trafic.

La figure 8.8 montre les temps de calcul nécessaires à la résolution de quelques instances et à la preuve de l'optimalité, obtenus sur un processeur Intel Xeon cadencé à 2,66 GHz. Ces temps sont de l'ordre de la minute pour les plus grandes instances et de quelques secondes sur les plus petites. Pour quelques rares instances, la solution optimale est trouvée dans un temps relativement faible (moins d'une minute) mais la preuve d'optimalité n'est obtenue qu'après une heure. Il est possible de limiter le temps de calcul ou le nombre de *backtracks* si nécessaire (*e.g.* lors d'une résolution en fenêtre glissante, on peut limiter le temps de calcul à la valeur du paramètre $\gamma$ du décalage).

### Choix d'une méthode de résolution

Le tableau 8.2 compare les résultats du modèle CSP et de l'algorithme génétique pour la journée du 12/08/2008 dans l'espace aérien français.

Le retard maximum alloué est similaire dans les deux approches (il est proche de la contrainte $\delta_{\max} = 90$ min), et son optimalité est prouvée dans le cas du CSP. Pour certaines instances, cette valeur est significativement meilleure avec le modèle CSP car l'algorithme

1. Les valeurs observées pour les plus petites instances (jusqu'à 300 vols environ) ne peuvent pas être interprétées de manière satisfaisante car la moyenne n'est pas représentative étant donné le faible nombre de vols retardés (souvent moins de dix pour ces instances).

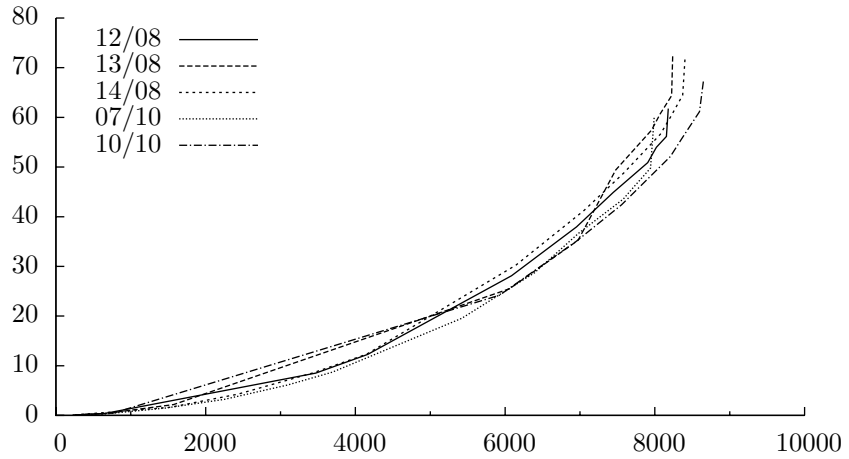



FIGURE 8.8 – Temps de calcul (s) de l'allocation de créneaux de décollage en fonction du nombre de vols pour quelques journées de trafic.

TABLE 8.2 – Comparaison des résultats du modèle CSP et de l'algorithme génétique (AG) sur une journée de trafic pour plusieurs valeurs de l'horizon glissant $H_r$. $H_r$ est exprimé en heures, les retards ainsi que les temps de calcul sont en minutes.

| $H_r$ | $\text{cost}_{\text{max}}$ | | $\text{cost}_{\text{sum}}$ | | Retard moyen/ vol retardé | | Vols retardés | | Temps CPU | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CSP | AG | CSP | AG | CSP | AG | CSP | AG | CSP | AG |
| 2 | 84 | 88 | 4782 | 11755 | 3,44 | 10,68 | 22% | 18% | 40,6 | 1619,2 |
| 3 | 84 | 88 | 4739 | 13731 | 3,41 | 12,52 | 23% | 18% | 47,2 | 2877,9 |
| 6 | 84 | 88 | 4785 | 17103 | 3,45 | 15,66 | 22% | 18% | 101,6 | 7839,3 |
| 24 | 84 | 89 | 4677 | 35590 | 3,38 | 33,08 | 22% | 17% | 0,4 | 8452,6 |

génétique ne converge pas vers la valeur optimale dans un temps raisonnable. Cependant, cette valeur n'est pas représentative de la performance globale d'une méthode de résolution et n'est donc pas le paramètre qui oriente le choix de celle-ci.

La quantité totale de retard allouée est en revanche un critère important. Or le modèle CSP donne des résultats 3 à 7 fois meilleurs en fonction des paramètres utilisés lors de la résolution, pour des temps de calcul [2] incomparablement plus faibles. En effet, le maintien de structures permettant de bons croisements et mutations dans l'algorithme génétique pénalisent lourdement le temps de calcul. La limite imposée (500 générations) ne permet pas une convergence suffisamment aboutie en comparaison de ce que permet la PPC dans notre cas.

L'algorithme génétique permet malgré cela d'obtenir un taux de vols retardés plus faible que le CSP. Ce phénomène est dû aux opérateurs de croisement et de mutation, qui orientent la recherche vers les vols les plus contraints (voir chapitre 6), laissant autant que possible les vols non contraints avec un retard nul. La combinaison d'une quantité totale de retard plus élevée avec un taux de vols retardés plus faible explique que le retard moyen par vol retardé soit beaucoup plus élevé (jusqu'à dix fois supérieur) avec l'algorithme génétique. Pour ces raisons, nous avons par la suite abandonné la résolution par algorithme génétique au profit de la PPC.

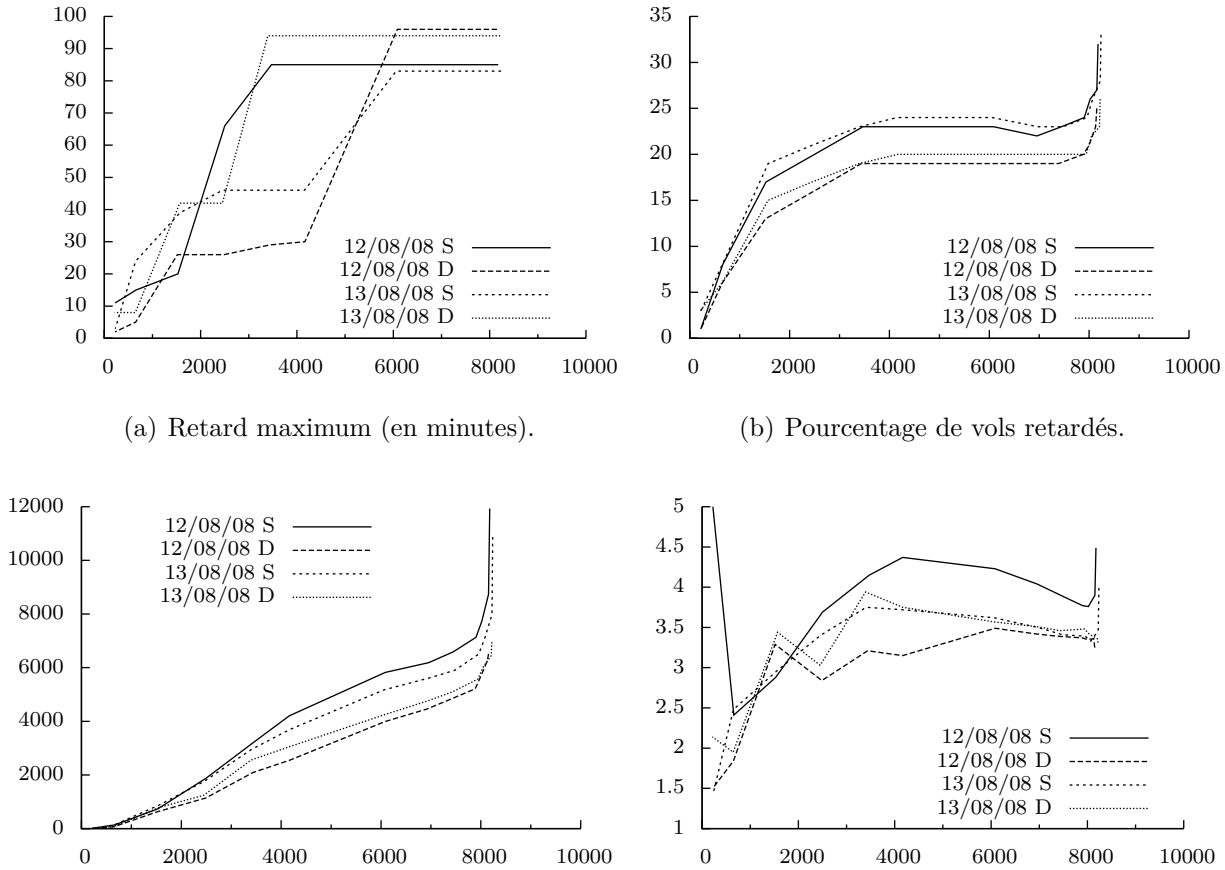
### 8.2.2 Comparaison des performances en routes standards et en routes directes

Nous avons souhaité évaluer les gains potentiels liés à l'utilisation de routes directes plutôt que de routes standards. La figure 8.9 montre les résultats de l'allocation de créneaux de décollage (sans allocation préalable de niveaux de vol) pour deux journées de trafic dans l'espace aérien français en routes standards et en routes directes.

On observe que le retard maximal attribué (figure 8.9(a)) varie peu et n'est pas systématiquement plus faible en routes directes qu'en routes standards. En revanche, la proportion de vols retardés (figure 8.9(b)) ainsi que la somme des retards alloués (figure 8.9(c)) sont sensiblement plus faibles en routes directes : une somme entre 20 et 30 % meilleure et en moyenne 5 % de vols retardés en moins. Ces résultats s'expliquent principalement par la topologie du réseau et les temps de vol. En effet, l'utilisation de routes directes induit des croisements potentiels plus nombreux que pour des routes standards, l'avion le plus contraint pouvant ainsi être plus difficile à séparer du reste du trafic, d'où un retard maximal potentiellement plus élevé qu'en routes standards.

