

HEC MONTRÉAL

Réoptimisation de plans de vol sous contraintes
par
Jeanne Gagnon

Sciences de la gestion
(Analytique d'affaires)

Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M.Sc.)

Août 2015
©Jeanne Gagnon, 2015

Résumé

Avec une augmentation prévue d'au moins 50 % du trafic aérien d'ici vingt ans, les entités qui gèrent le trafic aérien ont besoin de nouveaux outils qui permettent de mieux gérer un espace aérien dont l'engorgement a des impacts bien réels, notamment les délais fréquents de vols. Les délais des avions commerciaux représentent chaque année des coûts substantiels de l'ordre de milliards de dollars. Sans pouvoir éliminer les retards, il est possible d'en réduire le nombre et les impacts.

Ce mémoire présente une méthode de replanification de trajectoire pour un avion qui a dévié de son plan de vol initial. Suite à un événement qui perturbe le vol, comme une mise à jour de la météo ou un retard au décollage, une nouvelle trajectoire tenant compte de la perturbation doit être générée. La trajectoire calculée est contrainte par plusieurs éléments, notamment la quantité de carburant disponible, les zones à éviter et les réglementations de carburant. L'objectif de la méthode est de trouver le vol le plus économique rapidement en déterminant le juste équilibre entre les frais encourus pour un retard (correspondances manquées, insatisfaction de la clientèle, salaires de l'équipage, etc.) et ceux permettant de modérer le retard, notamment le coût du pétrole brûlé en augmentant la vitesse de croisière ou le survol de zones tarifées. Dans la modélisation proposée, une grille en trois dimensions de forme elliptique représente l'espace aérien. La modélisation de l'avion qui navigue dans cette grille permet de prendre en compte la météo dans le calcul de la consommation de carburant. Le coût du temps est intégré au coût de la trajectoire grâce à l'indice de coût, qui convertit le temps en kilogrammes de carburant. Le problème est modélisé comme un plus court chemin avec contrainte de ressource et est résolu par un algorithme à correction d'étiquettes qui cherche la trajectoire optimale dans la grille en visitant séquentiellement les noeuds dans celle-ci. Au cours de cette recherche, plusieurs trajectoires sont examinées et chacune est représentée par une étiquette composée du coût et de la consommation de carburant. Cette étiquette permet de comparer les différentes trajectoires et de choisir les plus appropriées. Une règle de dominance permet de ne conserver qu'un sous-ensemble des trajectoires générées, ce qui permet de trouver rapidement le chemin optimal dans la grille.

Le comportement de l'algorithme a été étudié en fonction de la variation des différents coûts, des contraintes et des paramètres, tels la précision de la grille. Les contraintes de carburant font en sorte que l'avion transporte une quantité de carburant qui ne sera pas utilisée. Ces contraintes, tout comme les zones tarifées, introduisent dans la trajectoire des virages ou des changements d'altitude qui sont fonction de la quantité de carburant transportée. Les expériences numériques montrent que le temps de calcul augmente avec la précision de la grille et, dans une moindre mesure, avec l'indice de coût.

Mots clés : modélisation mathématique, programmation dynamique, plus court chemin avec contraintes, planification de trajectoire, transport aérien.

Abstract

Considering an expected increase of at least 50 % in air traffic within the next twenty years, the organizations managing air traffic need new tools which allow a better management of an airspace in which the congestion has tangible impacts, such as frequent flight delays. These delays have a substantial cost for the airlines of the magnitude of billions of dollars every year. While it is not possible to completely eliminate delays, reducing their incidence and impact is within reach.

In this master's thesis, a replanning method is presented for a plane that diverted from its original flight plan. When an unexpected event such as new weather data or delayed takeoff disrupts the flight, a new trajectory taking into account the disrupting event must be computed. The new trajectory is constrained by factors such as fuel availability, restricted areas and fuel regulations. The objective of this method is to quickly find the most economic flight through an adequate balance of delay-related costs (missed connections, customer dissatisfaction, crew wages, etc) and costs that can shorten the delay, in particular the cost of fuel burnt when increasing cruise speed and overflying fare zones. In the proposed modeling, a three dimensional elliptical grid represents the airspace. The modeling of the plane navigating this grid takes weather into account when computing fuel consumption. The cost of time is integrated through the use of a cost index converting time into a mass of fuel. The problem is modeled as a resource constrained shortest path and is solved by a label-correcting algorithm that searches the grid for the optimal trajectory by going through the nodes sequentially. A dominance rules retains only a subset of all computed trajectories which enables a quick research of the optimal path in the grid.

The algorithm's behaviour has been tested with respect to the variation of the different costs, constraints and parameters such as grid accuracy. Fuel regulations cause the aircraft to carry an extra amount of fuel that will not be burned. These regulations, as well as fare zones, force turns and altitude changes in the trajectory that are both dependent on the quantity of fuel carried. Numerical experiments show that computing time increases with grid accuracy and, to a lesser extent, with the cost index.

Key words : mathematical modeling, dynamic programming, constrained shortest path, trajectory planning, air transportation.

Table des matières

Résumé	iii
Abstract	v
Liste des figures	ix
Liste des tableaux	xi
Liste des abréviations	xiii
Remerciements	xv
Introduction	1
1 Description du problème	5
1.1 Description d'un plan de vol	5
1.2 Décisions	5
1.3 Coûts	6
1.3.1 Coût fixe	6
1.3.2 Coût du carburant	7
1.3.3 Coût du temps	7
1.4 Contraintes	8
1.4.1 Plan de vol	8
1.4.2 Altitude	9
1.4.3 Carburant	9
1.4.4 EDTO et règle des 60 minutes	11
1.4.5 Zones interdites	12
1.4.6 Enveloppe de vol	13
1.4.7 Virages	13
2 Revue de littérature	15
2.1 Méthodes d'optimisation de trajectoires	15
2.1.1 Méthodes directes et indirectes	15
2.1.2 Métaheuristiques	16
2.1.3 Programmation dynamique	17
2.2 Coûts et contraintes d'une trajectoire de vol	17
2.3 Méthodes de mise à jour	18
2.4 Problème de plus court chemin avec contraintes	19
2.4.1 Formulation de programmation dynamique	21

2.4.2	Relaxation lagrangienne	25
2.5	Algorithmes de résolution	25
2.5.1	Algorithme générique pour le RCSP	25
2.5.2	Algorithmes de résolution par poussée	27
2.5.3	Algorithme de résolution par traction	29
2.5.4	Prétraitement du réseau	30
3	Méthodologie	33
3.1	Modélisation du problème	33
3.1.1	Réseau	33
3.1.2	Variables	35
3.1.3	Étiquette	35
3.1.4	Données, paramètres et ensembles	36
3.1.5	Fonctions d'extension de ressource	37
3.1.6	Formulation des contraintes	38
3.2	Algorithme de résolution	40
4	Résultats	45
4.1	Planification initiale et mise à jour	45
4.2	Étude de la solution	47
4.2.1	Trajectoire sans contraintes	47
4.2.2	Effet des bornes de la consommation de carburant	48
4.2.3	Effet des coûts	56
4.3	Effets de la variation des paramètres	59
	Conclusion	67
	Annexes	
	Annexe A <i>International Standard Atmosphere</i>	71
	Annexe B Calcul de la consommation de carburant	73
	Annexe C Notions d'aéronautique	75
	Annexe D Modèle BADA et paramètres	77
	Annexe E Algorithme de dominance	79
	Annexe F Algorithme de calcul du carburant à transporter	81
	Annexe G Algorithme de calcul du dual	83
	Liste des références	84

Table des figures

1.1	Profil vertical de vol typique	6
1.2	Relation entre la consommation de carburant et la vitesse	7
1.3	Exemple de variation du coût marginal du délai en fonction de sa durée	7
1.4	Route magnétique d'un avion	9
1.5	Illustration de la quantité minimale de carburant à transporter	10
1.6	Requis de carburant en fonction des seuils franchis	12
1.7	Illustration des ETP pour le scénario critique d'EDTO	12
1.8	Exemple d'un domaine de vol	13
2.1	Exemple de graphe $G = (V, E)$	20
2.2	Exemple de liste d'étiquettes au nœud i	23
2.3	Deux listes d'étiquettes	24
2.4	Deux techniques pour étendre les chemins	26
2.5	Structure de stockage pour l'implémentation de Dial de l'algorithme de Dijkstra	29
2.6	Mise à jour d'une liste d'étiquettes par traction	31
2.7	Réduction de l'espace de solutions à une ellipse	32
3.1	Cylindre elliptique et ses paramètres	33
3.2	Comparaison de différentes formes de réseau	34
3.3	Illustration du découpage en tranche	34
3.4	Nœud avec 5 arcs sortants	35
3.5	Illustration de la quantité maximale consommée à un nœud	42
3.6	Illustration des deux sens possibles pour le calcul	42
4.1	Comparaison de la consommation de trajectoires calculées par les algorithmes de planification initiale et de mise à jour	47
4.2	Comparaison de deux trajectoires avec des quantités de carburant disponibles différentes	49
4.3	Comparaison des chemins possibles entre le départ et l'arrivée	50
4.4	Bornes supérieures de consommation calculées avec les contraintes de carburant	51
4.5	Comparaison de trajectoires avec trois seuils de l'EDTO différents	52
4.6	Comparaison de chemins avec deux quantités de carburant disponibles	53
4.7	Bornes supérieures de consommation pour l'élagage par infaisabilité	54
4.8	Bornes supérieures de consommation calculées avec les contraintes de carburant combinées aux bornes de l'élagage par infaisabilité	55
4.9	Effet de l'élagage par infaisabilité en fonction du CI	56
4.10	Proportion des étiquettes éliminées par le dual	57

4.11 Impact de l'utilisation de l'élagage par infaisabilité en fonction de la quantité de carburant disponible	57
4.12 Effet de l'élagage par coupe en fonction du CI	58
4.13 Effet de l'élagage par coupe en fonction du CI avec peu de carburant disponible . .	59
4.14 Effet de la zone tarifée en fonction du carburant disponible	60
4.15 Effet de la variation du CI sur le calcul du RCSP	61
4.16 Effet de la variation de la quantité de carburant disponible sur le calcul du RCSP .	62
4.17 Effet de la variation de la précision de la grille sur le temps de calcul requis	63
4.18 Temps de calcul en fonction de la précision de la grille	63
4.19 Consommation de carburant en fonction du CI dans des grilles de précisions différentes	64
4.20 Illustration des manœuvres de montée et de descente	64
4.21 Comparaison de trajectoires dans des grilles de précisions différentes	65
C.1 Diagramme de corps libre de l'avion	75
C.2 Angle d'incidence	75
C.3 Angle de roulis	75
C.4 Surface et envergure des ailes d'un avion	75
E.1 Entrées et résultat de l'algorithme de dominance	79

Liste des tableaux

1.1	Niveaux de vol permis en fonction de l'orientation par rapport au Nord magnétique	9
4.1	Comparaison de temps de vol et de coût entre l'algorithme de planification initiale et l'algorithme de mise à jour	48
4.2	Consommation de carburant maximale en fonction de l'altitude	49
4.3	Coût et durée d'un vol en fonction du seuil maximal et du carburant disponible . .	51
4.4	Longueur du segment en fonction de la précision de la grille	65
A.1	Valeurs des paramètres pour les calculs de l'ISA	72

Liste des abréviations

Aéronautique

ATS	<i>Air Traffic Services</i>
CI	indice de coût (<i>Cost Index</i>)
EDTO	<i>Extended Diversion Time Operation</i>
ETP	<i>Equal Time Point</i>
FAA	Federal Aviation Administration
IFR	<i>Instrument Flight Rule</i>
ISA	<i>International Standard Atmosphere</i>
LRC	<i>Long Range Cruise</i>
MRC	<i>Maximum Range Cruise</i>
NOTAM	<i>Notice to Airmen</i>
OACI	Organisation de l'Aviation Civile Internationale
OEI	<i>One Engine Inoperative</i>
TOC	<i>Top of Climb</i>
TOD	<i>Top of Descent</i>
VFR	<i>Visual Flight Rule</i>

Recherche opérationnelle

DAG	graphe orienté acyclique (<i>Directed Acyclic Graph</i>)
PRM	<i>Probabilistic Roadmaps</i>
RCSP	problème de plus court chemin avec contraintes
RRT	<i>Rapidly-exploring Random Trees</i>

Remerciements

J'aimerais remercier mon directeur Sylvain Perron qui m'a donné l'opportunité de travailler sur ce projet. Son appui, sa disponibilité et sa patience tout au long du déroulement du projet m'ont permis de me dépasser. Ma reconnaissance va aussi à mon codirecteur, Steven Dufour.

Merci à Charles Pestieau de m'avoir accordé sa confiance et d'avoir pris tout le temps nécessaire pour répondre à mes (nombreuses) questions. Je garderai un bon souvenir de ces discussions.

J'aimerais aussi remercier David Saussié et Thierry Warin pour leur appui.

Je remercie également Wissem Maazoun et Caroline Dietrich pour leur générosité et leur gentillesse.

Merci à Bastian, à mes parents et à ma famille pour leur support et leurs précieux conseils. Ils ont été présents pour moi pendant les hauts et les bas des deux dernières années et ils ont toute ma reconnaissance pour leur patience.

Merci à mes collègues de bureau ainsi qu'à Adélaïde, Ivan, Julie et Mélik pour leur joyeuse compagnie.

Introduction

Cette recherche s'inscrit dans un projet d'envergure dont l'objectif est de permettre aux acteurs du monde de l'aviation civile de faire face à l'augmentation prévue du trafic aérien. Airbus (2014) estime que le trafic aérien de passager augmentera en moyenne de 2,6% chaque année dans les économies avancées. Selon l'International Air Transport Association (2015), le marché nord américain poursuivra une croissance moyenne de 3,6% jusqu'en 2034. En somme, une croissance d'au moins 50% du trafic aérien passager est attendue d'ici 20 ans en Amérique du Nord. Cette augmentation devrait se traduire principalement par une augmentation de la fréquence des vols et de la taille des avions (Airbus, 2014). Ces prévisions de croissance n'apportent pas que de bonnes nouvelles pour les compagnies aériennes qui évoluent dans un espace aérien déjà engorgé. Les données du *Bureau of Transportation Statistics* (2015) montrent que près du quart des vols sont arrivés avec au moins 15 minutes de retard aux États-Unis en 2014. Plus du quart de ces délais sont attribués au *National Airspace System*, entre autres en raison du volume élevé d'avions. Il y a donc lieu de penser qu'une augmentation importante du trafic aérien n'est pas envisageable avec les outils actuels. Effectivement, les délais représentent des coûts importants pour les compagnies aériennes et incluent, entre autres, les salaires des équipages, la consommation de carburant, les frais d'aéroport, les compensations et le déroutement des passagers ainsi que la perte de parts de marché reliée à l'insatisfaction (Cook et al., 2004). La congestion peut même provoquer des baisses de prix (Forbes, 2008). Cook et al. (2009) estiment qu'en Europe les délais des compagnies aériennes ont coûté plus d'un milliard d'euros en 2007. En outre, ce chiffre n'inclut pas les pertes encourues par les passagers eux-mêmes en conséquence d'un retard. En bref, les prévisions de croissance de la circulation aérienne et la congestion du système de contrôle aérien motivent le développement d'outils pour optimiser le fonctionnement de l'espace aérien et les opérations se déroulant dans celui-ci.