Dans la globalité, cependant, le trafic est plus facile à écouler qu'en routes standards, puisque les survols de balises, qui induisent des contraintes sur de nombreux vols (celles-ci sont souvent des points de convergence de plusieurs routes aériennes), sont alors supprimés. D'autre part, le temps de vol de chaque avion est réduit grâce à l'utilisation de routes directes, réduisant ainsi le nombre d'avions simultanément en vol (entre 6 % et 7 % de réduction selon [Alliot 97b]). Les contraintes en routes directes sont donc plus nombreuses

2. Les calculs sont réalisés sur un processeur Intel Xeon cadencé à 2,66 GHz.




(a) Retard maximum (en minutes).

(b) Pourcentage de vols retardés.

(c) Somme des retards (en minutes).

(d) Retard moyen (en minutes) par vol retardé.

FIGURE 8.9 – Comparaison des résultats de l'allocation de créneaux de décollage en routes standards (S) et en routes directes (D). En abscisse, la taille de l'instance.

mais moins « dures » qu'en routes standards. La somme des retards ainsi que le pourcentage de vols retardés re étant la performance globale de la résolution, il paraît donc normal de voir ces valeurs diminuer lors de l'utilisation de routes directes. Le retard moyen par vol retardé (figure 8.9(d)) est peu in uencé par l'utilisation de routes directes.

L'utilisation de routes directes est donc intéressante à plusieurs titres : les compagnies aériennes gagnent en temps de vol et donc en consommation de carburant, la quantité de gaz polluants rejetés dans l'atmosphère suit la même diminution et les mesures de régulation peuvent être moins coûteuses qu'en routes standards. Il faut néanmoins garder à l'esprit qu'un trafic suivant un réseau de routes directes rend le contrôle par un opérateur humain plus difficile et nécessite donc de fournir au contrôleur des outils d'aide à la détection et à la résolution des con its.

### 8.2.3 Modi cation de la norme de séparation

La précision des systèmes de gestion de vol ainsi que des systèmes de surveillance augmente sans cesse, de sorte qu'il est possible d'envisager de réduire les normes de séparation tout en conservant le niveau de sécurité actuel. De telles mesures avaient été prises au début des années 2000 pour réduire la norme de séparation verticale dans l'espace aérien supérieur (au-dessus du FL 290), passant de 2 000 ft à 1 000 ft. Augmenter la norme dans le modèle, au contraire, peut permettre de prendre en compte des incertitudes sur le suivi de trajectoire en utilisant une « zone tampon » (entre la norme de séparation du modèle et la norme de séparation réelle), de manière similaire au principe présenté dans la section 4.6.1.

La figure 8.10 montre l'évolution des coûts associés à l'allocation de créneaux de décollage en fonction de la valeur de la norme de séparation horizontale pour la journée du 10/10/2008. Même si les valeurs 1 NM et 2 NM ne sont pas réalistes, il est intéressant d'observer qu'une réduction de 1 NM de la norme permet de réduire la somme des retards d'environ 30 %, avec 4 % de vols retardés en moins et qu'une norme de séparation de 3 NM permet de diviser par deux la somme des retards et de réduire de près de 25 % le nombre de vols retardés. Le retard maximal est peu in uencé par cette modification, sauf pour les valeurs extrêmes.

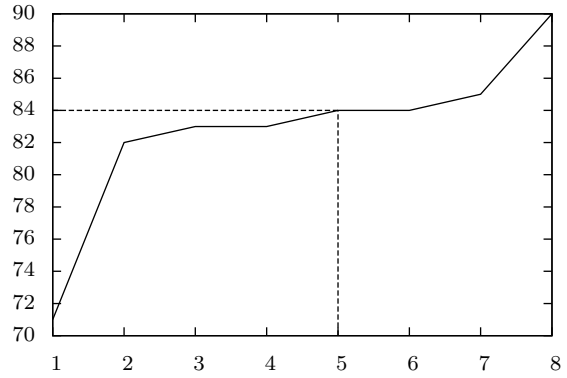
### 8.2.4 In uence de l'allocation de niveaux de vol sur l'allocation de créneaux

Afin d'améliorer les résultats de l'allocation de créneaux de décollage, nous avons combiné celle-ci avec une affectation préalable de niveaux de vol. Le principe est de séparer une partie des avions en altitude, afin d'obtenir une instance moins contrainte pour l'allocation de créneaux, menant ainsi à des solutions moins coûteuses en termes de retards. L'objet de cette section est d'évaluer les gains apportés par cette combinaison de méthodes de planification.

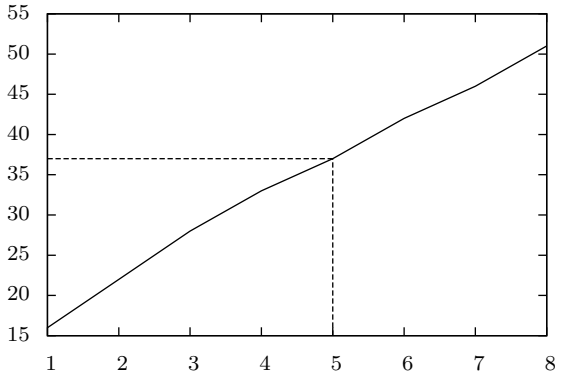
Les trajectoires en entrée de l'allocation de créneaux de décollage sont donc celles modifiées par l'allocation de niveaux de vol, avec une différence maximale de plus ou moins trois niveaux utilisables, *i.e.* $\Delta_{\max} = 30$, valeur qui permet la satisfaction de toutes les contraintes du CSP représentant l'allocation de niveaux.

La figure 8.11 compare les résultats de l'allocation de créneaux de décollage lorsque celle-ci est effectuée seule ou avec une allocation préalable de niveaux de vol. Le retard maximum alloué (figure 8.11(a)) est in uencé de manière très inégale selon les instances considérées (jusqu'à 50 % pour certaines instances, pas d'amélioration pour d'autres). Comme évoqué dans les sections précédentes, cela indique seulement que le con it le plus dur a été résolu ou simplifié par l'allocation de niveaux de vol dans certaines instances. Il est par ailleurs possible, pour les mêmes raisons, que cette valeur augmente, même si ce phénomène n'a pas été observé sur les échantillons de données à notre disposition.

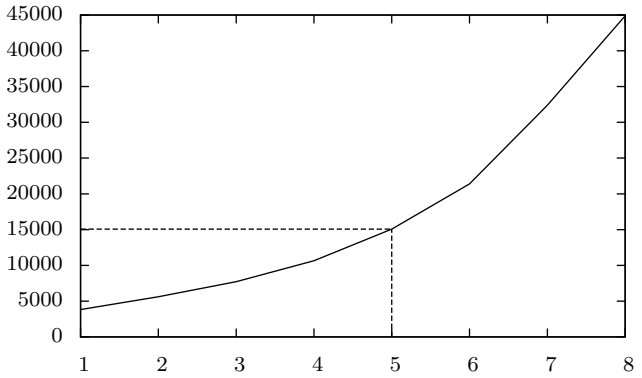
En revanche, le nombre de vols retardés (figure 8.11(b)) et la quantité totale de retard allouée (figure 8.11(c)) diminuent significativement grâce à l'allocation de niveaux de vol. On observe en moyenne 5 % de vols retardés en moins ainsi qu'une réduction moyenne de




(a) Retard maximal (min) en fonction de la norme de séparation (NM).

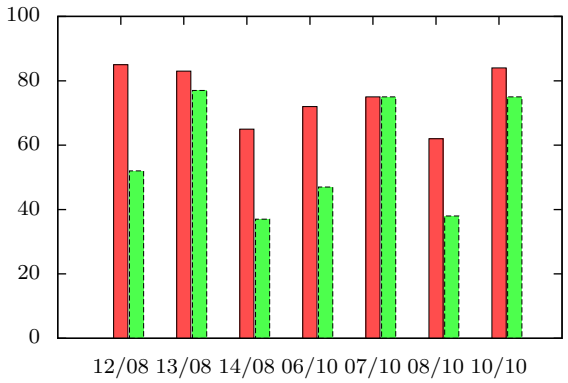



(b) Pourcentage de vols retardés en fonction de la norme de séparation (NM).

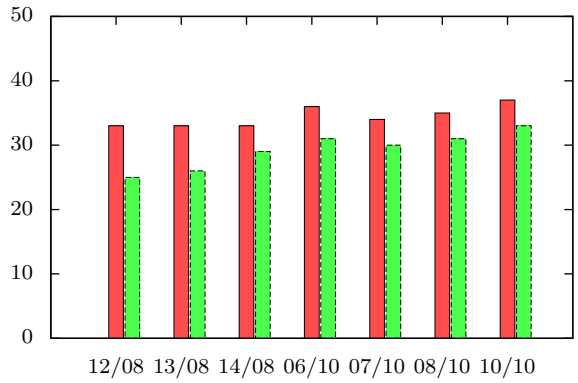



(c) Somme des retards (min) en fonction de la norme de séparation (NM).

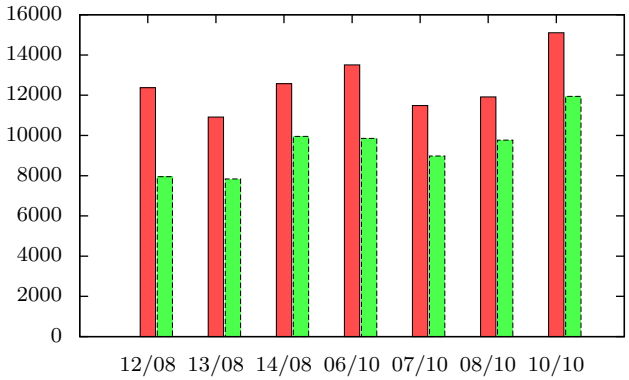
FIGURE 8.10 – In uence de la valeur de la norme de séparation horizontale sur le coût de l'allocation de créneaux de décollage. En pointillés, la valeur actuelle de la norme.




(a) Retard maximum (en minutes).




(b) Pourcentage de vols retardés.



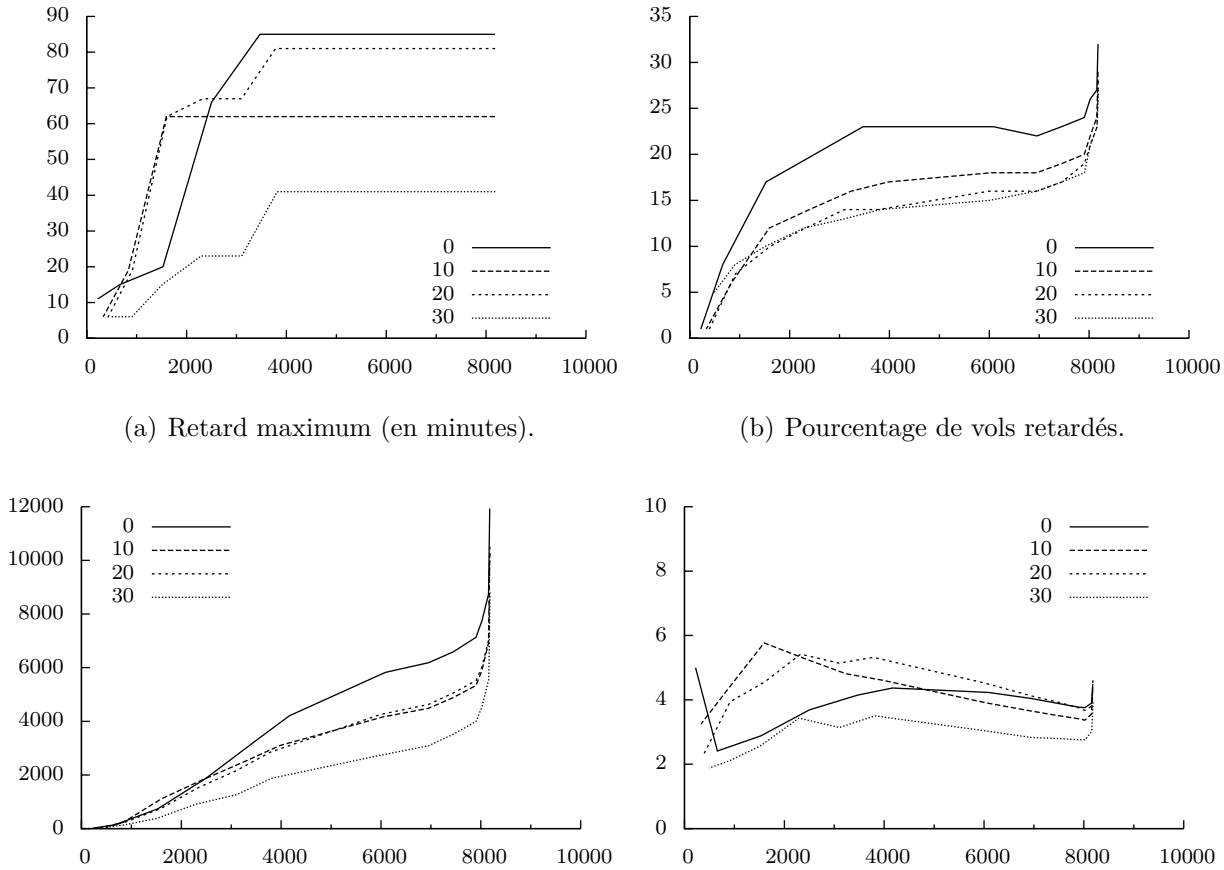

(c) Somme des retards (en minutes).

FIGURE 8.11 – Comparaison des résultats de l'allocation de créneaux de décollage avec (en vert) et sans (en rouge) allocation préalable de niveaux de vol. Les histogrammes verts ne prennent pas en compte l'éventuelle augmentation de temps de vol induite par l'affectation de niveaux de vol.

25 % de la somme des retards (ces valeurs pouvant atteindre respectivement 8 % et 36 % sur les instances les plus favorables). La phase d'allocation de niveaux de vol permet donc de rendre le trafic plus facile à réguler lorsque l'ensemble de ces contraintes est satisfait (comme c'est le cas avec $\Delta_{\max} = 30$ pour les instances traitées).

Toutefois, il peut être intéressant du point de vue du coût global de la régulation de réaliser l'allocation de niveaux de vol avec une valeur plus faible de $\Delta_{\max}$, afin de diminuer la consommation de carburant (voir section 5.1). La figure 8.12 montre l'in uence de la valeur de $\Delta_{\max}$ sur les résultats de l'allocation de créneaux de décollage.

Les valeurs de $\Delta_{\max}$ sont limitées à l'ensemble $\{0, 10, 20, 30\}$ : en effet, toutes les contraintes du modèle de l'allocation de niveaux de vol sont satisfaites pour $\Delta_{\max} = 30$ et il est donc inutile de tester des valeurs supérieures, qui n'apporteront rien du point de vue




(a) Retard maximum (en minutes).

(b) Pourcentage de vols retardés.

(c) Somme des retards (en minutes).

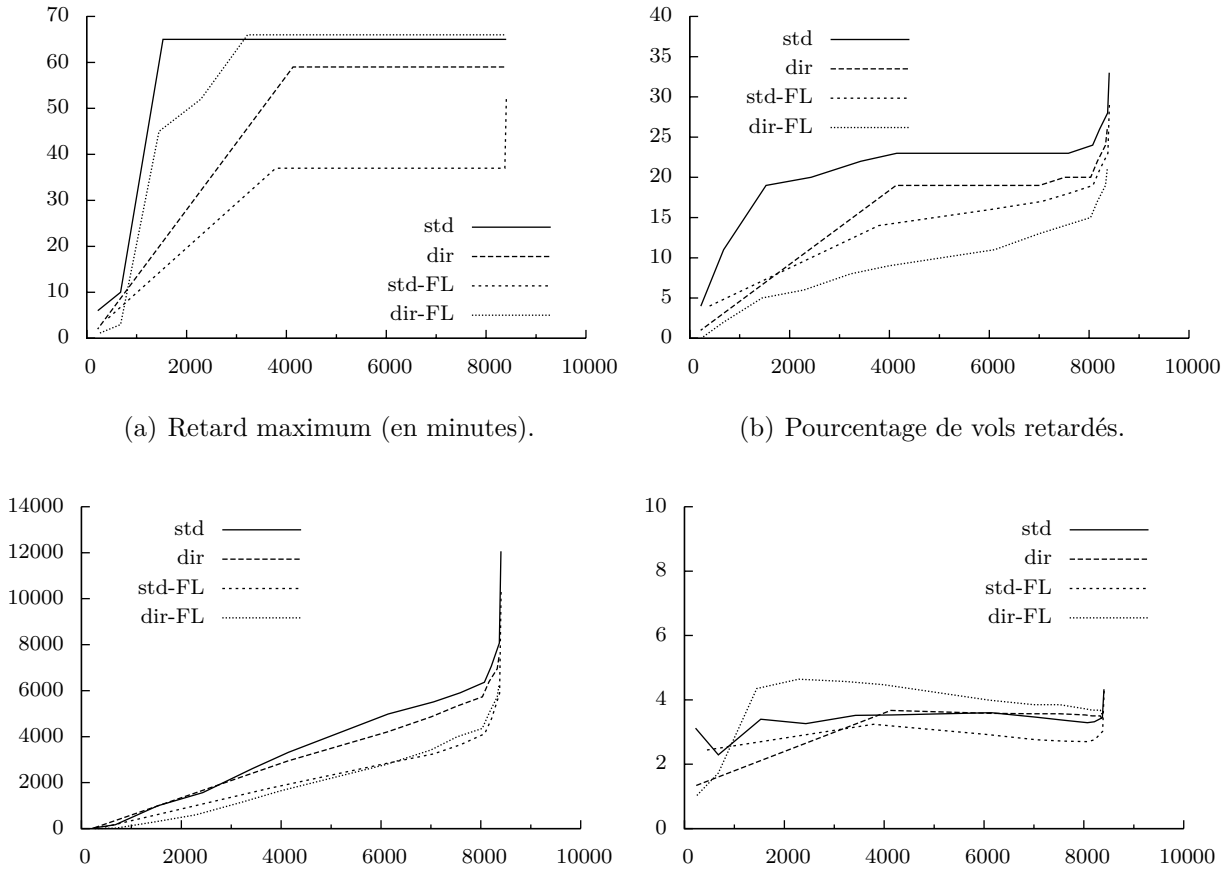
(d) Retard moyen (en minutes) par vol retardé.

FIGURE 8.12 – Comparaison des résultats de l'allocation de créneaux de décollage en fonction de la valeur de $\Delta_{\max}$ pour la journée du 12/08/2008 dans l'espace aérien français.

de la résolution de con its mais risquent de dégrader le coût de la solution en termes de consommation de carburant. On retrouve par ailleurs, pour cette valeur, les réductions de la somme des retards et du pourcentage de vols retardés déjà observées sur la figure 8.11.

Encore une fois, le critère d'optimisation (figure 8.12(a)) permet d'obtenir une preuve d'optimalité mais ne re ète pas la difficulté globale de la résolution, même si pour cette instance, l'allocation préalable de niveaux de vol permet de réduire sa valeur de manière significative.

Le nombre de vols retardés (figure 8.12(b)) est réduit par l'allocation de niveaux de vol et cette réduction est significative même pour une faible valeur de $\Delta_{\max}$. Il en est de même pour la somme des retards (figure 8.12(c)), qui est réduite d'environ 50 % pour $\Delta_{\max} = 30$ et 25 % pour $\Delta_{\max} = 10$ et $\Delta_{\max} = 20$. Il est d'ailleurs intéressant de remarquer que les réductions de ce coût obtenues avec ces deux valeurs sont très proches : si on cherche un compromis entre le coût de l'allocation de niveaux de vol et celui de l'allocation de créneaux




(a) Retard maximum (en minutes).

(b) Pourcentage de vols retardés.

(c) Somme des retards (en minutes).

(d) Retard moyen (en minutes) par vol retardé.

FIGURE 8.13 – Comparaison des résultats de l'allocation de créneaux de décollage avec et sans allocation de niveaux de vol, en routes standards et en routes directes pour la journée du 14/08/2008.

de décollage, on choisira celle de ces valeurs qui induira la plus petite valeur de $\text{cost}_{FL}$. Enfin, le retard moyen par vol retardé (figure 8.12(d)) n'est pas in uencé par l'allocation de niveaux de vol.

La figure 8.13 montre les résultats de l'allocation de créneaux de décollage pour la journée du 14/08/2008, en comparant les résultats obtenus en routes standards et en routes directes, avec ou sans allocation préalable de niveaux de vol. On observe les tendances déjà décrites précédemment : l'allocation préalable de niveaux de vol permet un gain en termes de retards alloués et l'utilisation de routes directes donne des solutions moins coûteuses que pour les routes standards.