Pour les compagnies aériennes, cette optimisation passe par une utilisation adéquate des ressources. Dans le cas des délais, une utilisation adéquate des ressources consiste à trouver le juste équilibre entre les frais encourus pour un retard et ceux permettant de modérer le retard, notamment le coût du pétrole brûlé en augmentant la vitesse de croisière ou le survol de zones tarifées. Cet équilibre peut être atteint grâce à une planification adéquate du vol (Cook et al., 2009). Cette planification peut être réalisée à deux moments. D'une part, on peut, avant l'approvisionnement en carburant, réaliser la planification initiale. Celle-ci permet de déterminer la trajectoire et la quantité de carburant à emporter. Le problème de la planification initiale a été étudié par de nombreux auteurs. Leurs travaux seront décrits au chapitre 2. D'autre part, on peut, après l'approvisionnement en carburant, mettre à jour le plan de vol, c'est-à-dire déterminer une nouvelle trajectoire avec une quantité de carburant fixée, un problème différent du premier et largement moins documenté. Pourtant, cet ajustement en cours de vol est important puisque l'utilisation d'un plan de vol optimisé permet des économies substantielles (Altus, 2009).

Dans le cadre de ce projet, une technique calculant précisément les coûts a été développée pour déterminer des trajectoires grâce à un algorithme de plus court chemin. Cette méthode optimise la trajectoire, mais sans contraintes, à l'aide de méthodes d'intégration numérique qui tiennent compte de la météo. Le coût du temps est intégré à l'aide d'un indice de coût. L'extension logique de ce travail est de pouvoir replanifier le vol après l'approvisionnement en carburant. C'est donc sur la mise à jour du plan de vol que porte ce projet de recherche. Plus précisément, on cherche à développer une méthode permettant de déterminer la trajectoire la moins coûteuse avec une quantité donnée de carburant. La nécessité de développer une nouvelle méthode provient du fait que l'algorithme de la planification initiale ne peut prendre en compte cette quantité. La méthode doit donc prendre en compte les contraintes propres à la mise à jour tout en utilisant les méthodes numériques développées pour la planification initiale. Elle doit être précise et rapide afin de permettre un bon ajustement en réponse à une perturbation. À plus long terme, un calcul rapide peut être utilisé dans la gestion du trafic aérien, un autre aspect du projet. Effectivement, la mise à jour permettrait au contrôle aérien de connaître les coûts reliés aux déviations qu'il impose et de les répartir équitablement entre les compagnies. Ce travail permettra de déterminer s'il est possible de trouver la trajectoire optimale en un temps de calcul de l'ordre des secondes. L'approche retenue pour la recherche est une méthode de programmation dynamique, la méthode de plus court chemin avec contraintes de ressources. Elle s'apparente à l'approche utilisée pour la planification initiale, ce qui permet une bonne compatibilité avec les méthodes numériques disponibles. Puisque l'approche permet d'intégrer des contraintes, elle requiert habituellement un temps de calcul plus long. La méthode développée doit donc intégrer des stratégies pour accélérer la résolution.

L'approche retenue se démarque de celles qui se trouvent dans la littérature reliée à la mise à jour, car elle assure que la trajectoire optimale sera évaluée et elle permet de ne chercher la trajectoire optimale que parmi les trajectoires réalisables.

La trajectoire doit être recalculée dès qu'une nouvelle information ou un événement perturbateur est connu. L'événement peut modifier les paramètres temporels, par exemple si le vol est en retard et que les vitesses doivent être recalculées. La quantité de carburant disponible peut diminuer, notamment si l'avion consomme plus de carburant que prévu au sol et que la quantité disponible pour le reste du trajet est réduite. Les conditions de vol peuvent changer si la météo est mise à jour et qu'alors la trajectoire peut être modifiée pour éviter les obstacles ou profiter de situations favorables. Finalement, l'avion peut dévier de la trajectoire prévue, par exemple à la suite de consignes du contrôle aérien. Ces quatre types de perturbations modifient les contraintes ou les coûts de la trajectoire. Ces éléments sont décrits au chapitre 1.

La réduction des coûts est un sujet qui intéresse un grand nombre d'acteurs de l'industrie de l'aviation civile, une industrie hautement compétitive. La génération de trajectoire a donc été le sujet de recherche de nombre de chercheurs universitaires et d'industriels. Le chapitre 2 recense différentes manières de générer des trajectoires et des approches pour le problème de planification initiale proposées par les acteurs de l'industrie. Dans la suite de ce chapitre, le lecteur constatera que la mise à jour est un problème peu étudié bien que les changements ne soient pas rares au cours d'un vol. L'approche choisie sera explorée en dernier lieu.

Ainsi, au chapitre 3, toutes les variables qui permettent de décrire le modèle seront établies et décrites pour permettre de formuler les éléments du chapitre 1 sous forme mathématique. La deuxième partie de ce chapitre décrit le modèle mathématique de programmation dynamique développé. Finalement, un algorithme sera proposé pour le résoudre et il sera décrit dans le détail.

Les résultats obtenus grâce à la méthodologie développée seront présentés au chapitre 4. Dans un premier temps, la forme de la solution fournie sera présentée et comparée à celle de

la planification initiale. Dans un deuxième temps, l'impact des différentes contraintes et des différents coûts sur la performance et la qualité de la solution sera analysé.

Chapitre 1

Description du problème

Ce chapitre rassemble l'ensemble des éléments qui devront être pris en compte dans la modélisation du problème. D'abord, le plan de vol à produire est décrit et les décisions à prendre pour le créer sont définies. À ces décisions sont associés des coûts qui sont listés. Les décisions qui sont prises doivent respecter des contraintes réglementaires et physiques qui sont détaillées dans la dernière section de ce chapitre.

1.1 Description d'un plan de vol

Avant tout, la section du vol qui sera optimisée doit être précisée. Pour représenter un vol, on peut utiliser deux profils : le profil horizontal et le profil vertical. Le profil horizontal montre la position de l'avion (longitude et latitude). Le profil vertical montre l'altitude de l'avion en fonction de la distance parcourue. La figure 1.1 présente un profil de vol vertical typique. Pour faciliter la lecture, les échelles verticales et horizontales ne respectent pas les proportions. Entre la fin du décollage et la fin de la montée, *Top of Climb* (TOC), le pilote exécute une série de manœuvres connues d'avance. Il en va de même entre le début de la descente, *Top of Descent* (TOD), et l'atterrissage. Ces étapes du vol représentent peu d'intérêt pour l'optimisation. Pour minimiser la consommation de carburant, les manœuvres sont repensées par des organisations comme l'OACI (2014c). Ce projet de recherche se concentre donc sur l'étape de croisière se situant entre le TOC et le TOD. À la figure 1.1, l'avion change d'altitude une fois, sa croisière se fait donc en deux paliers. Le nombre de paliers varie en fonction de la longueur du vol.

1.2 Décisions

La trajectoire de l'avion dans le plan de vol est décrite par une succession de points de passage et de routes. Certains points de passage sont fixes et portent un nom. Par exemple, un point de passage nommé TRENA se situe en $50^{\circ} 26' 7,00''$ N de latitude et $124^{\circ} 14' 11,00''$ O de longitude. D'autres points de passage sont indiqués directement par leur coordonnées sur le plan de vol. Il en va de même pour les routes. Par exemple, le point TRENA est relié au point UAR par la route J488. Tous les points ne sont pas nécessairement reliés par des routes identifiées. Le point UAR, par exemple, peut être atteint depuis l'aéroport YYD par un trajet direct, c'est-à-dire que l'avion vole en ligne droite entre UAR et YYD.

L'espace aérien est donc constitué d'une multitude de points de passage connectés par des routes. Cette structure s'apparente à celle d'un réseau dans lequel les points de passage sont des nœuds et les routes, des arcs. Afin que la méthode développée soit le plus compatible possible

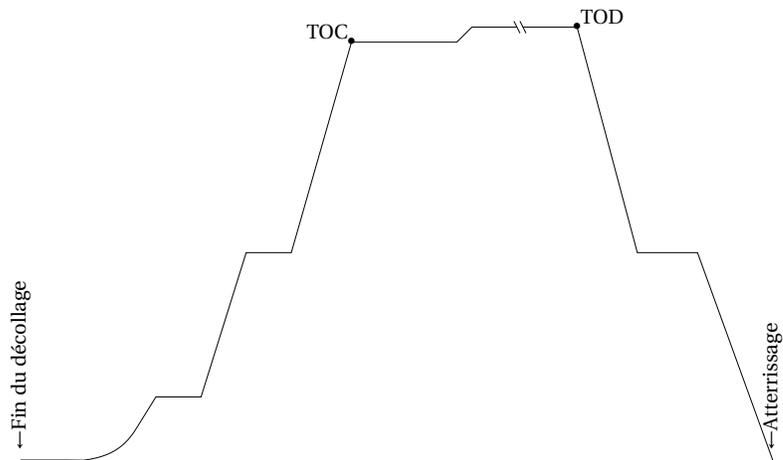


FIGURE 1.1: Profil vertical de vol typique

avec l'espace aérien, le problème sera modélisé comme un réseau. Dans celui-ci, chaque nœud possède une position dans l'espace aérien (longitude, latitude et altitude).

La méthode devra déterminer les points de passage, la trajectoire entre chaque paire de points de passage, la vitesse de l'avion ainsi que son altitude. Dans le réseau, ces décisions sont équivalentes à choisir un ensemble de nœuds qui forment une trajectoire et à déterminer pour chacun une heure de passage. Choisir un nœud est équivalent à choisir un point de passage et une altitude. L'arc qui joint une paire nœuds consécutifs donne la trajectoire. En utilisant les heures de passage, la vitesse est connue. Puisque les nœuds sont décrits par leur position dans l'espace en trois dimensions, les profils horizontal et vertical sont optimisés simultanément. La quantité de carburant ne fait pas partie des décisions, puisqu'au moment de la mise à jour, l'avion a déjà été ravitaillé en carburant. Les décisions prises doivent respecter les contraintes présentées à la section 1.4 tout en minimisant les coûts décrits à la section 1.3.

1.3 Coûts

De manière générale, le coût d'un vol d'avion peut être calculé comme

$$C = C_F + C_T \Delta T + C_C \Delta C \quad (1.1)$$

où C_F sont le coût fixe indépendant du temps, C_T , le coût du temps par minute de vol, ΔT , le temps de vol en minutes, C_C , le coût du carburant par kilogramme et ΔC , la quantité de carburant utilisée pour le vol en kilogrammes. Les trois types de coûts sont décrits dans les paragraphes suivants.

1.3.1 Coût fixe

Dans le coût fixe, seul le survol de zones tarifées sera considéré. Chaque avion doit payer des droits pour circuler, atterrir ou décoller dans un territoire. Ces frais sont payables à l'entité qui gère l'espace aérien, comme la Federal Aviation Administration (FAA) aux États-Unis, Nav Canada au Canada et Eurocontrol en Europe. Certains territoires sont coûteux à survoler, mais il peut être intéressant de les survoler s'ils permettent des gains importants en temps ou une réduction significative de la consommation de carburant.

1.3.2 Coût du carburant

Le coût du carburant est calculé comme la consommation de carburant en kilogrammes multipliée par le prix du carburant. La consommation de carburant et la vitesse varient approximativement comme la courbe de la figure 1.2. La consommation est minimale à une certaine vitesse (*Maximum Range Cruise* (MRC)) indiquée sur la figure. Cependant, un retard pourrait justifier d'accélérer, même si le coût en carburant augmente. En outre, une décélération peut permettre d'éviter une coûteuse attente au-dessus d'un aéroport congestionné.

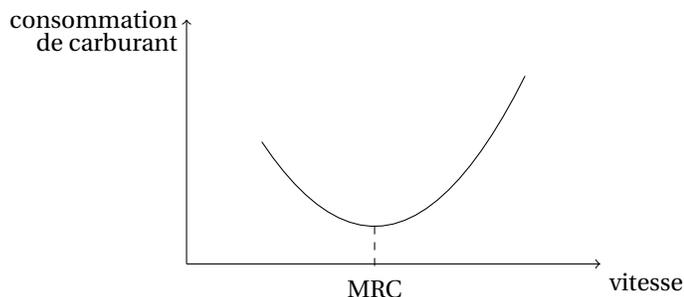


FIGURE 1.2: Relation entre la consommation de carburant et la vitesse

1.3.3 Coût du temps

Un retard à l'arrivée entraîne des coûts pour les compagnies aériennes. Les coûts les plus importants sont ceux qui sont liés aux passagers qui manquent leur correspondance, puisqu'il faut modifier les itinéraires et compenser les passagers (Cook et al., 2007). Pour compenser le retard d'un vol, les compagnies aériennes disposent de plusieurs outils comme annuler ou retarder d'autres vols, modifier l'assignation des équipages ou des avions, etc (Rosenberger et al., 2002). Certaines mesures, par exemple faire voler un avion vide, ont un coût évident. D'autres coûts sont plus difficiles à quantifier, notamment la perte de parts de marché due aux retards (Cook et al., 2007). Le coût du délai en fonction du temps ne varie pas linéairement. La figure 1.3 montre un exemple simple de l'augmentation du coût marginal du vol en fonction de la durée du délai. Les lignes pointillées décrivent deux éléments de coût, par exemple les salaires et les dédommagements aux passagers. La somme de ces deux coûts (ligne pleine) est une fonction complexe et irrégulière.