Nous avons également testé nos algorithmes sur une journée de trafic à l'échelle européenne de 2002 (22 000 vols). Cette instance, contenant environ 16 000 variables après filtrage, a été entièrement résolue, avec 31 % de vols retardés, environ 12 minutes de retard

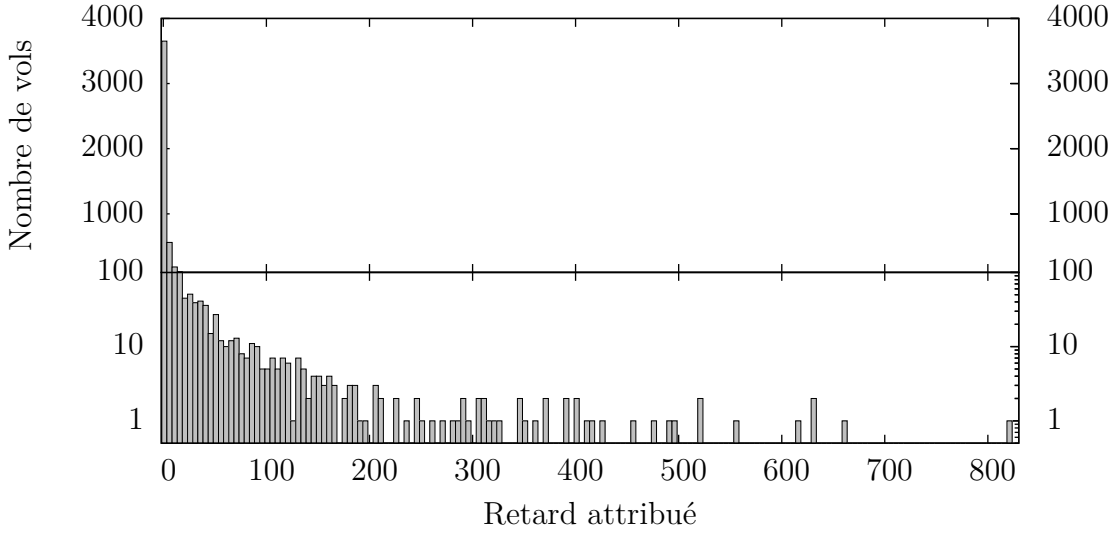



FIGURE 8.14 – Distribution des retards pour une journée de trafic à l'échelle européenne. L'échelle est logarithmique entre 0 et 100 pour faciliter la lecture. Seuls les vols retardés apparraissent (environ 31 %).

par vol retardé en moyenne et un retard maximal (prouvé optimal) de 829 minutes. Cette dernière valeur est très élevée (environ 14 heures) et nécessitera, dans un cadre opérationnel, de traiter ce vol avec d'autres méthodes. Il en sera de même pour tous les vols dont le retard atteint une valeur prohibitive. Cependant, on remarque en observant la distribution des retards (figure 8.14) que ces vols sont très minoritaires ; leur traitement ne devrait donc pas être particulièrement problématique. Il est également possible de modifier le modèle actuel de façon à pouvoir relaxer ces contraintes, comme nous l'avons déjà fait pour l'allocation de niveaux de vol. Cela permettrait de trouver des compromis entre le nombre de con its évités et le retard maximal alloué lors du processus.

Les résultats présentés dans cette section placent notre méthode de planification à un niveau comparable à la planification actuellement réalisée par la CFMU en termes de retards. Cependant, nous n'avons jusqu'alors pas pris en compte d'incertitudes, notamment sur les heures de décollage et sur le suivi longitudinal des trajectoires.

### 8.2.5 Prise en compte des incertitudes

Nous avons testé la robustesse des solutions trouvées en simulant à l'aide de CATS le trafic régulé et en introduisant une incertitude uniforme, prise dans l'intervalle $[-\frac{err}{2}, \frac{err}{2}]$ ($err$ étant alors un paramètre de la simulation), sur les heures de décollage des avions. La figure 8.15 montre le pourcentage de con its supplémentaires détectés par CATS en fonction de la valeur de $err$.

On observe que la plupart des con its évités grâce à la planification sont à nouveau présents dès lors qu'on introduit de l'incertitude sur les heures de décollage : même une petite incertitude de $\pm 1$ min « casse » environ 50 % des résolutions et ce pourcentage

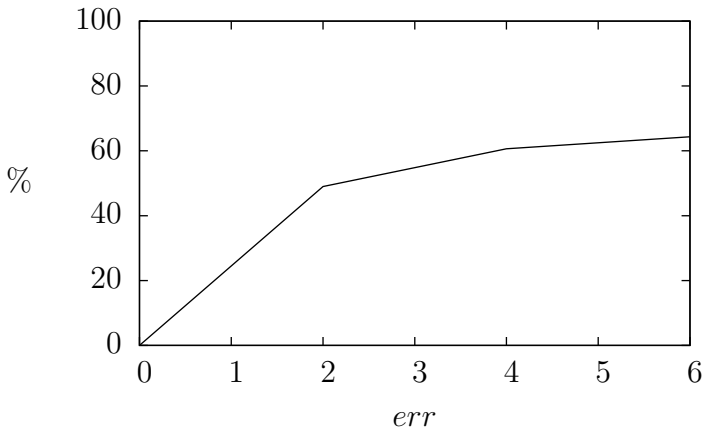



FIGURE 8.15 – Pourcentage de con its supplémentaires détectés par CATS en fonction de la valeur de l'incertitude sur les heures de décollage pour la journée du 14/08/2008.

augmente avec l'amplitude des incertitudes. En effet, la recherche d'un optimum a tendance à « saturer » les contraintes (*i.e.* avec des valeurs de $d_{ij}$ aux bornes des intervalles de con it), de sorte que la moindre incertitude peut entraîner la violation d'une partie d'entre elles. Afin d'améliorer la robustesse des solutions, nous avons mis en uvre les méthodes présentées dans la section 4.6.

### Extension des con its

La première technique de prise en compte des incertitudes, décrite dans la section 4.6.1, consiste à étendre les con its avec une amplitude $ext$ donnée (cette extension est symétrique de part et d'autre des intervalles de con it). Ainsi, une extension de $ext$ permettra d'« absorber » une incertitude $err \leq ext/2$. En revanche, pour $err > ext/2$, on se retrouve dans la situation, évoquée précédemment, où certaines résolutions ne sont plus valides à cause des incertitudes trop importantes. La figure 8.16, qui montre le pourcentage de résolutions invalides en fonction des paramètres $ext$ et $err$, illustre cette tendance.

On observe bien les effets attendus, puisqu'en deçà de la ligne $ext/2 = err$, aucun con it n'est observé pendant la simulation de validation. Au-delà de cette ligne, en revanche, la proportion de résolutions invalides augmente rapidement avec la différence $err - ext/2$.

En théorie, il est donc possible de prendre en compte les incertitudes sur les heures de décollage puisqu'il suffit de choisir $ext/2 \geq err$. Dans la pratique, cependant, l'augmentation de la valeur de ce paramètre rend les contraintes du modèle plus dures et entraîne des solutions beaucoup plus coûteuses, lorsque celles-ci existent. La figure 8.17 montre l'évolution du coût de la régulation (en termes de retard total alloué et de taux de vol retardés) en fonction de la valeur de $ext$ pour une journée de trafic.

On observe que la plus petite extension possible ($ext/2 = 1$) triple déjà la somme des retards alloués par rapport à l'instance où $ext = 0$ et que la prise en compte d'une incertitude de $\pm 2$ min (*i.e.* $ext/2 = 2$) la multiplie par près de 60, ce qui entraîne un retard

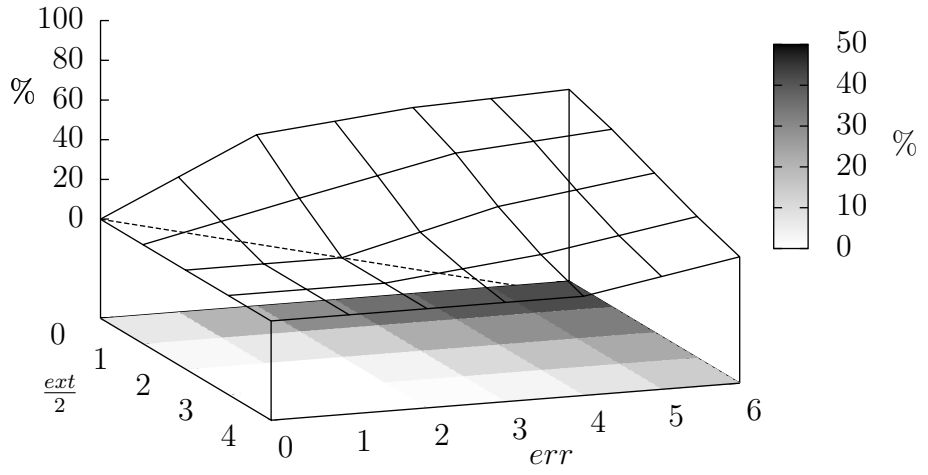



FIGURE 8.16 – Pourcentage de résolutions invalides en fonction de l'extension des con its et de l'incertitude sur les heures de décollage.

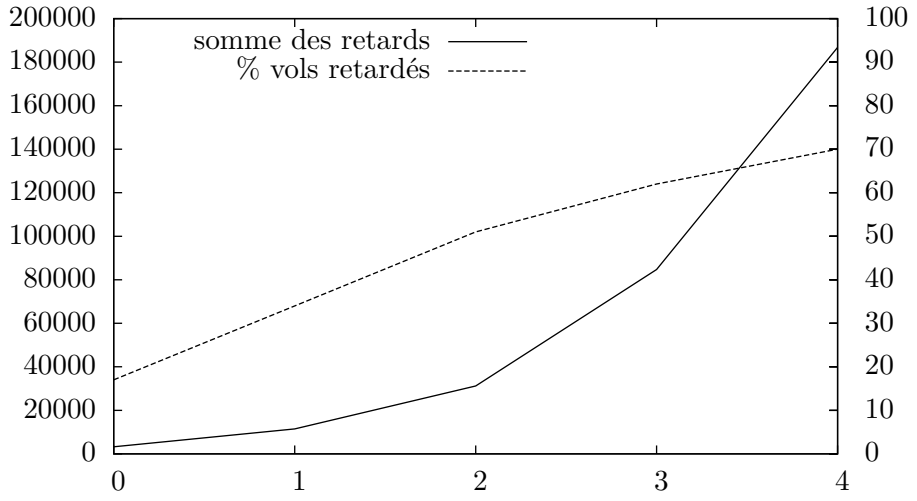



FIGURE 8.17 – In uence de la valeur de l'extension $ext/2$ des con its sur le coût de la régulation pour la journée du 14/08/2008 : somme des retards (échelle de gauche) et pourcentage de vols retardés (échelle de droite).

moyen de presque 40 min pour environ 70 % des vols.

Cette méthode de prise en compte statique des incertitudes n'est donc pas envisageable dans un cadre réaliste : si on souhaite anticiper le trafic sur la totalité de la journée, il est nécessaire de pouvoir prendre en compte une incertitude sur les heures de décollage de quelques minutes au moins, ce qui n'est clairement pas possible avec ce modèle. La section suivante présente quelques résultats préliminaires du modèle dynamique de prise en compte des incertitudes qui permet de remédier en partie à ce problème.

### Prise en compte dynamique avec une fenêtre glissante

La technique présentée dans la section 4.6.2 propose de traiter le problème avec un horizon glissant paramétré par sa taille $H_r$ et son pas de rafraîchissement $\gamma$. Les valeurs choisies pour ces paramètres (voir section 4.6.2) sont $H_r = 2\,\mathrm{h}$ et $\gamma = 5\,\mathrm{min}$. L'intérêt d'avoir un pas de rafraîchissement faible est que les incertitudes à prendre en compte ont une amplitude réduite : si la situation est reconsidérée toutes les cinq minutes, une amplitude de une à deux minutes pour les incertitudes sur les heures de décollage paraît être une hypothèse raisonnable.

L'utilisation de cette technique a permis de valider la prise en compte dynamique de petites incertitudes : malgré les vols « bruités » (section 4.6.2), chaque « tranche » a pu être résolue (exception faite des con its entre les vols que l'algorithme ne pouvait plus modifier). D'autre part, le découpage du problème a permis d'obtenir des solutions moins coûteuses vis-à-vis de la somme des retards (2 à 4 % de réduction), ce qui s'explique par le fait que :

- l'heuristique d'instanciation est plus efficace sur des problèmes de taille réduite ;
- l'optimisation est faite sur le retard maximal, qui est plus faible sur la plupart des « tranches » de deux heures que sur une journée.

En revanche, l'introduction d'incertitudes dans cette résolution en fenêtre glissante peut faire apparaître des con its sur lesquels l'algorithme n'a plus aucune marge de man uvre et qu'il sera donc nécessaire de prendre en compte par d'autres méthodes de contrôle en phase tactique.

# Conclusion et perspectives

Nous avons proposé dans cette thèse une nouvelle méthode de régulation du trafic aérien en Europe. Plutôt que de réguler le trafic sur les bases de mesures macroscopiques de capacité des secteurs de contrôle, notre méthode consiste à détecter et éviter les con its potentiels. Elle se décompose en deux étapes :

1. une allocation de niveaux de vol qui permet de séparer les vols en altitude pendant la phase de croisière et qui minimise l'écart entre le niveau de vol optimal et le niveau de vol attribué ;
2. une allocation de créneaux de décollage qui permet d'éviter les con its non traités dans la première étape et qui minimise les retards.

Deux types de méthodes ont été testés pour la résolution de ces problèmes d'optimisation combinatoires : algorithmes évolutionnaires et programmation par contrainte, et les solutions trouvées ont été validées à l'aide du simulateur de trafic CATS.

## Plani cation du tra c aérien

Nos résultats montre la faisabilité de ces principes de régulation en l'absence d'incertitudes sur le suivi des trajectoires 4D. Pour toutes les instances traitées, il a été possible d'éviter la totalité des con its. Les niveaux de vol attribués diffèrent peu des niveaux de vol optimaux, donc la consommation de carburant pour la phase de croisière est très peu impactée. D'autre part, les retards attribués sont comparables aux retards actuellement alloués par l'algorithme de la CFMU. Pour la plupart des instances traitées, l'optimalité du retard maximal alloué a été prouvée.

**Robustesse** Nous avons par ailleurs proposé deux méthodes permettant de considérer des incertitudes sur les heures de décollage : l'une statique qui consiste à étendre les con its, l'autre dynamique qui utilise une fenêtre glissante afin de prendre en compte de manière réactive les mises à jour des heures réelles de décollage dues aux incertitudes. Cependant, nous avons mis en évidence que la prise en compte statique de fortes incertitudes induit des coûts très élevés en termes de retards, de sorte que les méthodes mises en uvre ne sont adaptées que pour des faibles amplitudes d'incertitude.

La prise en compte d'autres sources d'incertitudes, notamment sur le suivi temporel des trajectoires, est importante pour l'amélioration du réalisme opérationnel de notre modèle. Les méthodes présentées dans cette thèse peuvent être adaptées à ces fins, avec

les inconvénients cités précédemment. Ces remarques soulignent l'importance de pouvoir prévoir les trajectoires depuis le sol avec une précision élevée. Cependant, la prévision de trajectoire entraîne des négociations entre les différents acteurs du trafic aérien afin de déterminer les informations importantes et celles qui sont susceptibles d'être échangées [1] [Huchet 06].

**Résolution de con it** Les méthodes présentées dans cette étude permettent d'éviter les con its dès la planification des vols. Cependant, les vols dont l'aéroport d'origine est situé en dehors de la zone CEAC ne sont pas pris en compte dans cette planification. La résolution des con its entre ces vols devra être effectuée à l'aide de procédures standards de contrôle ou de méthodes automatiques, comme celle disponible dans le simulateur CATS.

D'autre part, il est certain que, dans un cadre réaliste, on se trouve confronté à des incertitudes sur les heures de décollage ou le suivi de trajectoire 4D ayant une amplitude trop élevée pour pouvoir être totalement prises en compte par les méthodes évoquées dans notre étude. Il en résulte que certains vols que l'on pensait avoir séparés vont finalement être en situation de con it. Ces situations devront être évitées, là encore, par d'autres types de procédures. Il sera donc intéressant, dans de futurs travaux, d'étudier l'in uence de notre planification sur les résolutions dynamiques apportées par ces méthodes. Le trafic régulé est-il plus « facile » à écouler que le trafic non régulé ? La régulation permet-elle de réduire le coût des man uvres nécessaires ?

**Contraintes opérationnelles** Afin de rendre le modèle plus réaliste, il est important de prendre en compte certaines contraintes opérationnelles supplémentaires, notamment en ce qui concerne la gestion des départs et des arrivées. En effet, dans notre étude, nous ne prenons pas en compte la capacité limitée des aéroports en termes de nombre de mouvements [2]. Il est par ailleurs tout à fait possible que l'algorithme de planification attribue une heure de décollage identique à deux avions décollant du même aéroport, ce qui n'est pas réalisable dans la réalité [3]. Ces contraintes peuvent être ajoutées au modèle sans difficulté, mais il est nécessaire de déterminer ces mesures de capacité pour tous les aéroports concernés.

Les vols en correspondance et les temps de rotation des appareils constituent également une contrainte opérationnelle forte dans le cadre de la planification par allocation de créneaux de décollage, dans la mesure où le retard d'un vol peut entraîner le retard d'autres vols, en correspondance ou utilisant le même appareil. Ce type de contrainte peut être implémenté facilement au sein de notre modèle, mais les données nécessaires à leur mise en uvre sont cependant difficiles à obtenir. S'il est possible de déduire facilement une rotation d'avion sur un petit aéroport (en identifiant la compagnie aérienne et les heures d'arrivée et

1. Certaines informations sont en e et considérées comme stratégiques par les compagnies aériennes, qui ne souhaitent donc pas les communiquer.

2. Un mouvement est soit un décollage, soit un atterrissage.

3. Au décollage, deux avions successifs sont séparés par 1 à 3 min selon leur catégorie de masse respective à cause des turbulences de sillage (*e.g.* un avion léger devra attendre 3 min pour décoller après un avion lourd ). Seuls certains grands aéroports peuvent proposer des décollages simultanés sur plusieurs pistes.

de décollage), il est en revanche très difficile de déceler la politique de gestion des retards des différentes compagnies aériennes sur les plus gros aéroports, et encore plus particulièrement sur les *hubs* des compagnies. En effet, sur leurs *hubs*, les compagnies sont susceptibles de réaliser des échanges d'appareils entre leurs différents vols lorsque l'un d'entre eux est fortement retardé, mais ces données stratégiques ne sont jamais communiquées, pas même aux organismes de contrôle.

## Algorithmes

Cette étude a permis de montrer que les algorithmes évolutionnaires et la programmation par contraintes peuvent être utilisés pour la résolution de problèmes d'optimisation combinatoires de très grande taille (presque 10 000 variables). Grâce à la PPC, une preuve d'optimalité a été obtenue pour la quasi totalité des instances testées sur l'espace aérien français pour le critère retenu (minimisation du retard maximal alloué). D'autre part, l'utilisation d'une heuristique d'instanciation des valeurs, simple mais adaptée, a permis de maintenir le « coût global » (somme des retards) des solutions à une valeur faible.

Toutefois, les mesures de régulation présentées dans cette étude ne peuvent fonctionner correctement que si elles sont appliquées à l'ensemble du trafic européen. Il reste donc à savoir si nos méthodes de résolution, et notamment la PPC qui donne de meilleurs résultats sur les instances testées, permettront de résoudre des problèmes aussi grands (30 000 variables pour une journée de trafic en Europe actuellement, 50 000 variables à l'horizon 2020). Les premiers tests que nous avons faits sur une journée de trafic datant de 2002 (22 000 vols) semblent montrer la faisabilité de la méthode à l'échelle européenne, mais ces résultats doivent être confirmés sur d'autres échantillons de données afin de pouvoir conclure sur ce point.