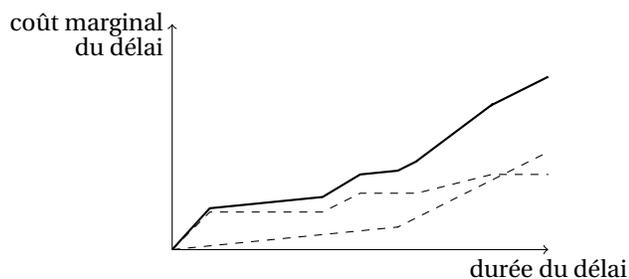


FIGURE 1.3: Exemple de variation du coût marginal du délai en fonction de sa durée (Cook et al., 2004)

Dans le modèle présenté à la section 3.1, le coût du temps est pris en compte dans le coût du délai. La figure 1.3 fournit une indication de la complexité de cette fonction. Pour contourner le

problème, tous les algorithmes du projet (planification initiale, mise à jour, etc.) utilisent le indice de coût (*Cost Index*) (CI), qui permet d'intégrer le coût du temps dans la fonction récursive. Le CI met en relation le coût du temps et le coût du carburant comme en (1.2).

$$\text{CI [kg/min]} = \frac{\text{coût du temps [$/min]}}{\text{coût du carburant [$/kg]} \quad (1.2)$$

Ce rapport permet de décider s'il vaut la peine de voler plus vite pour arriver plus tôt (et ainsi diminuer le coût du délai) ou s'il vaut mieux voler plus lentement pour économiser du carburant. Dans le calcul, le coût du temps ne prend en compte que les coûts qui varient avec l'heure d'arrivée, notamment certains coûts d'entretien, certaines composantes salariales, les compensations pour les passagers et les frais d'aéroport (Cook et al., 2004; Airbus Flight Operations Support & Line Assistance, 1998). Le CI est donc étroitement lié à l'heure d'arrivée. Au-delà de l'heure d'arrivée prévue, la valeur du CI augmente avec le coût du délai. Le module qui détermine le CI a été développé par un autre membre de l'équipe, Mootez Ben Nasr (travail non publié à la rédaction du présent document).

1.4 Contraintes

Deux types de contraintes existent : les contraintes reliées à la réglementation et celles reliées à la physique de l'avion. La réglementation est vaste. Elle régit le contenu du plan de vol ainsi que les altitudes permises pour le vol. Elle définit les quantités minimales de carburant à transporter. Elle limite la distance maximale entre la trajectoire et les aéroports par l'*Extended Diversion Time Operation* (EDTO). Elle interdit aussi le survol de certaines zones. Les contraintes physiques considérées sont l'enveloppe de vol et le rayon de virage.

1.4.1 Plan de vol

En tout temps, la variation de vitesse permise est de $\pm 5\%$ par rapport à la vitesse spécifiée dans le plan de vol (OACI, 2005). En outre, l'ensemble des points de passage du plan de vol doit contenir au minimum (Nav Canada, 2014) :

- les points où il y a changement d'altitude ;
- les points où il y a changement de vitesse ;
- les points de passage de *Visual Flight Rule* (VFR) à *Instrument Flight Rule* (IFR) ou de IFR à VFR ;
- si la route n'est pas désignée par l'*Air Traffic Services* (ATS), au moins un point tous les 370 km.

Les règlements (OACI, 2005) indiquent encore que les points de passage doivent être survolés à une heure précise, cependant, sauf dans certains pays comme la Russie, ce n'est plus le cas en pratique. Selon les personnes que nous avons rencontrées chez Air Canada le 17 novembre 2014, il suffit en général d'informer les contrôleurs du changement de trajectoire. De plus, la tendance est à libéraliser certains espaces aériens et à retirer les points de passage (SE-SAR Consortium, 2012). Sinon, les variations permises par rapport aux points de passage soumis dans le plan de vol sont :

- ± 2 minutes par rapport à l'heure prévue au point de passage (OACI, 2005) ;
- ± 90 mètres par rapport à l'altitude prévue (OACI, 2002).

1.4.2 Altitude

La première contrainte est imposée par le fabricant, sous la forme d'une altitude maximale spécifiée pour chaque avion. La deuxième contrainte est d'ordre réglementaire. En fonction de la direction de l'avion, les altitudes de vol permises par l'Organisation de l'Aviation Civile Internationale (OACI) pour les vols IFR varient tel que décrit au tableau 1.1. On utilise la route magnétique (*magnetic track*) de l'avion (illustrée à la figure 1.4), c'est-à-dire la direction de l'avion par rapport au Nord magnétique, sans prendre en compte le cap de l'avion.

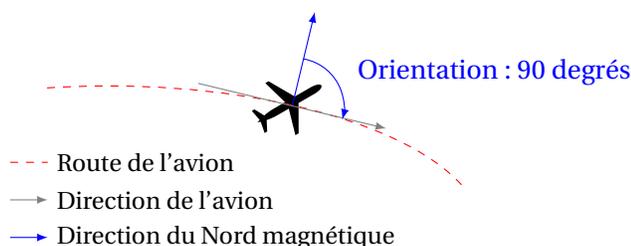


FIGURE 1.4: Route magnétique d'un avion

Les niveaux assignés en fonction de la direction sont donnés au tableau 1.1. Chaque niveau est donné en multiple de 100 pieds. L'abréviation FL 100 indique un niveau de vol de 10 000 pieds. Par exemple, à la figure 1.4, l'angle entre la route de l'avion et le Nord magnétique est de 90 degrés. Selon le tableau 1.1, cet avion devrait donc voler à un niveau de vol pour lequel les deux premiers chiffres forment un nombre impair, par exemple FL 350.

TABLEAU 1.1: Niveaux de vol permis en fonction de l'orientation par rapport au Nord magnétique (OACI, 2005)

de 0 à 179 degrés	de 180 à 359 degrés
FL 010	FL 020
FL 030	FL 040
FL 050	FL 060
...	...
FL 370	FL 380
FL 390	FL 400
FL 410	

1.4.3 Carburant

La quantité minimale de carburant à transporter est donnée par l'OACI à l'annexe 6 de la Convention internationale de l'aviation civile (2010) à la section 4.3.6. Le pilote peut toujours décider d'ajouter du carburant discrétionnaire (*discretionary fuel*), c'est-à-dire une quantité ajoutée à la quantité que la méthode aura calculée.

On peut diviser la quantité de carburant en deux blocs, un bloc qui comprend une quantité de carburant qui ne dépend pas de la trajectoire choisie et un bloc qui comprend une quantité de carburant qui dépend de la trajectoire choisie, des aéroports de dégagement (tel que décrit à la définition 1.1) sélectionnés et de la destination alternative (définie à la définition 1.2) choisie.

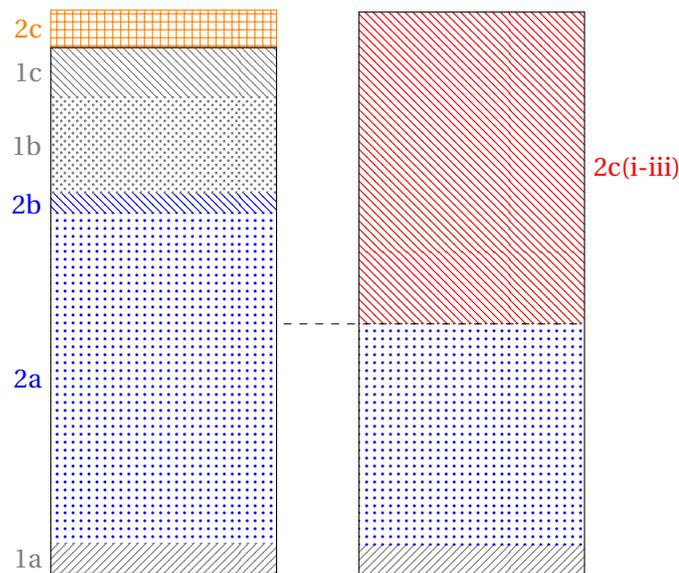


FIGURE 1.5: Illustration de la quantité minimale de carburant à transporter (OACI, 2014b). À gauche, tous les blocs de carburant sont détaillés. À droite, on considère le pire scénario entre 2(c)i, ii et iii. La ligne pointillée est le point le plus critique du trajet.

Définition 1.1 (Aéroport de dégivrage). Aéroport vers lequel le vol peut être poursuivi lorsqu'il est inopportun d'atterrir à l'aérodrome de destination prévu (Gouvernement du Canada, 1996).

Définition 1.2 (Destination alternative). Aéroport de dégivrage se situant à proximité de l'aéroport de destination.

Les différents blocs sont illustrés à la figure 1.5. La numérotation utilisée dans la figure correspond à la numérotation des éléments dans le texte.

1. Dans le bloc fixe, en gris à la figure 1.5, on doit compter :
 - (a) le carburant pour la circulation au sol (*taxi fuel*) pour les aéroports de départ et de destination ;
 - (b) la réserve de carburant pour la destination alternative (*destination alternate fuel*) :
 - sans destination alternative - quantité requise pour voler 5 minutes dans des conditions normales à vitesse d'attente (*holding speed*) à 450 mètres au-dessus de l'aéroport de destination ;
 - avec une destination alternative - quantité requise pour effectuer la procédure suivante :
 - i. exécuter une approche interrompue à l'aéroport de destination,
 - ii. monter à l'altitude de croisière,
 - iii. se rendre à l'aérodrome de dégivrage,
 - iv. approcher l'aérodrome de dégivrage,
 - v. atterrir à l'aérodrome de dégivrage.
 - avec deux destinations alternatives - quantité calculée comme pour une destination alternative en utilisant l'aérodrome de dégivrage le plus éloigné de l'aéroport de destination.

- (c) la réserve finale (*final reserve fuel*) - dans le cas des avions commerciaux, la quantité requise pour voler 30 minutes dans des conditions normales à vitesse d'attente à 450 mètres au-dessus de l'élévation de l'aéroport de destination (Coulmeau, 2005).
2. Dans le bloc qui dépend de la trajectoire, il faut inclure :
- (a) le carburant jusqu'à destination (*trip fuel*) ;
 - (b) la réserve de carburant en cas d'imprévu (*contingency fuel*) - 5% de 2a, mais au moins la quantité requise pour voler 5 minutes dans des conditions normales à vitesse d'attente à 450 mètres au-dessus de l'élévation de l'aéroport de destination ;
 - (c) le carburant additionnel (*additionnal fuel*) (illustré en orange à la figure 1.5) - carburant additionnel à ajouter si la somme des quantités calculées en 1, 2a et 2b n'est pas suffisante pour satisfaire les trois conditions suivantes :
 - i. permettre, dans l'éventualité d'une panne du moteur ou d'une dépressurisation qui survient au point le plus critique et qui requiert le plus de carburant, de :
 - A. descendre à l'altitude requise et se rendre à un aéroport de dégagement ;
 - B. voler 15 minutes dans des conditions normales à vitesse d'attente à 450 mètres au-dessus de l'élévation de l'aéroport de dégagement ;
 - C. faire une manœuvre d'approche et un atterrissage.
 - ii. permettre à un avion en situation d'EDTO de se conformer au scénario critique d'EDTO de l'État de l'exploitant (voir 1.4.4), s'il y a lieu ;
 - iii. parer à toutes les situations qui ne sont pas décrites ci-haut.

1.4.4 EDTO et règle des 60 minutes

L'EDTO et la règle des 60 minutes limitent la distance à laquelle les aéronefs peuvent se trouver par rapport aux aéroports de dégagement. À noter, les aéroports de départ et de destination sont considérés comme des aéroports de dégagement (OACI, 2010).

Tous les seuils sont donnés en minutes et la distance est calculée en fonction du nombre de moteurs de l'aéronef :

- deux moteurs : à *International Standard Atmosphere* (ISA) (voir l'annexe A), sans vent et avec un seul moteur en fonction à la vitesse approuvée (*One Engine Inoperative speed*) pour le type d'aéronef ;
- plus de deux moteurs : calculée à ISA sans vent et avec tous les moteurs en fonction à la vitesse approuvée (*All Engine Operative speed*) pour le type d'aéronef.

La vitesse est souvent donnée sous la forme d'un rapport Mach/IAS. L'altitude est fournie par le constructeur et dépend de la vitesse et de la masse de l'aéronef.

Il existe trois seuils importants (OACI, 2010) qui sont illustrés à la figure 1.6. Ont été retenues ici les conditions pertinentes à la mise à jour du plan de vol.

1. Le premier est fixé par l'OACI à 60 minutes et est valide pour tous les types d'aéronef. À partir de ce seuil, les aéroports de dégagement doivent être identifiés ;
2. Le deuxième, le seuil EDTO est fixé par l'État de l'opérateur pour chaque type d'aéronef. Il est de 60 minutes au minimum. À partir de ce seuil, la quantité de carburant doit être suffisante pour se conformer au scénario critique d'EDTO ;
3. Le troisième, le temps de diversion maximal, est fixé par l'État de l'opérateur pour chaque type d'aéronef et pour chaque opérateur. Ce seuil ne peut être dépassé.

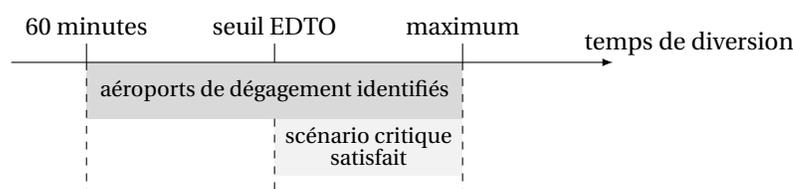


FIGURE 1.6: Requis de carburant en fonction des seuils franchis (OACI, 2010)

Pour satisfaire le scénario critique, la quantité de carburant doit être suffisante pour se rendre à un aéroport de dégagement à un point critique dans la pire des situations suivantes :

- une dépressurisation de la cabine ;
- une panne de moteur ;
- la combinaison des deux.

Le point critique au sens de l'EDTO est l'un des *Equal Time Point* (ETP), c'est à dire l'un des points se trouvant sur la trajectoire à partir desquels le temps de vol pour se rendre à deux aéroports de dégagement est le même, en prenant en compte les vents et la température, avec un seul moteur en opération (OACI, 2014a). Deux points ETP sont illustrés à la figure 1.7.

La panne de moteur seule n'est jamais limitante (OACI, 2014a). Dans le cas d'une dépressurisation, l'avion doit en premier lieu effectuer une descente d'urgence à vitesse maximale jusqu'à FL100 ou à l'altitude minimale prescrite pour la région. S'il y a panne de moteur, la vitesse de croisière est la vitesse *One Engine Inoperative* (OEI) approuvée. Sinon, la vitesse de croisière est la vitesse *Long Range Cruise* (LRC).

Le carburant doit être suffisant pour :

1. se rendre à l'aéroport de destination ;
2. voler 15 minutes dans des conditions normales à vitesse d'attente à 450 mètres au-dessus de l'élévation de l'aéroport de dégagement ;
3. faire une manœuvre d'approche et un atterrissage aux instruments.

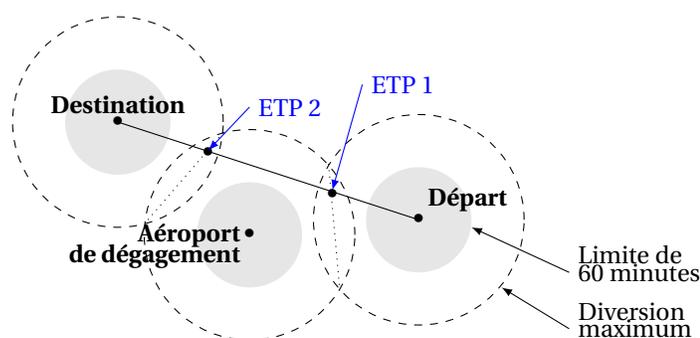


FIGURE 1.7: Illustration des ETP pour le scénario critique d'EDTO (Transport Canada, 2007)

1.4.5 Zones interdites

La trajectoire ne peut passer par les zones interdites. Les zones interdites peuvent être des zones réservées aux militaires, des zones interdites temporairement par les *Notice to Airmen* (NOTAM) (voir définition 1.3), etc.