Dans le cadre de l'allocation de créneaux de décollage, il serait intéressant d'adapter les algorithmes classiques de coloration de graphes (*tabucol* [Hertz 87, Dorne 98] ou algorithme mémétique [Lü 10] par exemple) à notre problème afin d'améliorer l'efficacité de la méthode. La principale modification à apporter tient du fait que ce type d'algorithme cherche à minimiser le nombre de couleurs (*i.e.* de niveaux de vol) alors que dans notre problème, il s'agit de minimiser pour chaque vol l'écart de niveau par rapport au niveau optimal. Des premières tentatives d'adaptation de l'algorithme *tabucol* donnent des résultats équivalents à ceux présentés dans notre étude (mais sans fournir de garantie d'optimalité).

Il pourrait également être intéressant de combiner la PPC avec d'autres paradigmes de recherche, métaheuristiques ou recherche locale par exemple, pour l'allocation de créneaux de décollage. Une telle hybridation pourrait permettre d'accélérer la recherche de solution afin de rendre le système plus réactif ou d'améliorer le coût des solutions en termes de somme des retards.

# Glossaire

**2D** deux dimensions.
**4D** quatre dimensions.

**AERA** Automated En-Route Air Traffic Control.
**ATFCM** Air Traffic Flow and Capacity Management.
**ATM** Air Traffic Management.

**BADA** Base of Aircraft DAta.

**CASA** Computer Assisted Slot Allocation.
**CATS** Complete Air Traffic Simulator.
**CEAC** Conférence Européenne de l'Aviation Civile.
**CENA** Centre d'Études de la Navigation Aérienne.
**CFMU** Central Flow Management Unit.
**CSP** Problème de Satisfaction de Contraintes.

**ÉNAC** École Nationale de l'Aviation Civile.
**ETA** Estimated Time of Arrival.
**ETFMS** Enhanced Tactical Flow Management System.
**EUROCAE** European Organisation for Civil Aviation Equipment.

**FAA** Federal Aviation Administration.
**FL** Flight Level.
**FMS** Flight Management System.
**FRAM** Free Route Airspace Maastricht.
**FREER** Free-Route Experimental Encounter Resolution.

**IFR** Instrument Flight Rules.

**JSP** Jobshop Scheduling Problem.

**LAC** Liste d'Allocation de Créneaux.

**NM** Nautical Mile.

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

**PPC** Programmation par Contraintes.

**RFL** Requested Flight Level.

**RNAV** Area Navigation.

**SESAR** Single European Sky ATM Research.

**STATFOR** Statistics and Forecasts.

**VFR** Visual Flight Rules.

# Bibliographie


[Abad 04] Antonio Manuel Abad. Using Tactical Flight Level Resource Allocation to Alleviate Congested En-Route Airspace. Master of science in aeronautics and astronautics, Massachusetts Institute of Technology, Boston, Massachusetts, juin 2004. *Cité p.* 37

[Airbus 04] Airbus. *Getting to Grips With Fuel Economy.* Rapport technique 3, Flight Operations Support and Line Assistance, Toulouse, juillet 2004. *Cité p.* 70

[Allignol 09] Cyril Allignol. *4D Trajectory Management and Complexity Reduction.* Simulation Report D 4.3.2-02, Episode 3 Consortium, septembre 2009. *Cité p.* 3, 96

[Allignol 10] Cyril Allignol et Nicolas Barnier. *Allocation de créneaux de décollage sans con it en programmation par contraintes.* ROADEF 2010, Toulouse, février 2010. *Cité p.* 3

[Allignol 11] Cyril Allignol, Nicolas Barnier et Nicolas Durand. *Allocation de créneaux de décollage sans con it.* ROADEF 2011, Saint-Étienne, mars 2011. *Cité p.* 3

[Alliot 97a] Jean-Marc Alliot, Jean-François Bosc, Nicolas Durand et Lionnel Maugis. *CATS : A Complete Air Tra c Simulator.* 16th DASC, 1997. *Cité p.* 42

[Alliot 97b] Jean-Marc Alliot, Jean-François Bosc, Nicolas Durand et Lionnel Maugis. *An Experimental Study of ATM Capacity.* International Air Traffic Management R&D Seminar ATM-1997, Saclay, juin 1997. *Cité p.* 124

[Alliot 03] Jean-Marc Alliot et Dominique Colin de Verdière. *ATM : 20 ans d e ort et perspectives.* Symposium de l'Académie Nationale de l'Air et de l'Espace : vers l'automatisation du vol et sa gestion, 2003. *Cité p.* 28

[Angerand 92] Luc Angerand et Hervé Le Jeannic. *Bilan du projet SAINTEX.* Rapport technique, CENA, mars 1992. *Cité p.* 29

[Archambault 04a] Nicolas Archambault. *Speed Uncertainty and Speed Regulation in Con ict Detection and Resolution in Air Tra c Control.* ICRAT'2004, 2004. *Cité p.* 33

[Archambault 04b] Nicolas Archambault, Géraud Granger et Nicolas Durand. *Heuristiques d ordonnancement pour une résolution embarquée de con its aériens par une méthode séquentielle.* RIVF'04, 2004. *Cité p.* 32

[Averty 98] Philippe Averty. *Les e ets de la charge de tra c sur le niveau d activation psychophysiologique du contrôleur aérien.* Thèse de doctorat, Institut de psychologie, Université Lyon 2, 1998. *Cité p.* 34

[Bagley 67] John D. Bagley. *The Behavior of Adaptative Systems Which Employ Genetic and Correlation Algorithms.* Doctoral dissertation, University of Michigan, Ann Arbor, décembre 1967. *Cité p.* 89

[Baker 85] James E. Baker. *Adaptative Selection Methods for Genetic Algorithms.* Proceedings of the 1st International Conference on Genetic Algorithms, pages 101–111, 1985. *Cité p.* 85

[Baptiste 01] Philippe Baptiste, Claude Le Pape et Wim Nuijten. Constraint-based scheduling, applying constraint programming to scheduling problems, volume 39 de *Kluwer s International Series in Operations Research & Management Science.* Springer, 2001. *Cité p.* 110

[Barnier 01a] Nicolas Barnier et Pascal Brisset. *FaCiLe : a Functional Constraint Library.* Colloquium on Implementation of Constraint and LOgic Programming Systems CICLOPS'01 (Workshop of CP'01), Paphos, Cyprus, décembre 2001. *Cité p.* 99

[Barnier 01b] Nicolas Barnier, Pascal Brisset et Thomas Rivière. *Slot Allocation with Constraint Programming : Models and Results.* International Air Traffic Management R&D Seminar ATM-2001, Santa Fe (NM), USA, décembre 2001. *Cité p.* 36

[Barnier 02a] Nicolas Barnier. *Application de la programmation par contraintes à des problèmes de gestion du tra c aérien.* Thèse de doctorat, Institut National Polytechnique de Toulouse, décembre 2002. *Cité p.* 97

[Barnier 02b] Nicolas Barnier et Pascal Brisset. *Graph Coloring for Air Tra c Flow Management.* CPAIOR'02 : Fourth International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems, pages 133–147, Le Croisic, France, mars 2002. *Cité p.* 36, 72

[Barnier 09] Nicolas Barnier et Cyril Allignol. *4D-Trajectory Decon iction Through Departure Time Adjustment.* International Air Traffic Management R&D Seminar ATM-2009, Napa (CA), USA, juillet 2009. *Cité p.* 3

[Barnier 11] Nicolas Barnier et Cyril Allignol. *Combining Flight Level Allocation with Ground Holding to Optimize 4D-Decon iction.* International Air Traffic Management R&D Seminar ATM-2011, Berlin, juin 2011. *Cité p.* 3

[Barnier 12] Nicolas Barnier et Cyril Allignol. *Trajectory Decon iction with Constraint Programming.* The Knowledge Engineering Review, 2012. À paraître. *Cité p.* 3

[Becker 10] Caroline Becker. Allocation de niveaux de vol pour la décon iction de trajectoires 4d. Master Recherche Informatique et Télécommunications, parcours Intelligence Artificielle, Université de Toulouse, septembre 2010. *Cité p.* 54

[Bessière 93] Christian Bessière et Marie-Odile Cordier. *Arc-Consistency and Arc-Consistency Again.* Proceedings of the 11th National Conference on Artificial Intelligence, AAAI'93, pages 108–113, Washington, D.C., 1993. AAAI Press. *Cité p.* 105

[Bessière 96] Christian Bessière et Jean-Charles Régin. *MAC and Combined Heuristics : Two Reasons to Forsake FC (and CBJ ?) on Hard Problems.* Proceedings CP'96, pages 61–75, Cambridge MA, 1996. *Cité p.* 106

[Bontemps 04] Christian Bontemps et Kévin Guittet. *Commentaires sur l étude de l Université de Westminster portant sur les coûts des délais ATC.* Note technique, Centre d'Études de la Navigation Aérienne, Toulouse, juillet 2004. *Cité p.* 18

[Brindle 81] Anne Brindle. *Genetic Algorithms for Function Optimization.* Rapport technique TR 81-2, University of Alberta, Edmonton, Canada, 1981. *Cité p.* 85

[Carlier 03] Jacques Carlier, Dritan Nace, Vu Duong et Huy-Hoang Nguyen. *Using Disjunctive Scheduling for a New Sequencing Method in Multiple-Con icts Solving.* Intelligent Transportation Systems Conference (ITSC'03), Shanghai, octobre 2003. *Cité p.* 32

[Cerf 94] Raphaël Cerf. *Une théorie asymptotique des algorithmes génétiques.* Thèse de doctorat, Université de Montpellier II, mars 1994. *Cité p.* 82

[CFMU 11] Eurocontrol CFMU, Brussels. *Basic CFMU Handbook - General & CFMU Systems*, 15.0 edition, mars 2011. *Cité p.* 15

[Chiang 97] Yi-Jen Chiang, James T. Klosowski, Changkil Lee et Joseph S. B. Mitchell. *Geometric Algorithms for Con ict Detection / Resolution in Air Tra c Management.* 36th IEEE Conference on Decision and Control, pages 1835–1840, 1997. *Cité p.* 32

[CODA 09] CODA. *CODA Digest Delays to Air Transport in Europe.* Rapport technique, Eurocontrol – Central Office for Delay Analysis, 2009. *Cité p.* 17

[Constans 05a] Sophie Constans, Bastian Fontaine et Rémy Fondacci. *Minimizing Potential Con ict Quantity with Speed Control.* Fourth Eurocontrol Innovative Research Workshop and Exhibition, pages 265–274, Brétigny-sur-Orge, décembre 2005. *Cité p.* 33