Définition 1.3 (NOTAM). Avis donné aux aviateurs concernant la création, la modification ou l'état de tout service, installation ou procédure aéronautique, ou les dangers compromettant la sécurité aérienne, dont la connaissance est essentielle au personnel participant à des opérations aériennes (Gouvernement du Canada, 1996).

1.4.6 Enveloppe de vol

L'aérodynamique de l'avion limite les couples vitesse/altitude qui peuvent être choisis. La figure 1.8 donne un exemple de domaine de vol sous la forme d'une zone grise encadrée par trois courbes. Chaque avion possède une altitude maximale (en bleu sur la figure) et une vitesse maximale (en vert sur la figure). La courbe de décrochage à gauche de la figure illustre la vitesse minimale que doit maintenir l'avion pour maintenir son altitude. Les courbes varient selon l'altitude et le modèle d'avion. Ainsi, les mesures ne sont pas indiquées sur les échelles de la figure 1.8.

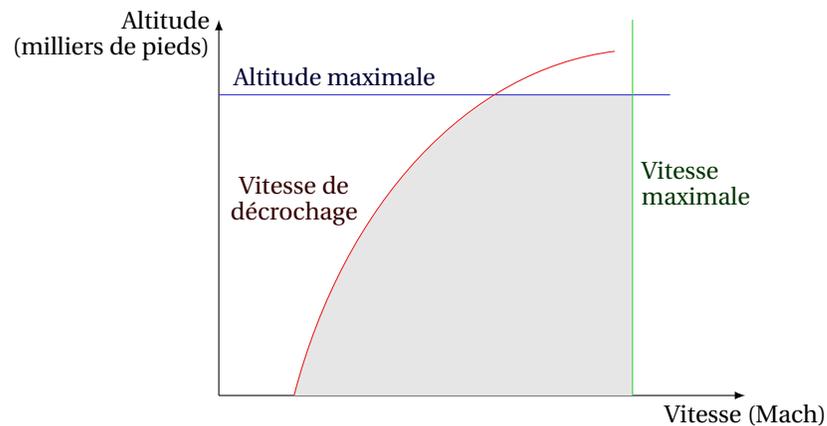


FIGURE 1.8: Exemple d'un domaine de vol

1.4.7 Virages

Typiquement, le taux de giration maximal est de 3 degrés par seconde. Les caractéristiques de l'avion et le confort des passagers limitent ce taux de giration.

Chapitre 2

Revue de littérature

Ce chapitre est consacré à la revue d'ouvrages de littérature qui présentent une pertinence pour le projet de recherche. Dans un premier temps, les approches les plus fréquemment utilisées dans la littérature pour optimiser des trajectoires seront présentées. Cette revue permettra de valider le choix du réseau pour la modélisation du problème. Ensuite, les différentes méthodes développées pour la création de trajectoires d'avion seront comparées selon les coûts et les contraintes qu'elles considèrent. Puis, la littérature concernant précisément la mise à jour de trajectoires sera revue. Le lecteur sera ainsi en mesure de comprendre la pertinence d'une nouvelle méthode de mise à jour. La suite du chapitre portera sur le problème de plus court chemin et les algorithmes pour le résoudre.

2.1 Méthodes d'optimisation de trajectoires

Cette section passe en revue trois familles de méthodes fréquemment utilisées pour l'optimisation de trajectoires. Les méthodes directes et indirectes qui seront présentées en premier se basent sur l'analyse des équations de vol. Deux métaheuristiques seront présentées dans la section suivante. La dernière section porte sur la programmation dynamique.

2.1.1 Méthodes directes et indirectes

Pour les méthodes directes et indirectes, le problème de l'optimisation de trajectoire est formulé comme un problème de contrôle optimal. Un système décrit par un ensemble d'équations différentielles ordinaires (2.1) est résolu.

$$\dot{x} = f(x(t), u(t), t) \quad (2.1)$$

La variable t est habituellement le temps. Le vecteur $x(t)$ décrit l'état du système au temps t et $u(t)$, les décisions prises au temps t . Les valeurs de $u(t)$ sont choisies pour tenter de minimiser (2.2), c'est-à-dire la valeur d'une fonction Φ au cours de l'intervalle de temps $[y_0, y_t]$ où le système est observé. La valeur initiale $y_0 = y(t_0)$ est donnée.

$$J = \min\{\Phi(x(t), t)\}, \quad t_0 \leq t \leq t_f \quad (2.2)$$

Ce système est transformé en un problème de programmation non linéaire.

La famille des méthodes directes regroupe plusieurs approches, notamment le tir direct, les tirs directs multiples et la collocation. Le problème de contrôle devient un problème de valeur

initiale, car $x(t)$ et $u(t)$ sont choisis. La taille du programme non linéaire varie en fonction de la méthode, la méthode de tir direct produisant les plus petits programmes (Betts, 1998).

Pour les méthodes indirectes, les conditions d'optimalité sont dérivées et résolues. Le problème se transforme ainsi en problème aux limites. Ce problème requiert parfois d'estimer avant la résolution les valeurs de certains paramètres. Encore une fois, elle regroupe plusieurs méthodes, notamment le tir indirect, les tirs indirects multiples. La dérivation des conditions d'optimalité peut être ardue. L'algorithme ne converge pas nécessairement, surtout si plusieurs valeurs doivent être estimées Betts (1998). Chakravarty (1995) utilise cette méthode pour minimiser la consommation de carburant sur un vol.

Dans l'ensemble, les méthodes directes trouvent souvent des minimaux locaux et produisent des résultats peu précis (von Stryk et Bulirsch, 1992). Les méthodes indirectes gèrent mal les contraintes de trajectoire Huang et al. (2012).

Wu (2012) utilise le contrôle optimal pour générer une trajectoire complète tenant compte de la météo. Une importance particulière est portée aux montées et aux descentes. Le CI est utilisé pour équilibrer le temps de vol et la consommation de carburant en fonction des besoins. La méthode de Yang et al. (2009) permet de trouver des trajectoires sans obstacle dans un environnement dynamique. Ces deux méthodes ne prennent pas en compte les contraintes de l'espace aérien décrites au chapitre 1. Soler et al. (2011) dérivent une formulation de programmation linéaire à variables mixtes du problème de contrôle optimal. Ils tiennent compte des points de passage, des vents et des coûts de survol. Cependant, leur méthode n'optimise que le profil horizontal du vol.

Pour expliquer la pertinence de leur méthode, Tang et Mukherjee (2000) indiquent que le contrôle optimal ne permet tout simplement pas de prendre en compte les règles de l'espace aérien. Devulapalli (2012) rejette le contrôle optimal pour la même raison. De plus, bien qu'il loue le fait que ces trajectoires obtenues par contrôle optimal soient lisses et continues, il considère que la convergence dépend trop de la valeur initiale, surtout lorsque les contraintes sont nombreuses.

2.1.2 Métaheuristiques

Les métaheuristiques sont des techniques d'optimisation qui permettent de trouver une « bonne » solution en un temps acceptable. Il existe un grand nombre de techniques, notamment le recuit simulé, les algorithmes génétiques, la recherche à voisinage variable, la recherche avec tabous, et chacune peut être appliquée à une variété de problèmes. Les métaheuristiques ne garantissent pas l'optimalité et, de manière générale, ne donnent pas d'approximation entre la solution trouvée et l'optimum du problème (Talbi, 2009).

Betts (1998) considère que les métaheuristiques ne sont pas appropriées pour des méthodes reposant sur l'analyse. Au contraire, Huang et al. (2012) considèrent que les algorithmes génétiques sont robustes, peu sensibles aux conditions initiales et assez performants. Girardet (2014) considère que la flexibilité des algorithmes permet de prendre en compte une grande variété des contraintes et de conditions, par exemple les vents. Dougui et al. (2013) utilisent un modèle de propagation de la lumière et les algorithmes génétiques pour générer des trajectoires qui évitent les obstacles. Deux problèmes d'importances font en sorte que l'utilisation des algorithmes génétiques n'est pas idéale pour le problème à l'étude. D'abord, l'algorithme peut converger vers un optimum local. De plus, un ajustement est nécessaire pour chaque problème étudié (Girardet, 2014). Finalement, Reeves et al. (2010) considèrent qu'il faut générer plusieurs solutions et choisir parmi celles-ci.

Huang et al. (2012) incluent aussi les algorithmes de colonie de fourmis dans leur revue. Ces algorithmes s'inspirent du comportement des colonies de fourmis qui recherchent un chemin entre une source de nourriture et leur nid. Cette technique converge lentement et tend à trouver des minima locaux. Elle est utilisée pour des applications en aérospatiale (Huang et al., 2012) et pour des problèmes multiobjectifs (Dorigo et al., 2010) qui ressemblent au problème à l'étude.

2.1.3 Programmation dynamique

La programmation dynamique est une technique de recherche qui permet de résoudre des problèmes de contrôle. Elle peut être continue ou discrète. Dans le cas continu, un problème de contrôle optimal est écrit sous la forme de l'équation d'Hamilton-Jacobi-Bellman. Le cas discret se base sur le principe d'optimalité selon lequel toutes les parties qui composent la solution sont elles-mêmes optimales.

Betts (1998) considère que la programmation dynamique n'est pas appropriée pour trouver un chemin unique. Effectivement, cette technique requiert énormément de calculs tel que l'avait constaté Bellman. Huang et al. (2012) concède que les calculs sont coûteux, mais fait valoir la précision et les qualités mathématiques de la technique. Girardet (2014) utilise l'équation de Hamilton-Jacobi-Bellman notamment, car l'optimalité est garantie. Elle solutionne le problème d'optimisation de trajectoire pour un scénario de type ciel ouvert (*open skies*). Les autres travaux utilisent la programmation dynamique discrète. Une partie importante de la littérature est centrée sur des techniques permettant de résoudre le fléau de la dimension (*curse of dimensionality*). Sidibé (2014) optimise un profil vertical de vol grâce à la programmation dynamique avec une technique qui mitige le fléau de la dimension. Il découpe le vol selon les phases d'un profil de vol typique pour former les étapes du problème de programmation dynamique. Il envisage plusieurs scénarios pour chacune des phases, par exemple en faisant varier la vitesse pour la montée. Un algorithme de plus court chemin permet de trouver la combinaison de scénarios à coût minimal pour le trajet complet. Devulapalli (2012) utilise l'algorithme A* (voir l'algorithme 2.4) pour résoudre le fléau de la dimension. Il optimise les profils vertical et horizontal séparément tout comme Schultz et Shaner (2001); Schultz et al. (2003); Cabos (2013). Tang et Mukherjee (2000) rejettent toutes les solutions qui considèrent le profil horizontal séparément du profil vertical en raison de l'interdépendance de la position, l'altitude et la vitesse. Leur méthode est en deux étapes : ils réduisent l'espace de solutions à l'aide des données météorologiques, puis utilisent la programmation dynamique pour trouver le plus court chemin dans cet espace. Maazoun (2015) optimise un vol dans une grille 3D de forme elliptique grâce à un algorithme de court chemin. La météo et le coût du temps sont intégrés dans le coût des arcs.

Étant donné les nombreuses contraintes auxquelles sont soumis les avions commerciaux, la programmation dynamique semble être l'approche à privilégier pour résoudre le problème de recherche. De plus, les modèles récents Tang et Mukherjee (2000); Maazoun (2015) montrent que les temps de calcul raisonnables sont à la portée des ordinateurs actuels.

2.2 Coûts et contraintes d'une trajectoire de vol

Dépendant des objectifs de recherche et de l'application, les coûts et les contraintes qui sont pris en compte dans la création de trajectoires varient. Ils sont présentés dans cette section.

Les différents algorithmes de la littérature prennent en compte plusieurs types de contraintes. Agam et Agam (2009) discutent des contraintes de carburant liées au point critique et à l'EDTO qui ont été présentées aux sections 1.4.3 et 1.4.4. Ce brevet s'attarde davantage au matériel et

aux interfaces entre les composantes. Boroujerdi et Uhlmann (1998) transforment le graphe initial et éliminent les chemins qui ne respectent pas les contraintes de virage et utilisent l'algorithme de Dijkstra pour trouver le plus court chemin. L'algorithme est repris dans un brevet (Boroujerdi et Uhlmann, 2002). Il est évident que la transformation proposée augmente significativement la taille du réseau. Par ailleurs, il est étonnant de constater que les auteurs ne justifient pas leur préférence pour cette méthode par rapport à une méthode où l'information sur la direction serait stockée dans chaque nœud. Adams et McKenzie (1989) trouvent une trajectoire de coût minimum entre deux points dans une grille à deux dimensions en utilisant une fonction de coût non spécifiée. Au point final, le cap de l'avion doit se trouver à l'intérieur d'un intervalle prédéterminé. Schultz et Shaner (2001) traitent le problème du cap en filtrant la trajectoire pour la rendre plus lisse. Cabos (2013) génère plusieurs trajectoires et élimine par la suite celles qui ne respectent pas les contraintes de l'espace aérien (zones interdites, NOTAM, etc.).

En outre, les coûts pris en compte varient selon les méthodes. Myers (2000) utilise un algorithme de programmation dynamique dans un réseau où le coût des segments est une combinaison linéaire des coûts du temps, du carburant et du risque. Schultz et al. (2003) utilisent une fonction du même type qui inclut aussi les coûts de survol de zones tarifées. Soler et al. (2011) calculent précisément ces coûts, sans considérer d'autres coûts importants comme le coût du temps. Verlut et Lohmiller (2010) proposent d'ajouter au coût des segments une pénalisation pour les manœuvres qui peuvent causer un inconfort en cabine, un élément dont se soucie peu le reste de la littérature. Zhang et al. (2011) associent des coûts à la météo, contrairement à Soler et al. (2011) qui l'intègrent dans le modèle de l'avion. La pénalisation associée à la météo n'est pas précisée. Cabos (2013) évite d'assigner un coût à des éléments comme le temps en proposant plusieurs trajectoires parmi lesquelles le décideur choisit.

2.3 Méthodes de mise à jour

La littérature compte quelques travaux qui proposent des méthodes de mise à jour de trajectoire. La mise à jour peut porter plusieurs noms : réoptimisation, réparation (*repairing*) ou encore replanification (*replanning*). D'abord, les mises à jour de trajectoire d'aéronefs seront présentées. Ensuite, deux autres applications seront introduites.