[Constans 05b] Sophie Constans, Nicolas Gadenne et Rémy Fondacci. *Applying Genetic Techniques to the Tactical Flight Level Assignment.* Andrzej Jaszkiewicz, éditeur, Advanced OR and AI Methods in Transportation, pages 69–75, Poznan, Pologne, septembre 2005. *Cité p.* 37, 81

[Cook 04] Andrew J. Cook, Graham Tanner et Stephen Anderson. *Evaluating the True Cost to Airlines of One Minute of Airborne or Ground Delay : Final Report.* Rapport technique, Eurocontrol, mai 2004. *Cité p.* 17, 61

[Cook 09] Andrew J. Cook et Graham Tanner. *The challenge of managing airline delay costs.* Conference on Air Traffic Management (ATM) Economics, Belgrade, septembre 2009. *Cité p.* 17

[Dalichampt 97] Marc Dalichampt, E. Petit, Ulrich Junker et Jérôme Lebreton. *Innovative Slot Allocation (ISA).* Rapport technique, Eurocontrol, Brétigny, décembre 1997. *Cité p.* 35, 97

[de Berg 98] Mark de Berg, Marc van Kreveld, Mark Overmars et Otfrie d Schwarzkopf. Computational Geometry – Algorithms and Applications. Springer, second edition, 1998. *Cité p.* 54

[De Jong 75] Kenneth Alan De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptative Systems.* Doctoral dissertation, University of Michigan, 1975. *Cité p.* 85

[Dechter 89] Rina Dechter et Itay Meiri. *Experimental Evaluation of Preprocessing Techniques in Constraint Satisfaction Problems.* Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, pages 271–277. Morgan Kaufmann, 1989. *Cité p.* 106

[Dechter 90] Rina Dechter. *Enhancement schemes for constraint processing : Backjumping, learning, and cutset decomposition.* Artificial Intelligence, vol. 41, pages 273–312, 1990. *Cité p.* 105

[Deffeyes 06] Kenneth S. Deffeyes. Beyond Oil – The View from Hubbert's Peak. Hill and Wang, 2006. *Cité p.* 23

[Dodin 99] Pierre Dodin. Résolution de con its via la programmation semi-définie. Rapport de dea en mathématiques appliquées, Université de Paris VI, septembre 1999. *Cité p.* 33

[Dorne 98] Raphaël Dorne et Jin-Kao Hao. Meta-heuristics : Advances and Trends in Local Search Paradigms for Optimization, chapitre 6 : Tabu search for graph coloring, T-colorings and set T-colorings, pages 77–92. Kluwer, 1998. *Cité p.* 137

[dos Santos 02] José Martins dos Santos. *Eight-States Free Route Airspace Project : Free Route Airspace Concept.* Rapport technique 2.0, Eurocontrol, Bruxelles, février 2002. *Cité p.* 28

[Duong 97] Vu Duong, Eric Hoffman et Jean-Pierre Nicolaon. *Initial Results of Investigation into Autonomous Aircraft Concept (FREER-1).* International Air Traffic Management R&D Seminar ATM-1997, Saclay, juin 1997. *Cité p.* 30

[Duong 01] Vu Duong, Gilles Gawinowski, Jean-Pierre Nicolaon et Darren Smith. *Sector-Less Air Tra c Management.* International Air Traffic Management R&D Seminar ATM-2001, Sante-Fe, décembre 2001. *Cité p.* 28

[Durand 96] Nicolas Durand. *Optimisation de trajectoires pour la résolution de con its.* Thèse de doctorat, Institut National Polytechnique de Toulouse, 1996. *Cité p.* 96

[Durand 97] Nicolas Durand et Jean-Marc Alliot. *Optimal Resolution of En Route con icts.* International Air Traffic Management R&D Seminar ATM-1997, Saclay, juin 1997. *Cité p.* 42

[Durand 98] Nicolas Durand et Jean-Marc Alliot. *Genetic Crossover Operator for Partially Separable Functions.* Proceedings of the 3rd annual Genetic Programming Conference, 1998. *Cité p.* 87

[Durand 99] Nicolas Durand et Jean-Marc Alliot. *Peut-on supprimer le contrôle au sol ?* La Recherche, no. 319, pages 57–61, avril 1999. *Cité p.* 28

[Durand 10] Nicolas Durand, Cyril Allignol et Nicolas Barnier. *A ground holding model for aircraft decon iction.* 29th DASC, Salt Lake City, UT, octobre 2010. *Cité p.* 3

[Eiben 99] Ágoston E. Eiben, Robert Hinterding et Zbigniew Michalewicz. *Parameter Control in Evolutionary Algorithms.* IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, pages 124–141, juillet 1999. *Cité p.* 95

[Eiben 03] Ágoston E. Eiben et James E. Smith. Introduction to evolutionary computing. Natural Computing. Springer, première edition, 2003. *Cité p.* 82

[Eurocontrol 08] Eurocontrol. *Challenges of Growth.* Rapport technique, Eurocontrol, Brussels, 2008. *Cité p.* 23

[Eurocontrol 11] Eurocontrol, Brussels. *As the Crow Flies Free Route Airspace Maastricht*, mars 2011. *Cité p.* 12

[EWG 07] EWG. *Crude Oil The Supply Outlook.* Rapport technique, Energy Watch Group, Ottobrunn, octobre 2007. *Cité p.* 23

[FAA 11] FAA. *NextGen Implementation Plan.* Rapport technique, Federal Aviation Administration, mars 2011. *Cité p.* 30

[Ferchaud 05] Frédéric Ferchaud. *Absorption Areas and En-Route Slot Allocation.* EEC INO Workshop, 2005. *Cité p.* 35

[Feydy 05] Thibaut Feydy, Nicolas Barnier, Pascal Brisset et Nicolas Durand. *Mixed Con ict Model for Air Tra c Control.* Interval Analysis, Constraint Propagation, Applications (IntCP 2005), 2005. *Cité p.* 33

[Flener 07] Pierre Flener, Justin Pearson, Magnus Ågren, Carlos Garcia Avello, Mete Çelitkin et Søren Dissing. *Air-Tra c Complexity Resolution In Multi-Sector Planning.* Journal of Air Transport Management, vol. 13, no. 6, pages 323–328, novembre 2007. *Cité p.* 34, 97

[Forrest 85] Stephanie Forrest. Documentation for prisoners dilemma and norms programs that use the genetic algorithm. University of Michigan, Ann Arbor, 1985. *Cité p.* 89

[Frieze 95] Alan M. Frieze et Mark Jerrum. *Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION.* 4th International Conference on Integer Programming and Combinatorial Optimization, pages 1–12, 1995. *Cité p.* 36

[Fron 93] Xavier Fron, Bernard Maudry et Jean-Claude Tumelin. *ARC 2000 : Automatic Radar Control.* Rapport technique, Eurocontrol, 1993. *Cité p.* 29

[Frost 95] Daniel Frost et Rina Dechter. *Look-Ahead Value Ordering for Constraint Satisfaction Problems.* IJCAI'95, pages 572–578, Montréal, Canada, 1995. *Cité p.* 106

[Garot 05] Jean-Marc Garot et Nicolas Durand. *Failures in the Automation of Air Tra c Control.* Colloque de l'AAA, janvier 2005. *Cité p.* 30

[Gaschnig 79] John Gary Gaschnig. *Performance measurement and analysis of certain search algorithms.* PhD thesis, Carnegie Mellon University, 1979. *Cité p.* 105

[Gianazza 04a] David Gianazza. *Optimisation des ux de tra c aérien.* Thèse de doctorat, Institut National Polytechnique de Toulouse, 2004. *Cité p.* 81

[Gianazza 04b] David Gianazza, Nicolas Durand et Nicolas Archambault. *Allocating 3D trajectories to air tra c ows using $A^\star$ and genetic algorithms.* CIMCA'04, 2004. *Cité p.* 34

[Gianazza 06] David Gianazza et Kévin Guittet. *Selection and Evaluation of Air Tra c Complexity Metrics.* 25th DASC, 2006. *Cité p.* 24

[Gianazza 10] David Gianazza. *Forecasting workload and airspace con guration with neural networks and tree search methods.* Artificial Intelligence, vol. 174, no. 7-8, mai 2010. *Cité p.* 24

[Gillies 85] Andrew M. Gillies. Machine learning procedures for generating image domain feature detectors. University of Michigan, Ann Arbor, 1985. *Cité p.* 89

[Goemans 94] Michel X. Goemans et David P. Williamson. *0.878-Approximation Algorithms for MAXCUT and MAX2SAT.* 26th Annual ACM Symposium on Theory of Computing, pages 422–431, Montréal, 1994. *Cité p.* 36

[Goldberg 87] David E. Goldberg et Jon Richardson. *Genetic Algorithms with Sharing for Multimodal Function Optimization.* Proceedings of the 2nd International Conference on Genetic Algorithms and their Applications, pages 41–49, 1987. *Cité p.* 90

[Goldberg 89] David E. Goldberg. Genetic algorithms in search, optimization and machine learning. Reading MA Addison Wesley, 1989. *Cité p.* 81, 85

[Gotteland 04] Jean-Baptiste Gotteland. *Optimisation du tra c au sol sur les grands aéroports.* Thèse de doctorat, Institut National Polytechnique de Toulouse, 2004. *Cité p.* 81

[Gourmel 07] Olivier Gourmel. Allocation de créneaux de décollage par décomposition arborescente de modèles CSP. Master Recherche Sciences de la Modélisation, de l'Information et des Systèmes, École Nationale de l'Aviation Civile, Toulouse, septembre 2007. *Cité p.* 45, 46

[Graham 06] Robert Graham et David Young. *Preparing an Initial Assessment of the SESAR Concept of Operations EP3 : Single European Sky Implementation Support through Validation .* Rapport technique, Eurocontrol Experimental Centre, France, 2006. *Cité p.* 55

[Granger 98] Géraud Granger. *Résolution de con its embarquée dans les espaces de faible densité.* DEA en Informatique Fondamentale et Parallélisme, ENSEEIHT, septembre 1998. *Cité p.* 31, 32, 42