La mise à jour du plan de vol diffère du problème de l'optimisation de plan de vol puisque des contraintes supplémentaires sont ajoutées. La contrainte principale est évidemment la quantité de carburant

Une technique proposée par Coulmeau (2005) repose sur le maintien d'un excédent de carburant choisi par le pilote. En soustrayant au carburant restant les quantités nécessaires pour atteindre la destination et respecter la réglementation, il devrait toujours rester au moins cet excédent. Si ce n'est pas le cas, la technique révisant chacune des phases du vol en commençant par celle qui présente le plus grand potentiel de réduction de consommation de carburant. La consommation prévue pour chaque phase est réduite jusqu'à ce que l'excédent soit au moins aussi élevé que ce que le pilote avait choisi.

Coulmeau et al. (2007) proposent de se baser sur les concepts entourant le CI pour faire la mise à jour du vol. Les systèmes de gestion de vol actuels ne permettent pas de considérer les solutions pour lesquelles l'heure d'arrivée est à l'extérieur du créneau horaire prévu. Effectivement, les systèmes ne peuvent gérer que des CI fournis par l'exploitant. Ces CI sont calculés à partir d'une estimation de l'heure d'arrivée optimale (ETA_{opt}) et d'une estimation de la quantité optimale de carburant restant à l'arrivée ($EFOB_{opt}$). Coulmeau et al. (2007) proposent plutôt d'asservir ETA et EFOB pour qu'ils se rapprochent le plus de leurs valeurs optimales. Contraire-

ment au CI, les valeurs courantes d'ETA et EFOB sont connues par le système de navigation de l'avion.

La première technique se concentre sur la consommation de carburant, sans prendre en compte le coût du délai. Dans les deux cas, il n'y a pas de garantie que le plan de vol le plus économique soit choisi.

L'invention de Tang et Mukherjee (2000) est aussi valide pour la mise à jour du plan de vol. Ils utilisent un algorithme de programmation dynamique vers l'avant alors que la trajectoire initiale est calculée à rebours.

Wzorek et al. (2010) choisissent l'apprentissage machine pour mettre à jour des trajectoires d'hélicoptère sans pilote. L'apprentissage est tiré de l'environnement et permet de choisir une stratégie pour mettre à jour la trajectoire puis de choisir les paramètres les plus appropriés pour cette stratégie. Ces deux étapes combinées visent à fournir la meilleure mise à jour possible en fonction du temps disponible. Les deux stratégies proposées sont les *Rapidly-exploring Random Trees* (RRT) pour leur rapidité et les *Probabilistic Roadmaps* (PRM) pour la qualité des solutions.

Une des applications où les mises à jour de trajectoires sont fréquentes est celle d'un robot se déplaçant dans sur un terrain inconnu. En se déplaçant, le robot acquiert de nouvelles informations à propos de son environnement et réoptimise la portion restante de la trajectoire. La réoptimisation doit être rapide puisque pendant le calcul le robot doit arrêter ou poursuivre son chemin sur une trajectoire non optimale. Koenig et Likhachev (2002) proposent une version améliorée de l'algorithme de Stentz (1994, 1995) qui dérive de l'heuristique A*. La force de leur algorithme est de minimiser l'espace qui doit être exploré de nouveau après l'apparition d'un obstacle. Ils calculent la mise à jour dans le sens inverse du sens du calcul initial, tout comme Tang et Mukherjee (2000). Ferguson et al. (2006) ajoutent une dimension probabiliste au travail de Koenig et Likhachev (2002) et proposent un algorithme qui répare des RRT. Un algorithme de RRT recherche un chemin dans un réseau. La forme des chemins explorés ressemble à un arbre dont la base est le départ et l'une des feuilles est l'arrivée. Lorsqu'un obstacle se présente, Ferguson et al. (2006) retirent toutes les branches de l'arbre invalidées par cet obstacle. Ils construisent une nouvelle solution à partir des branches restantes de l'arbre.

Les mises à jour sont aussi utilisées pour des applications autres que la génération de trajectoires. Desrochers et Soumis (1985b) s'intéressent à la construction de routes et d'horaires pour les véhicules. Ils modélisent le problème comme un réseau dans lequel chaque nœud ne peut être visité que dans un intervalle de temps fixé. Ils recherchent dans ce réseau le plus court chemin qui respecte les contraintes de temps. Ils utilisent une approche de programmation dynamique (voir algorithme 2.5 pour une idée de la stratégie) pour la résolution. Si une perturbation invalide une solution initiale, ils réoptimisent le chemin. La solution initiale est réutilisée dans le processus de réoptimisation pour créer des bornes à chaque nœud. L'ensemble des chemins de la solution initiale est utilisé comme une borne inférieure. Cet ensemble représente le meilleur cas : celui où la perturbation n'engendre pas de coût supplémentaire. La borne supérieure est représentée par l'ensemble des chemins de la solution initiale duquel ont été retirés tous les chemins perturbés. Celui-ci représente le pire cas : celui où la perturbation inflige un coût très élevé aux arcs qu'elle affecte.

2.4 Problème de plus court chemin avec contraintes

Étant donné que la programmation dynamique semble la méthode la plus appropriée, c'est celle qui sera adoptée. Pour s'assurer que la trajectoire ne requiert pas une consommation de

carburant supérieure à la quantité transportée par l'avion, une contrainte de ressource doit être imposée à la trajectoire. Ce type particulier de problème de plus court chemin est connu dans la littérature sous le nom de *Resource Constrained Shortest Path* ou problème de plus court chemin avec contraintes (RCSP). Certains auteurs utilisent aussi l'appellation *Constrained Shortest Path* (CSP) (Santos et al., 2007; Lozano et Medaglia, 2013) ou *Shortest Path Problem with Resource Constraint* (SPPRC) (Desrochers, 1988; Irnich et Desaulniers, 2004).

Définissons un graphe $G = (V, E)$ formé des sous-ensembles non vides de nœuds et arcs, respectivement. Dans ce graphe, dont un exemple est représenté à la figure 2.1, un chemin est formé d'une séquence de nœuds

Deux éléments de ce graphique méritent mention. D'abord, remarquons que tous les chemins de ce graphe qui relient s à t comporteront six segments. Ce graphe représente donc un problème de décisions séquentielles, c'est-à-dire où les décisions sont prises par étapes. Par exemple, tous les nœuds se trouvant sur la ligne pointillée font partie de l'étape k . Ensuite, ce graphe est un graphe orienté acyclique (*Directed Acyclic Graph*) (DAG), c'est-à-dire qu'aucune combinaison d'arcs ne permet de créer un cycle.

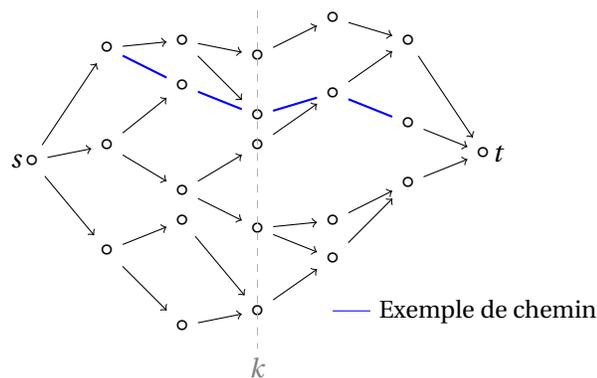


FIGURE 2.1: Exemple de graphe $G = (V, E)$

Le RCSP est un problème de plus court chemin dans lequel au moins une ressource est disponible en quantité limitée pour parcourir le chemin. Irnich et Desaulniers (2004) proposent une formulation générale présentée à la section 2.4.1. La résolution du problème de plus court chemin avec contraintes requiert plus de mémoire et de calculs que la résolution d'un problème de plus court chemin (Bertsekas, 2005), car chaque ressource ajoutée multiplie la taille maximale du problème.

L'une des approches de solutions au problème de RCSP est de lister l'ensemble des chemins possibles et de choisir celui dont le coût est le plus faible. Cette approche correspond à celle qu'a proposée Joksch (1966). Joksch (1966) pense que le calcul n'en vaut la peine que si l'on recherche un ensemble de solutions. Bertsekas (2005) considère que la résolution est encore très coûteuse aujourd'hui. Certains auteurs (Bertsekas, 1998; Dumitrescu et Boland, 2003; Muhandirange et Boland, 2009; Lozano et Medaglia, 2013) proposent des solutions pour accélérer la résolution. Handler et Zang (1980) considéraient que la technique était coûteuse en temps et ont alors proposé une relaxation du problème, qui a été reprise par plusieurs auteurs, notamment Beasley et Christofides (1989); Santos et al. (2007), qui sera présentée en deuxième lieu. Pour une revue complète des méthodes exactes, voir Di Puglia Pugliese et Guerriero (2013).

2.4.1 Formulation de programmation dynamique

Joksch (1966) élabore l'approche de programmation dynamique pour le problème de plus court chemin contraint. Irnich et Desaulniers (2004) proposent la formulation présentée dans cette section. La fin de la section est consacrée aux méthodes d'élagage du réseau qui montrent un potentiel pour la réduction du temps de calcul.

Modèle mathématique

Introduisons d'abord les entités nécessaires à la résolution du RCSP. Pour respecter les contraintes en chaque nœud, il est nécessaire de connaître la consommation de chacune des ressources $m = 1, \dots, M$ pour chaque nœud. Définissons un vecteur $R_i(C)$ décrivant la consommation de ressources au nœud i faisant partie d'un chemin C quelconque. $R_i(C)$ est nommé une étiquette au nœud i .

$$R_i(C) = [R_i^1(C), R_i^2(C), \dots, R_i^M(C)] \quad (2.3)$$

Par exemple, un chemin C pourrait consommer trois dollars, deux litres de carburant et cinq minutes. On aurait alors $R_i(C) = [3, 2, 5]$. Considérons que la ressource $m = 1$ est toujours le coût pour passer d'un nœud à un autre.

Définissons les ressources comme des quantités qui varient d'un nœud à l'autre selon une fonction f , nommée fonction d'extension de ressource. La variation de la quantité de l'ensemble des ressources entre les nœuds i et j est décrite le vecteur $f_{ij}(R_i(C))$.

$$f_{ij}(R_i(C)) = [f_{ij}^1(R_i(C)), \dots, f_{ij}^M(R_i(C))] \quad (2.4)$$

La variation de la quantité de la ressource m entre les nœuds i et j dans un problème de plus court chemin avec contraintes classique est décrite en (2.5).

$$R_j^m(C) = f_{ij}^m(R_i(C)) = R_i^m(C) + r^m(i, j) \quad (2.5)$$

La donnée $r^m(i, j)$ décrit la consommation de ressource m pour se déplacer de i à j . Ainsi, si la ressource 2 est le temps, on consommera $r^2(3, 4)$ secondes pour passer du nœud 3 au nœud 4.

Ce travail se concentre sur des problèmes pour lesquels

$$R_j^m(C) = f_{ij}^m(R_i(C)) = R_i^m(C) \blacksquare r^m(i, j) \quad (2.6)$$

où \blacksquare est une fonction non décroissante qui assure que $R_i^m(C) \leq R_j^m(C)$. La fonction 2.5 en est un exemple.

Tous les problèmes n'utilisent pas que des fonctions non décroissantes. Par exemple, on pourrait vouloir ne choisir un chemin C que si la probabilité de le traverser en sécurité atteint un seuil minimal. Si pour chaque arc (i, j) on définit la ressource m comme la probabilité de traverser le chemin en sécurité, alors on calcule l'état de la ressource comme en (2.7).

$$R_j^m(C) = f_{ij}^m(R_i(C)) = R_i^m(C) \cdot r^m(i, j) \quad (2.7)$$

Notons cependant que si la ressource ne varie pas selon la forme présentée en (2.6), alors l'hypothèse de monotonie¹ de la programmation dynamique n'est pas nécessairement respectée (voir Reinhardt et Pisinger (2011) pour plus de détails sur ce type de fonctions).

1. L'hypothèse de monotonie garantit que pour un plus court chemin C , tous les sous-chemins de C sont plus courts.

L'objectif du problème de plus court chemin avec contraintes est de trouver un chemin C qui commence au noeud s et se termine au noeud t et qui respecte les contraintes tout en minimisant le coût total. On pourrait formuler cet objectif comme en (2.8), c'est-à-dire qu'on tenterait de minimiser la somme du coût de tous les arcs du chemin grâce à une fonction-objectif.

$$\sum_{(i,j) \in C} r^1(i,j) \quad (2.8)$$

Cette formulation imposerait de tenir à la fois compte des contraintes et de la minimisation du coût. Or, plutôt que de minimiser le coût, on peut transformer la minimisation en contrainte. Elle requiert que la somme des coûts de tous les arcs soit inférieure ou égale à L^* , le coût du chemin optimal (qui est habituellement inconnu). Le problème est alors similaire à un problème multiobjectif et peut être résolu comme tel. En d'autres mots, on considère que le coût est une ressource qui peut être consommée. Le coût ne sera utilisé comme critère pour choisir un chemin que parmi les chemins qui relieront s et t .

$$0 \leq \sum_{(i,j) \in C} r^1(i,j) \leq L^* \quad (2.9)$$

La contrainte de ressource est intégrée à l'aide d'un intervalle. À chaque noeud i est associé un intervalle $[a_i^m, b_i^m]$ dans lequel doit être comprise la quantité R_i^m de la ressource m . L'ensemble des intervalles pour un noeud i est donc décrit par $[a_i, b_i]$.

$$[a_i, b_i] = [(a_i^1, b_i^1), \dots, (a_i^M, b_i^M)] \quad (2.10)$$

L'équation (2.11) présente la forme générale des contraintes du problème.

$$a_i \leq R_i(C) \leq b_i, \quad \forall i \in C \quad (2.11)$$

On attribue la qualité *faisable* à un chemin pour lequel $R_i(C)$ respecte la contrainte (2.11).

Définissons pour chaque noeud l'ensemble $\mathcal{F}(i, t)$ des chemins faisables pour passer du noeud i au noeud de destination t (voir (2.12)). Soit le noeud x_k , un noeud de l'état k . Le calcul de l'ensemble $\mathcal{F}(x_k, t)$, nécessite de connaître l'ensemble $\mathcal{F}(x_{k+1}, t)$ pour chacun des états de l'étape $k + 1$.

$$\mathcal{F}(i, t) = \{C_1, C_2, C_3, \dots\} \quad (2.12)$$

Jusqu'à maintenant, le problème a été traité comme un problème multiobjectif. S'il en était un, le décideur serait appelé à choisir un chemin parmi les chemins de l'ensemble $\mathcal{F}(s, t)$ pour choisir celui qui représente le meilleur compromis à ses yeux. Or, nous connaissons déjà sa préférence : le décideur désire le chemin le moins cher. Ainsi, parmi les chemins de l'ensemble $\mathcal{F}(s, t)$, on choisit celui pour lequel la valeur de $R_s^1(C)$ est la plus faible. En d'autres mots, on choisit :

$$\min_{C \in \mathcal{F}(s,t)} \{R_s^1(C)\} \quad (2.13)$$

La programmation dynamique repose sur le principe d'optimalité. Soit un chemin C qui relie s à t et soit l'arc (i, t) , le dernier arc de C . Selon le principe d'optimalité, si C est optimal, alors, le sous-chemin de C reliant s à i est le chemin le plus court reliant s à i . Le principe d'optimalité doit être modifié pour pouvoir être utilisé dans le cas du plus court chemin avec contraintes (Desrosiers et al., 1983).

Soit la contrainte $[a_t, b_t]$ et l'ensemble de chemins \mathcal{C} arrivant en t . La consommation de ressource de chaque chemin $C \in \mathcal{C}$ se situe dans l'intervalle $[a_t, b_t]$. Si le chemin C^* minimise le coût et si l'arc (i, t) est le dernier arc de C^* , alors le sous-chemin de s à i est le chemin à coût minimum parmi les chemins arrivant en i qui respectent $R_i^m(C) \leq b_i^m - r^m(i, j)$.

Élagage du réseau

Étant donné la quantité importante de calculs que requiert le RCSP, une attention particulière est portée à réduire l'espace de solutions. On élimine dès qu'on les rencontre les solutions qui ne respectent pas certains critères pour accélérer les calculs. Ce processus s'appellera élagage. Bertsekas (2005) propose d'élaguer l'arbre de solutions en éliminant le plus tôt possible les solutions infaisables et les solutions inférieures. Lozano et Medaglia (2013) ajoutent l'élagage par bornes. Ces types d'élagage seront présentés dans les paragraphes suivants.

Les concepts de dominance et de non infériorité, présentés dans les deux définitions suivantes, sont des concepts-clés de l'élagage.

Définition 2.1 (Dominance). Soit un chemin C et un chemin C' . C domine C' si $R_i^m(C) \leq R_i^m(C'), \forall m = 1, \dots, M$ et que l'inégalité est stricte pour au moins un m .

Définition 2.2 (Non infériorité). Un chemin C est non inférieur s'il n'existe pas de chemin C' qui le domine. Un chemin non inférieur peut aussi être appelé Pareto optimal ou efficace.

Élagage par dominance De manière générale, l'élagage par dominance vise à éliminer les solutions inutiles (Irnich et Desaulniers, 2004). C'est un incontournable de tous les algorithmes de plus court chemin avec contraintes. C'est la forme d'élagage la plus simple dans laquelle on ne conserve que les solutions non inférieures.

Considérons un problème à une ressource, le temps, dont l'objectif est de minimiser le coût. La figure 2.2 présente un ensemble d'étiquettes (\bullet) associées à différents chemins au nœud i . Chaque étiquette contient donc R_i^1 , le coût du chemin C , et R_i^2 , le temps requis pour le parcourir. Le temps de parcours doit se situer dans l'intervalle $[a_i^2, b_i^2]$. Le chemin C_1 est le moins coûteux, mais il est aussi le plus long. Inversement, le chemin C_5 est coûteux, mais demande beaucoup moins de temps. Chaque étiquette sur la figure est prolongée par un trait. Celui-ci indique quel chemin suivre si on dispose de plus de ressource que nécessaire pour les chemins indentifiés. Par exemple, si on disposait de la quantité de temps b_i^2 , il vaudrait mieux parcourir le chemin C_1 .

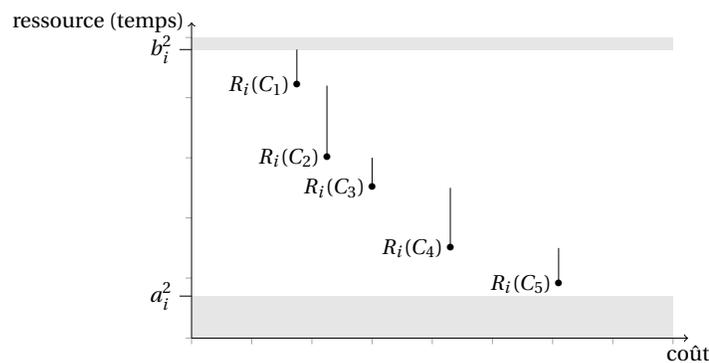


FIGURE 2.2: Exemple de liste d'étiquettes au nœud i (Source : Desrosiers et al. (1983))

L'ensemble représenté à la figure 2.2 ne contient que des étiquettes non inférieures. Celles-ci sont classées en ordre strictement décroissant de coût, ce qui fait en sorte que la consommation de la ressource est strictement croissante. Cette propriété permet de fusionner deux listes d'étiquettes non inférieures en $m + n$ opérations, où m et n sont les longueurs des deux ensembles d'étiquettes à fusionner.

Considérons les deux ensembles d'étiquettes à la figure 2.3. Supposons que la consommation de ressource doit se trouver dans l'intervalle $[0, 9]$. Considérons différentes situations en utilisant la nomenclature de Desrosiers et al. (1983). L'étiquette A est *sans effet* puisque la consommation de ressource (10) est à l'extérieur de l'intervalle. L'étiquette B est *insérée* entre les étiquettes 1 et 2 puisqu'elle ne domine ni 1, ni 2 et que ni 1, ni 2 ne dominant B. L'étiquette C est *éliminée*, car elle est dominée par l'étiquette 2.

#	coût	ressource	#	coût	ressource
1	3	7	A	1	10
2	8	4	B	6	6
			C	9	5

FIGURE 2.3: Deux listes d'étiquettes

Lozano et Medaglia (2013) observent que de conserver toutes les solutions non inférieures à chaque nœud pourrait consommer une quantité très importante d'espace de stockage. Ils proposent de déterminer pour chaque nœud une quantité maximale Q de chemins partiels pouvant être stockés. En utilisant $Q = 3$, ils utilisent deux règles pour remplacer les chemins déjà stockés. D'abord, toujours conserver le chemin partiel qui consomme le moins de ressource et celui qui coûte le moins cher (*règle élitiste*). Ensuite, remplacer de manière aléatoire un chemin non dominé (*règle aléatoire*). Notons toutefois que cette méthode permet de trouver des solutions exactes, mais que l'utilisation de règle aléatoire fait en sorte que l'optimalité n'est jamais garantie.

Élagage par bornes L'élagage par borne se trouve dans de nombreux algorithmes (Minoux, 1975; Dumitrescu et Boland, 2003; Bertsekas, 1998, 2005; Zhu et Wilhelm, 2012; Lozano et Medaglia, 2013). On l'appelle aussi propagation des contraintes. Lorsque l'élagage se fait sur la ressource, il peut être appelé « élagage par infaisabilité ». Pour débiter, la contrainte doit être connue. La quantité maximale de ressource est habituellement connue. Comme coût maximal, Lozano et Medaglia (2013); Bertsekas (1998) utilisent le coût d'un chemin non optimal mais faisable. Pour l'élagage en tant que tel, le principe est le même, que l'élagage se fasse sur une ressource ou le coût. On cherche à éliminer le plus rapidement possible les chemins infaisables ou trop coûteux, c'est-à-dire qui violent une contrainte. La stratégie consiste à vérifier s'il est possible que le chemin qu'on construit respecte la contrainte à l'arrivée, sachant la dépense déjà engagée pour parcourir le chemin de s vers i . Dans le cas où la fonction faisant varier la ressource est de la forme vue en (2.5), on peut formuler la vérification par le critère (2.14).

$$\tilde{J}^m(i) + R_i^m(C) \leq b_i^m, \quad m = 2, \dots, M \quad (2.14)$$

Pour chaque état et les ressources $m = 1, \dots, M$, on calcule la consommation minimale $\tilde{J}^m(i)$ de ressource pour passer de l'état i à l'état t , l'arrivée, sans tenir compte des autres contraintes. C'est l'équivalent de déterminer la solution du problème du plus court chemin dans le sens inverse du calcul de la trajectoire. Si la valeur de $\tilde{J}^m(i)$ dépasse la quantité de ressource m disponible, le nœud i est supprimé (Dumitrescu et Boland, 2003). Pour Minoux (1975), le gain en

temps permis par l'élagage par infaisabilité est « appréciable ». Cet élagage est agressif et est particulièrement avantageux lorsque les contraintes sur les ressources sont très serrées (Lozano et Medaglia, 2013). Pour que cette méthode soit utile, le temps requis pour déterminer les $\tilde{J}^m(i)$ doit être inférieur au gain de temps que l'élagage par infaisabilité permet, ce qui pourrait ne pas être le cas.

2.4.2 Relaxation lagrangienne

La relaxation lagrangienne a été introduite par Handler et Zang (1980) qui considéraient que la formulation de la section précédente demandait trop de calculs et de temps. Leur travail a été repris et amélioré par Beasley et Christofides (1989); Jüttner et al. (2001); Santos et al. (2007); Carlyle et al. (2008); Di Puglia Pugliese et Guerriero (2013); Lozano et Medaglia (2013) Le principe de base est de pénaliser la consommation de ressource plutôt que d'imposer des contraintes fermes afin de réduire le temps de calcul.

En d'autres mots, la relaxation lagrangienne consiste à résoudre un problème de plus court chemin pour lequel le coût d'un arc entre i et j , qui était initialement $r^1(i, j)$, devient

$$E(i, j) = r^1(i, j) + (t_2 - s_2) \cdot r^2(i, j) + \dots + (t_M - s_M) \cdot r^M(i, j) \quad (2.15)$$

où $t_2, s_2, \dots, t_M, s_M$ sont des coefficients non négatifs à optimiser. Le coût du chemin le plus court pour un ensemble de coefficients est donné par L . Par une méthode itérative, on fait varier les coefficients pour trouver $t_2^*, s_2^*, \dots, t_M^*, s_M^*$ qui maximisent L .

La relaxation lagrangienne donne une solution optimiste, car elle est le dual du problème original de plus court chemin avec contraintes. Or, la solution du problème dual garantit seulement des bornes inférieure et supérieure pour le coût de la solution au problème primal. La solution du problème dual peut être une solution infaisable, mais elle peut être utilisée pour trouver une solution faisable.

Cette technique requiert de connaître les coûts de tous les arcs du réseau. La taille du réseau du problème à l'étude est si importante que la relaxation lagrangienne serait excessivement coûteuse à utiliser.

2.5 Algorithmes de résolution

Plusieurs algorithmes ont été développés pour résoudre les problèmes de plus court chemin, qu'ils soient avec ou sans contrainte de ressources. La première section présente un algorithme générique pour le RCSP. Les deux sections suivantes couvrent les grandes classes de méthodes pour résoudre les problèmes de plus court chemin. La classe des algorithmes de résolution par poussée comprend les algorithmes à correction d'étiquettes et les algorithmes à fixation d'étiquettes. La classe des algorithmes de résolution par traction comprend la méthode primal-dual. La dernière section aborde le prétraitement du réseau, c'est-à-dire l'ensemble des opérations que l'on peut faire subir à la grille avant le calcul pour accélérer celui-ci.

2.5.1 Algorithme générique pour le RCSP

Irnich et Desaulniers (2004) proposent un algorithme générique reposant sur l'extension de chemins (approche de poussée ou *reaching* (voir figure 2.4a)) pour la recherche du plus court chemin (algorithme 2.1). L'algorithme repose sur l'utilisation de l'ensemble \mathcal{U} contenant des chemins qui n'ont pas d'extension et de l'ensemble \mathcal{C} des chemins utiles.

En bref, cet algorithme choisit un chemin C dans l'ensemble \mathcal{U} tant que ce dernier n'est pas vide. Ce chemin C peut être étendu en ajoutant un segment. Les chemins étendus faisables sont ajoutés à l'ensemble \mathcal{U} . Après avoir généré des chemins étendus, le chemin C est considéré comme utile et est donc ajouté à l'ensemble \mathcal{C} . L'algorithme de dominance élimine les chemins dominés des ensembles \mathcal{U} et \mathcal{C} . Lorsque l'ensemble \mathcal{U} est vide, le chemin complet minimisant le coût est choisi parmi les chemins se trouvant dans \mathcal{C} .

Algorithme 2.1 Algorithme générique pour le RCSP (Source : Irnich et Desaulniers (2004))

```

1:  $\mathcal{U} \leftarrow t$ 
2:  $\mathcal{C} \leftarrow \emptyset$ 
3: tant que  $\mathcal{U} \neq \emptyset$  faire
4:   Choisir un chemin  $C \in \mathcal{U}$ 
5:    $\mathcal{U} = \mathcal{U} - C$ 
6:   pour tout  $(f_p, i) \in E$  faire
7:     si  $(C, i) \in \mathcal{F}_{t,i}$  alors
8:        $\mathcal{U} = \mathcal{U} \cup (C, i)$ 
9:     fin si
10:  fin pour
11:   $\mathcal{C} = \mathcal{C} \cup C$ 
12:  si condition alors Algorithme de dominance ▷ Pourrait être déplacé
13:  fin si
14: fin tant que
15: Choisir un chemin  $C$  t.q.  $\min_{C \in \mathcal{C}} \{\min_{R_i(C) \in \mathcal{R}(C)} R_s^1(C)\}$ 

```

Cet algorithme peut se décliner de différentes manières. Premièrement, le choix du chemin à étendre à la ligne 4 est libre. Les options incluent un algorithme à fixation d'étiquettes ou un algorithme à correction d'étiquettes qui sont décrits à la section 2.5.2. Deuxièmement, la technique d'extension pourrait être modifiée. L'algorithme 2.1 utilise une méthode de poussée (*reaching* dans laquelle un chemin est choisi, puis étendu. Elle est illustrée à la figure 2.4a, où tous les arcs sortent du même nœud, le dernier nœud du chemin à étendre. Une autre approche, la traction (ou *psulling*), consiste à choisir un nœud et à étendre plusieurs chemins vers celui-ci. Elle a été proposée par Desrochers et Soumis (1985b). Elle est illustrée à la figure 2.4b où les arcs se terminent tous au même nœud. Desrochers (1988) compare l'approche de poussée à l'approche de traction. Bien que la complexité mathématique des deux approches soit la même, l'approche de traction peut permettre d'obtenir des temps de calcul moindre que l'approche de poussée (Desrochers, 1988).