[Granger 01] Géraud Granger, Nicolas Durand et Jean-Marc Alliot. *Optimal Resolution Of En Route Con icts.* International Air Traffic Management R&D Seminar ATM-2001, Santa Fe (NM), USA, décembre 2001. *Cité p.* 32, 33

[Granger 02] Géraud Granger. *Détection et résolution de con its aériens : modélisations et analyse.* Thèse de doctorat, École Polytechnique, 2002. *Cité p.* 45, 46, 81, 96

[Grefenstette 86] John J. Grefenstette. *Optimization of Control Parameters for Genetic Algorithms.* IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-16, no. 1, pages 122–128, janvier-février 1986. *Cité p.* 95

[Griewank 82] Andreas Griewank et Philippe Toint. *On the unconstrained optimization of partially separable functions.* Michael J.D. Powell, éditeur, Nonlinear Optimization 1981, NATO Conference Series : Systems Science, pages 301–312. Academic Press, 1982. *Cité p.* 87

[Haralick 80] Robert M. Haralick et Gordon L. Elliott. *Increasing Tree Search E ciency for Constraint Satisfaction Problems.* Artificial Intelligence, vol. 14, pages 263–313, 1980. *Cité p.* 106

[Hertz 87] Alain Hertz et Dominique de Werra. *Using tabu search techniques for graph coloring.* Computing, vol. 39, no. 4, pages 345–351, décembre 1987. *Cité p.* 137

[Holland 75] John H. Holland. Adaptation in natural and artificial systems. University of Michigan Press, 1975. *Cité p.* 81

[Huchet 06] François Huchet. *Introduction à la prévision de trajectoire.* Rapport technique, DSNA/DTI, juillet 2006. *Cité p.* 136

[IAE 09] IAE. *Medium-Term Oil Market Report.* Rapport technique, International Energy Agency, Paris, décembre 2009. *Cité p.* 23

[ITA 00] ITA. *Cost of air transport delay in Europe.* Rapport technique, Institut du Transport Aérien, Paris, novembre 2000. *Cité p.* 18

[Kirkpatrick 83] Scott Kirkpatrick, Charles Daniel Gelatt et Mario P. Vecchi. *Optimization by Simulated Annealing.* Science, vol. 220, no. 4598, pages 671–680, mai 1983. *Cité p.* 96

[Lauderdale 11] Todd A. Lauderdale, Andrew C. Cone et Aisha R. Bowe. *Relative Signi cance of Trajectory Prediction Errors on an Automated Separation Assurance Algorithm.* International Air Traffic Management R&D Seminar ATM-2011, Berlin, juin 2011. *Cité p.* 62

[Leroy 11] Xavier Leroy, Damien Doligez, Alain Frisch, Jacques Garrigue, Didier Rémy et Jérôme Vouillon. *The OCaml System : Documentation and user s manual.* INRIA, 3.12 edition, juillet 2011. *Cité p.* 42

[Letrouit 98] Vincent Letrouit. *Optimisation du réseau de routes aériennes en Europe.* Thèse de doctorat, Institut National Polytechnique de Grenoble, 1998. *Cité p.* 36, 72

[Lü 10] Zhipeng Lü et Jin-Kao Hao. *A Memetic Algorithm for Graph Coloring.* European Journal of Operational Research, vol. 203, no. 1, pages 241–250, 2010. *Cité p.* 137

[Mackworth 77] Alan K. Mackworth. *Consistency in Networks of Relations.* Artificial Intelligence, vol. 8, pages 99–118, 1977. *Cité p.* 103, 104, 105

[Meckiff 94] Colin Meckiff et Philip Gibbs. *PHARE Highly Interactive Problem Solver.* Rapport technique, Eurocontrol, novembre 1994. *Cité p.* 29

[Michalewicz 92] Zbigniew Michalewicz. Genetic algorithms + data structures = evolution programs. Springer-Verlag, 1992. *Cité p.* 83

[Nace 03] Dritan Nace, Jacques Carlier, Nhat Linh Doan et Vu Duong. *A Linear Programming Approach for Route and Level Flight Assignment.* International Air Traffic Management R&D Seminar ATM-2003, Budapest, Hongrie, juin 2003. *Cité p.* 34

[Niedringhaus 83] W. P. Niedringhaus, I. Frolow, J. C. Corbin, A. H. Gisch et N. J. Taber. *Automated En Route Air Tra c Control Algorithmic Speci cations.*

Rapport technique 3, Federal Aviation Administration, Washington, D.C., septembre 1983. *Cité p.* 28

[Nuic 11] Angela Nuic. *User Manual for the Base of Aircraft Data (BADA) Revision 3.9.* Eurocontrol Experimental Centre, Brétigny, avril 2011. *Cité p.* 43

[OACI 96] International Civil Aviation Organization, Montréal, Canada. *Procedures for Air Navigation Services  Rules of the Air and Air Tra  c Services*, 13th edition, novembre 1996. *Cité p.* 20

[Pallottino 02] Lucia Pallottino, Éric Féron et Antonio Bicchi. *Con  ict Resolution Problems for Air Tra  c Management Systems Solved with Mixed Integer Programming.* IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, pages 3–11, 2002. *Cité p.* 32

[Prosser 93] Patrick Prosser. *Hybrid Algorithms for the Constraint Satisfaction Problem.* Computational Intelligence, vol. 9, no. 3, pages 268–299, 1993. *Cité p.* 105

[Prot 10] Damien Prot, Christophe Rapine, Sophie Constans et Rémy Fondacci. *Using graph concepts to assess the feasibility of a sequenced air tra  c  ow with low con  ict rate.* European Journal of Operational Research, vol. 207, no. 1, pages 184–196, 2010. *Cité p.* 36

[Rey 10] David Rey, Christophe Rapine, Sophie Constans et Rémy Fondacci. *A MIP for Potential Con  icts Minimization by Speed Regulations.* 24th European Conference on Operational Research, Lisbonne, juillet 2010. *Cité p.* 33

[Rivière 06] Thomas Rivière. *Optimisation de graphes sous contrainte géométrique : création d un réseau de routes aériennes pour un contrôle Sector-Less.* Thèse de doctorat, Institut National Polytechnique de Toulouse, mai 2006. *Cité p.* 34

[RTCA 95] RTCA. *Report of the RTCA Board of Directors  Select Committee on Free Flight.* Rapport technique, janvier 1995. *Cité p.* 28

[Régin 05] Jean-Charles Régin. *AC-* : A Con  gurable, Generic and Adaptive Arc Consistency Algorithm.* Peter van Beek, éditeur, Principles and Practice of Constraint Programming – CP 2005, volume 3709 de *LNCS*, pages 505–519. Springer, 2005. *Cité p.* 105

[Sahni 76] Sartaj Sahni et Teofilo Gonzalez. *P-Complete Approximation Problems.* Journal of the ACM, vol. 23, no. 3, pages 555–565, juillet 1976. *Cité p.* 36

[SCA 06] Service de l'Information Aéronautique. *Annexe 2 à l arrêté du 3 mars 2006 modi  é (Services de la circulation aérienne)*, décembre 2006. *Cité p.* 18

[Schiex 92] Thomas Schiex. Informatique théorique et intelligence artificielle, chapitre 14. Problèmes de satisfaction de contraintes. Cépaduès, 1992. *Cité p.* 98

[Schiex 93] Thomas Schiex et Gérard Verfaillie. *Nogood Recording for Static and Dynamic Constraint Satisfaction Problems.* International Journal of Artificial Intelligence Tools, vol. 3, pages 48–55, 1993. *Cité p.* 105

[SESAR 07] SESAR. *SESAR Concept of Operations.* Rapport technique, SESAR Consortium, juillet 2007. *Cité p.* 30, 56

[STATFOR 10] STATFOR. *Eurocontrol Long-Term Forecast : IFR Flight Movements 2010-2030.* Rapport technique, Eurocontrol – Air Traffic Statistics and Forecasts, Brussels, décembre 2010. *Cité p.* 21

[Tran Dac 03] Huy Tran Dac, Philippe Baptiste et Vu Duong. *Airspace Sectorization by Constraint Programming.* RIVF'03, novembre 2003. *Cité p.* 97

[Van Hentenryck 92] Pascal Van Hentenryck, Helmut Simonis et Mehmet Dincbas. *Constraint Satisfaction Using Constraint Logic Programming.* Artificial Intelligence, vol. 58, no. 1-3, pages 113–159, 1992. *Cité p.* 97

[Van Hentenryck 95] Pascal Van Hentenryck. *Constraint Solving for Combinatorial Search Problems : A Tutorial.* Ugo Montanari et Francesca Rossi, éditeurs, Principles and Practice of Constraint Programming, volume 976 de *Lecture Notes in Computer Science*, pages 564–587. Springer, septembre 1995. *Cité p.* 98

[Vela 09a] Adan Vela, Senay Solak, William Singhose et John-Paul Clarke. *A Mixed Integer Program for Flight-Level Assignment and Speed Control for Con ict Resolution.* Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, pages 5219–5226, Shanghai, R.P. Chine, décembre 2009. *Cité p.* 33

[Vela 09b] Adan Vela, Senay Solak, Éric Féron, Karen Feigh, William Singhose et John-Paul Clarke. *A Fuel Optimal and Reduced Controller Workload Optimization Model for Con ict Resolution.* 28th Digital Avionics Systems Conference, 2009. *Cité p.* 33

[Villiers 04] Jacques Villiers. Automatisation du contrôle de la circulation aérienne « ERASMUS » – Une voie conviviale pour franchir le « mur de la capacité », volume 58 de *ITA études & documents.* Institut du Transport Aérien, juin 2004. *Cité p.* 30, 33

[Vose 91] Michael D. Vose. *Generalizing the notion of schema in genetic algorithms.* Artificial Intelligence, vol. 50, no. 3, août 1991. *Cité p.* 83

[Yin 93] X. Yin et N. Germay. *A Fast Genetic Algorithm with Sharing Scheme Using Cluster Analysis Methods in Multimodal Function Optimization.* R.F. Albrecht, N.C. Steele et C.R. Reeves, éditeurs, Artificial Neural Nets and Genetic Algorithms, pages 450–457. Springer-Verlag, 1993. *Cité p.* 91

[Zeghal 94] Karim Zeghal. *Vers une théorie de la coordination d actions. Application à la navigation aérienne.* Thèse de doctorat, Université Paris VI, 1994. *Cité p.* 31