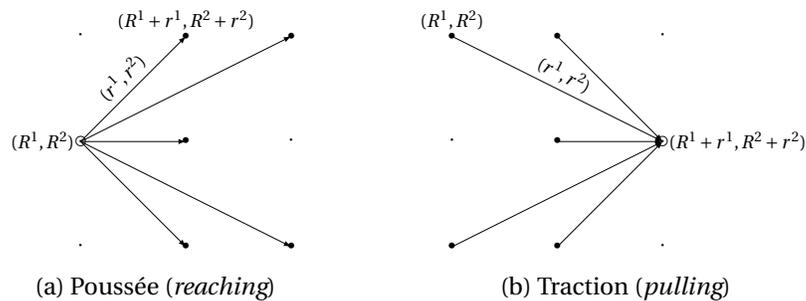


FIGURE 2.4: Deux techniques pour étendre les chemins

2.5.2 Algorithmes de résolution par poussée

La littérature distingue deux types d'approche pour les algorithmes de résolution de plus court chemin, soient les algorithmes à fixation d'étiquettes et à correction d'étiquettes. Les algorithmes à fixation d'étiquette déterminent une étiquette optimale à chaque itération. Ils sont valides pour les problèmes de plus court chemin dans les réseaux acycliques ou dans un réseau où les coûts des arcs sont tous non négatifs. Les algorithmes à correction d'étiquette ne considèrent les étiquettes comme optimales qu'une fois rendus à l'itération finale (Ahuja et al., 1993). Cette classe d'algorithme est considérée comme moins efficace en théorie, cependant elle est applicable à toutes les classes de problèmes (Ahuja et al., 1993; Bertsekas, 1998).

En pratique, la variation de performance est plus nuancée. Les algorithmes à correction d'étiquettes permettent de tirer avantage d'une initialisation plus sophistiquée des nœuds et de la structure des graphes acycliques. Ils sont aussi plus avantageux pour solutionner les graphes dont le nombre d'arrêtes est faible par rapport au nombre de nœuds ($|E| \ll |V|^2$). Au contraire, si le graphe est dense ($|E| \approx |V|^2$), alors les algorithmes à fixation d'étiquette sont plus appropriés. Notons aussi que ceux-ci permettent de tirer un avantage significatif des problèmes n'ayant qu'un nœud de destination en réduisant significativement le temps de calcul (Bertsekas, 1998). Dans le cadre d'un plus court chemin avec contraintes, Irnich et Desaulniers (2004) considèrent qu'un graphe acyclique mène naturellement à l'utilisation d'un algorithme à fixation d'étiquettes. Pour des problèmes multiobjectifs, Santos et Paixa (2013) obtiennent le plus souvent les meilleurs résultats avec un algorithme à correction d'étiquettes. Pour les plus grands réseaux, les meilleurs résultats sont toujours obtenus avec un algorithme à correction d'étiquettes. Joch (1966); Desrosiers et al. (1983) utilisent un algorithme à correction d'étiquettes alors que Aneja et al. (1983) utilisent un algorithme à fixation d'étiquettes.

Dans tous les cas, Ahuja et al. (1993); Bertsekas (1998) mettent l'accent sur l'importance de choisir la bonne structure de données pour le problème, ce choix étant critique pour la bonne performance de l'algorithme.

Pour la revue de quelques algorithmes, nous considérons toujours le graphe $G = (V, E)$ tel que décrit au début de la section 2.4.

Algorithmes à correction d'étiquettes

En 1956, Bellman proposait un algorithme à correction d'étiquettes pour solutionner le problème de plus court chemin (voir algorithme 2.2). L'algorithme de Bellman-Ford résout le cas général du plus court chemin à un nœud de départ. Dans une boucle qui se répète au plus $|V| - 1$ fois, l'algorithme parcourt toutes les arrêtes du graphe. Pour chaque nœud, si le coût cumulé calculé est inférieur au coût cumulé actuel, ce dernier est remplacé.

Bertsekas (1998) propose de remplacer la condition de la ligne 11 de l'algorithme 2.2 par

$$c_i + x_{ij} < \min\{c_j, c_s - \tilde{J}_j^1\} \quad (2.16)$$

En somme, si on connaît le coût minimal pour se rendre de j à s , on peut initialiser les nœuds (ligne 2) avec une valeur plus faible que l'infini dans l'espoir de ne pas insérer j dans l'ensemble à visiter S à la ligne 11.

Desrosiers et al. (1983) diminuent le temps de calcul dans un graphe acyclique en ordonnant les nœuds de telle sorte que

$$(i, j) \in E, \text{ alors } i < j \quad (2.17)$$

Algorithme 2.2 Bellman-Ford

```

1: pour tout  $i \in V$  faire
2:    $c_i \leftarrow \infty$ 
3:    $p_i \leftarrow \text{NIL}$ 
4: fin pour
5:  $c_t \leftarrow 0$ 
6: pour  $i = \{1, 2, \dots, |V|\}$  faire
7:   pour tout  $(i, j) \in E$  faire
8:     si  $c_i + x_{ij} < c_j$  alors
9:        $c_j \leftarrow c_i + x_{ij}$ 
10:       $p_j \leftarrow i$ 
11:       $S = S \cup j$ 
12:     fin si
13:   fin pour
14: fin pour

```

On dira que les nœuds sont classés topologiquement. De cette manière, à la ligne 7 de l'algorithme, plutôt que de visiter tous les arcs du réseau, on ne visite que les arcs sortant du nœud visité à cette itération. L'astuce permet de ne visiter chaque nœud qu'une fois.

Algorithmes à fixation d'étiquettes

Dijkstra (1959) proposait un algorithme à fixation d'étiquettes (voir algorithme 2.3). La stratégie consiste à parcourir les sommets en choisissant toujours celui qui possède le coût cumulé le plus faible d'abord (cette priorisation est un ajout de Fredman et Tarjan (1984)). Notons que cet algorithme requiert que les coûts de tous les arcs soient non négatifs.

Les nœuds sont répartis dans deux ensemble : l'ensemble des nœuds déjà visités (S) et l'ensemble de tous les autres nœuds (\bar{S}). Le coût cumulé pour les nœuds de S est permanent alors que celui des nœuds de \bar{S} est temporaire. L'algorithme sélectionne dans \bar{S} le nœud i qui possède le coût cumulé le plus faible et l'ajoute à S . Pour chaque arc sortant de i , un nœud est visité et son coût cumulé peut être modifié. Ces opérations sont répétées tant qu'il reste des nœuds à visiter ou que le nœud de destination n'a pas été visité. La sélection des nœuds peut ralentir l'exécution de l'algorithme de Dijkstra. C'est ce qui fait en sorte qu'il n'est pas nécessairement plus rapide qu'un algorithme à fixation d'étiquettes.

Hart et al. (1968) proposent une extension de l'algorithme de Dijkstra qui permet d'exploiter de l'information à propos de la destination, l'algorithme A^* (algorithme 2.4). Pour la sélection du nœud, ils utilisent un estimé du coût du chemin complet plutôt que le coût du chemin déjà parcouru. Au nœud i , l'estimé est noté comme \tilde{c}_i . Cet estimé est la somme du coût du chemin parcouru jusqu'à i et du coût estimé du chemin entre i et la destination. La différence entre l'algorithme 2.4 et l'algorithme 2.3 ne tient qu'à quatre lignes : la ligne 3 a été ajoutée pour initialiser la valeur de l'estimé pour chaque nœud, les lignes 7 et 17 ont été ajoutés pour calculer l'estimé et la ligne 11 a été modifiée pour utiliser \tilde{c}_i plutôt que c_i .

L'une des propriétés de l'algorithme de Dijkstra est que les coûts cumulés permanents sont non décroissants. Ainsi, en classant ces coûts cumulés en ordre croissant, on s'assure de ne jamais revenir sur ses pas. Dial (1969) propose de créer un certain nombre d'ensembles dans lesquels classer les coûts cumulés (en supposant que les coûts des arcs sont entiers). Pour ré-

Algorithme 2.3 Dijkstra

```

1: pour tout  $i \in V$  faire
2:    $c_i \leftarrow \infty$ 
3:    $p_i \leftarrow \text{NIL}$ 
4: fin pour
5:  $c_t \leftarrow 0$ 
6:  $S \leftarrow \emptyset$ 
7:  $\bar{S} \leftarrow E$ 
8: tant que  $\bar{S} \neq \emptyset$  OU  $s \notin S$  faire
9:    $i \leftarrow \min_{j \in \bar{S}} c_j$ 
10:   $S := S \cup i$ 
11:   $\bar{S} := \bar{S} - i$ 
12:  pour tout  $(i, j) \in E$  faire
13:    si  $c_j > c_i + x_{ij}$  alors
14:       $c_j \leftarrow c_i + x_{ij}$ 
15:       $p_j \leftarrow i$ 
16:    fin si
17:  fin pour
18: fin tant que

```

duire l'espace de stockage nécessaire pour pouvoir stocker tous les coûts cumulés possibles, il propose d'utiliser $K = \max_{x_{ij} | (i,j) \in E}$ et de les parcourir de manière circulaire (voir figure 2.5).

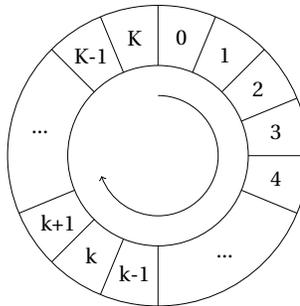


FIGURE 2.5: Structure de stockage pour l'implémentation de Dial de l'algorithme de Dijkstra (Source : Ahuja et al. (1993))

2.5.3 Algorithme de résolution par traction

Desrochers (1988) propose une généralisation de la méthode initialement présentée dans Desrochers et Soumis (1985b), qui est une version modifiée de l'algorithme de Dijkstra. Cette méthode est nommée primal-dual (voir algorithme 2.5). À chaque nœud sont associés deux ensembles d'étiquettes P_i et Q_i , qui représentent les bornes supérieure et inférieure, respectivement, des étiquettes non inférieures du nœud i . La figure 2.6a montre les ensembles Q_j et P_j à l'initialisation. L'ensemble Q_j envisage le scénario le plus optimiste : un chemin à coût $-\infty$. L'ensemble P_j montre qu'aucun chemin ne permet relier le nœud j au nœud de départ en respectant les contraintes, car le coût est infini. Lorsque P_i et Q_i ont la même valeur, les étiquettes non inférieures de i sont connues. Si P_i et Q_i ont des valeurs différentes, c'est qu'il existe une

Algorithme 2.4 A*

```

1: pour tout  $i \in V$  faire
2:    $c_i \leftarrow \infty$ 
3:    $\tilde{c}_i \leftarrow \infty$ 
4:    $p_i \leftarrow \text{NIL}$ 
5: fin pour
6:  $c_t \leftarrow 0$ 
7:  $\tilde{c}_t \leftarrow c_t +$  estimation heuristique du coût de  $t$  jusqu'à  $s$ 
8:  $S \leftarrow \emptyset$ 
9:  $\bar{S} \leftarrow E$ 
10: tant que  $\bar{S} \neq \emptyset$  OU  $s \notin S$  faire
11:    $i \leftarrow \min_{j \in \bar{S}} \tilde{c}_j$ 
12:    $S := S \cup i$ 
13:    $\bar{S} := \bar{S} - i$ 
14:   pour tout  $(i, j) \in E$  faire
15:     si  $c_j > c_i + x_{ij}$  alors
16:        $c_j \leftarrow c_i + x_{ij}$ 
17:        $\tilde{c}_j \leftarrow c_j +$  estimation heuristique du coût de  $j$  jusqu'à  $s$ 
18:        $p_j \leftarrow i$ 
19:     fin si
20:   fin pour
21: fin tant que

```

zone d'incertitude (voir les zones grises de la figure 2.6). Cette zone est déterminée par l'intervalle $[U_i, b_i]$.

Après l'initialisation, un nœud à traiter est choisi (ligne 9). Le nœud choisi est celui qui est le nœud *lexicographiquement le plus petit* de l'ensemble des nœuds à visiter. La définition 2.3 de l'ordre lexicographique est celle de Desrosiers et al. (1983). Notons que dans cette définition, contrairement à la convention utilisée dans ce texte, le coût est le dernier élément du vecteur.

Définition 2.3. Soit $v = [v^2, \dots, v^M, v^1]$, $w = [w^2, \dots, w^M, w^1] \in \mathfrak{R}^M$. v est dit lexicographiquement plus petit que w ($v < w$) si et seulement si $v \neq w$ et que le premier élément non-nul de $w - v$ est positif. (Desrosiers et al., 1983)

À la figure 2.6b, le nœud est choisi pour la première fois. En appliquant la fonction f_{ij} à tous les prédécesseurs i de j , il est déterminé qu'un chemin est faisable sur tout l'intervalle à un coût de 7 (lignes 11 à 14 de l'algorithme 2.5). Au-delà de U_i , le coût de 7 constitue une borne supérieure. Ce nœud est choisi à nouveau lorsque de l'information sur un chemin qui consomme au moins U_i unités de ressources a été acquise. En utilisant à nouveau la fonction f_{ij} , on détermine qu'il existe un chemin à coût plus faible, mais qui consomme plus de ressource sur la deuxième moitié de l'intervalle. Puisque $b_i = U_i$, c'est donc que $Q_i = P_i$ c'est-à-dire que toutes les étiquettes non inférieures ont été déterminées. Ainsi, ce nœud ne sera plus visité. Tout comme pour l'algorithme de Dijkstra, le choix du nœud à considérer peut être coûteux en temps. Voir Desrochers et Soumis (1985a) pour une solution à ce problème.

2.5.4 Prétraitement du réseau

Pour réduire le temps de calcul, on peut effectuer des traitements sur le réseau qui permettent de réduire sa taille ou de réduire le nombre d'étiquettes à calculer.

Algorithme 2.5 Méthode primal-dual (Source : Desrochers (1988))

```

1: pour tout  $i \in V$  faire
2:    $P_i \leftarrow \emptyset$ 
3:    $Q_i \leftarrow (-\infty, (a_i^1, \dots, a_i^M))$ 
4: fin pour
5:  $P_t \leftarrow (0, 0)$ 
6:  $Q_t \leftarrow (0, 0)$ 
7:  $S \leftarrow \cup_{j \in V} (Q_j - P_j)$ 
8: tant que  $S \neq \emptyset$  faire
9:    $j | F(S) = \min_{x \in S} \{R_j(C)\}$ 
10:   $S = S - (Q_j - P_j)$ 
11:   $LB_j = \text{EFF}\{R_j = (\max\{a_i^m, R_i^m + r^m(i, j), U_i\}, m = 1, \dots, M) | \forall (i, j) \in E, R_i \in Q_i\}$ 
12:   $UB_j = \text{EFF}\{R_j = (\max\{a_i^m, R_i^m + r^m(i, j), U_i\}, m = 1, \dots, M) | \forall (i, j) \in E, R_i \in P_i\}$ 
13:   $P_j \leftarrow P_j \cup (LB_j \cap UB_j)$ 
14:   $Q_j \leftarrow \{Q_j - (Q_j - P_j)\} \cup LB_j$ 
15:   $S = S \cup (Q_j - P_j)$ 
16: fin tant que

```

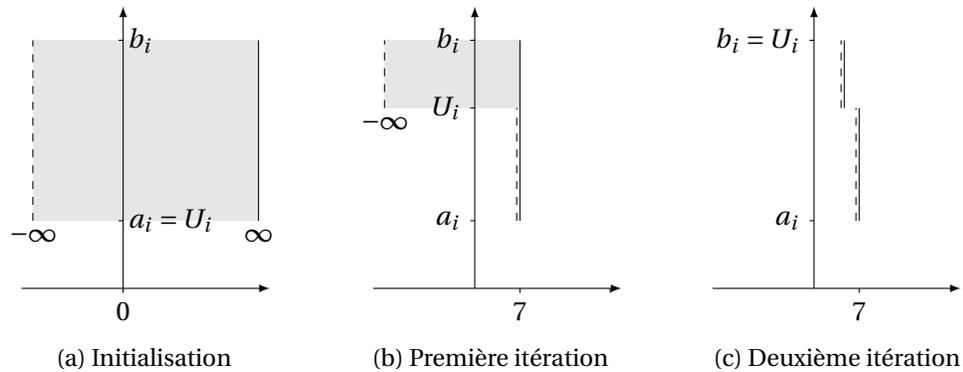


FIGURE 2.6: Mise à jour d'une liste d'étiquettes par traction (Source : Desrochers (1988))

Forme elliptique

Soit une grille, avec un point de départ s et un point d'arrivée t , dans laquelle on veut trouver le chemin le plus court entre s et t . Pour réduire l'espace de solutions, Terrovitis et al. (2005) proposent de déterminer une région elliptique qui contiendra la trajectoire finale à partir d'un chemin non optimal. D'abord, on calcule ce chemin qui doit respecter les contraintes du problème. Supposons que l'algorithme a identifié le chemin $s - p_1 - p_2 - t$, qui est un chemin non optimal de longueur l , tel qu'indiqué à la figure 2.7. On constate que le point p_3 ne fera pas partie de l'espace de solutions réduit, car le chemin le plus court permettant de l'inclure est le chemin $s - p_3 - t$. Or, $|s, p_3| + |p_3, t| > l$, c'est-à-dire que ce chemin est plus long que le chemin $s - p_1 - p_2 - t$. Ensuite, à partir du chemin de longueur l , l'ellipse qui contiendra la trajectoire optimale de longueur l^* est dessinée. Une ellipse est décrite par les points dont la somme des distances avec les foyers est constante. Dans le cas de la figure 2.7, les foyers sont les points s et t et la distance est l . L'espace de solution est réduit à la zone grise. Plus la distance l est courte, plus l'aire de l'ellipse sera réduite, ce qui permet de réduire l'espace de solutions et donc le

temps de calcul. On a donc avantage à déterminer une longueur l qui se rapproche de la longueur l^* .

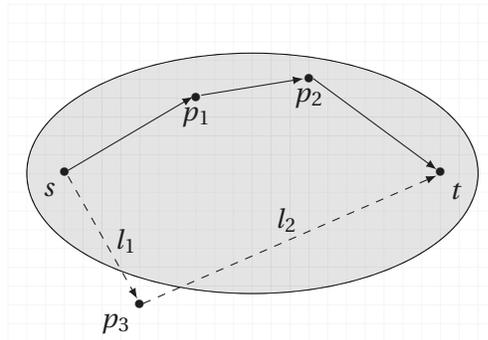


FIGURE 2.7: Réduction de l'espace de solutions à une ellipse

Devulapalli (2012) utilise cette stratégie pour réduire l'espace de solutions dans le calcul du profil horizontal de vol. Tang et Mukherjee (2000) rejettent cette stratégie, car ils considèrent que dans certaines conditions, l'ellipse peut être difficile à construire ou peut contenir un espace défavorable pour le vol. Ils utilisent les points par lesquels passent les plus courts chemins qui tiennent compte du vent pour définir l'espace de solutions réduit.

Élimination des nœuds inatteignables et des arcs inutiles

Aneja et al. (1983) proposent de déterminer le plus court chemin entre s et tous les autres nœuds pour chacune des ressources $m = 2, \dots, M$ en utilisant r^m comme longueur d'arc. Si pour un nœud i la quantité \tilde{J}_i^m dépasse la quantité de ressource m disponible, alors le nœud est éliminé. De plus, ils déterminent la quantité \underline{J}_i^m , soit la quantité minimale de ressource m nécessaire pour se rendre de i à t . Pour chaque nœud i et chaque ressource m , on supprime tous les nœuds pour lesquels $\tilde{J}_i^m + \underline{J}_i^m$ dépasse la quantité totale de ressource m disponible. De plus, ils suppriment aussi tous les arcs (i, j) pour lesquels $\tilde{J}_i^m + r^m(i, j) + \underline{J}_j^m$ dépasse la quantité de ressource m disponible. Beasley et Christofides (1989) ne reprennent que cette dernière étape.

Beasley et Christofides (1989) utilisent la borne supérieure sur le coût calculée par la relaxation lagrangienne pour éliminer les chemins dont le coût est trop élevé. Dumitrescu et Boland (2003) obtiennent de meilleurs résultats que Aneja et al. (1983) et Beasley et Christofides (1989) en répétant le prétraitement tant que des nœuds ou des arcs sont supprimés du réseau. Le prétraitement seul leur permet de résoudre certains problèmes de plus court chemin avec contraintes à 10 ressources et 500 nœuds. Muhandiramge et Boland (2009) utilisent le prétraitement pour améliorer la qualité de la relaxation lagrangienne, qui est recalculée chaque fois que des nœuds ou des arcs sont supprimés.

Chapitre 3

Méthodologie

Ce chapitre décrit la méthode qui est employée pour répondre au problème de recherche. Dans un premier temps, le problème décrit au chapitre 1 est transformé en modèle mathématique. La formulation utilisée est celle d'un plus court chemin avec contraintes de ressource, présentée à la section 2.4. L'idée générale est de construire une trajectoire pour l'avion dans un réseau représentant l'espace aérien. Les chemins sont construits progressivement ajoutant un segment à un chemin existant. Parmi ces chemins, on choisit celui qui coûte le moins cher, mais ne dépense pas plus de carburant que disponible. Dans la deuxième section de la méthodologie, la mise en œuvre du modèle est présentée. Elle permet de préciser dans quel ordre les chemins sont construits et comment ils sont sélectionnés.

3.1 Modélisation du problème

Dans cette section, le modèle est présenté en détail. La première section montre comment l'espace aérien est représenté sous la forme d'un réseau et pourquoi la construction choisie est adéquate. Les autres sous-sections se trouvent dans un ordre classique pour un problème d'optimisation, c'est-à-dire l'ordre suivant : variables, fonction de coût, paramètres puis contraintes. Ces sous-sections permettent de traduire en langage mathématique les éléments décrits verbalement au chapitre 1. La dernière partie est consacrée à la description de la consommation des ressources.

3.1.1 Réseau

Lors du développement de la méthode de trajectoire initiale, un espace de la forme d'un cylindre elliptique a été utilisé. La trajectoire sera toujours contenue dans un cylindre elliptique si les paramètres a et b de l'ellipse (voir figure 3.1) sont choisis adéquatement. La hauteur de ce cylindre sera celle de l'espace aérien, ce qui permettra de considérer toutes les altitudes permises.

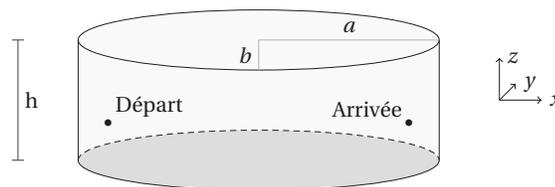


FIGURE 3.1: Cylindre elliptique et ses paramètres

L'idée la plus simple est de considérer un espace de forme rectangulaire (en bleu à la figure 3.2) autour du départ et de l'arrivée (\star sur la figure). Peu importe la forme, les zones hachurées ne seront jamais atteintes par l'avion en croisière. De plus, on sait que la zone ombrée est inatteignable ou qu'un avion dans cette zone ne peut atteindre l'arrivée en raison de limitations physiques. L'hexagone irrégulier (en rouge sur la figure) qui représente la forme d'un réseau dans lequel on considérerait toutes les possibilités est difficile à dessiner. L'ellipse (en noir) couvre environ la même surface utile que l'hexagone et n'a que deux paramètres à ajuster. La théorie présentée à la section 2.5.4 présente une méthode d'ajustement adaptée au problème, puisque des trajectoires déjà utilisées par une compagnie aérienne pourraient être utilisées pour la dessiner. L'ellipse semble donc être une forme adéquate pour la grille.

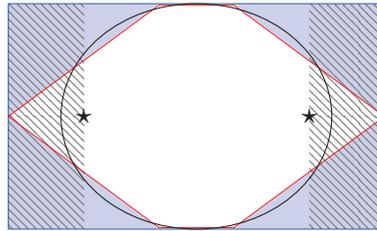


FIGURE 3.2: Comparaison de différentes formes de réseau

Les méthodes d'intégrations numériques à utiliser sont adaptées à un espace discret. L'espace est donc discrétisé, ce qui permettra de construire un graphe $G = (V, E)$. Dans un premier temps, l'ellipse est découpée en tranches identifiées par un indice i (voir figure 3.3). Chacune de ces tranches est ensuite divisée horizontalement et verticalement. Chaque cube ainsi produit est associé à un point (i, j, k) . Ce cube est représenté par un point situé en son centre qui deviendra un nœud du graphe. Pour obtenir une trajectoire lisse, tout comme pour la planification initiale, l'ellipse est orientée de manière à ce que son grand axe relie le départ et l'arrivée de la trajectoire.

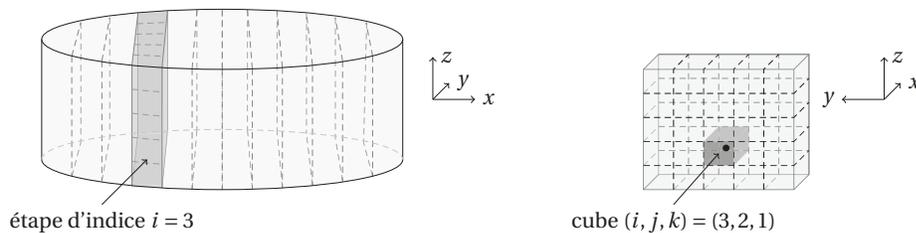


FIGURE 3.3: Illustration du découpage en tranche

Notons que l'on pourrait choisir des nœuds différents. Par exemple, si l'on souhaitait inclure dans le réseau des points de passage obligatoires, on pourrait remplacer un ou plusieurs points du cylindre elliptique par le point de passage désiré. Étant donné que le réseau présenté plus haut est un problème plus général, c'est celui qui sera utilisé pour la recherche. Le programme qui le construit a été développé par un membre de l'équipe, Charles Pestieau (travail non publié).

Le graphe doit aussi contenir des arcs. Le choix des arcs est délicat. Il doit offrir le meilleur compromis entre couvrir un maximum de possibilités et minimiser l'erreur numérique. Les arcs les plus courts possible seront donc utilisés. Le déplacement selon l'axe des x sera donc toujours maintenu à une unité, sauf pour la montée et la descente. Effectivement, ces deux manœuvres

nécessitent deux unités sur l'axes des x pour permettre les accélérations et les changements d'altitude. En outre, le nombre d'arcs doit être aussi faible que possible puisqu'un trop grand nombre d'arcs augmenterait le temps de calcul. De plus, l'amplitude des virages doit être limitée par la structure du graphe, ce qui limite le nombre d'arcs sortants possibles pour un nœud. Effectivement, il est difficile d'imaginer un avion commercial qui pourrait se déplacer selon y de 120 kilomètres en n'avançant que de cinq kilomètres selon x . Le virage maximal permis est de 45° . Pour une grille orthonormée, la configuration choisie est montrée à la figure 3.4. L'état courant (symbole \bullet) permet d'atteindre cinq nœuds (symbole \star). Trois de ceux-ci se trouvent à la même altitude à l'indice de coût adjacent. Les virages se font donc à altitude constante. Tel que mentionné plus haut, la montée et la descente requièrent deux unités sur l'axe des x .

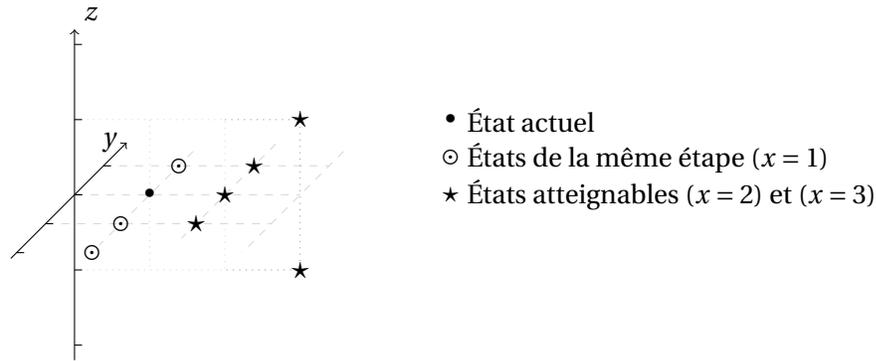


FIGURE 3.4: Nœud avec 5 arcs sortants

3.1.2 Variables

Les variables permettent de stocker l'information sur la position de l'avion et sa vitesse dans l'espace et le temps.

- i_c : indice de l'étape, position sur l'axe des x . Dans l'exemple de la figure 3.3, la tranche colorée montre l'étape d'indice $i_c = 3$.
- j_c, k_c : variables d'état, positions sur les axes des y et z . Un exemple de position (i, j, k) a été illustré à la droite de la figure 3.3.
- i_s, j_s, k_s : variables de décision, position sur les axes des x, y et z , respectivement. L'aéroport de départ est noté (I_0, J_0, K_0) et l'aéroport d'arrivée est noté (I_A, J_A, K_A) .
- u : variable de décision, temps (en secondes) pour atteindre (i_s, j_s, k_s) à partir de (i_c, j_c, k_c) .

3.1.3 Étiquette

La nomenclature du chapitre 2 est conservée. L'étiquette est donc nommée $R_{i_c j_c k_c}(C)$. Les ressources considérées sont le coût total ($m = 1$) tel que décrit en 1.3 et le carburant disponible au lancement de la mise à jour ($m = 2$). Étant donné la quasi-disparition des points de passage expliquée à la section 1.4.1, deux ressources ne sont pas considérées. Puisque le temps de passage n'est plus critique, le temps n'est pas considéré dans les ressources. En outre, pour s'assurer de passer par tous les points de passage, il aurait aussi fallu une ressource comptant les points de passage sur le chemin, ressource qui n'est plus nécessaire.

$R_{i_c j_c k_c}(C)$: vecteur contenant $R_{i_c j_c k_c}^1(C)$ et $R_{i_c j_c k_c}^2(C)$. Ces valeurs sont interprétées de la manière suivante : $R_{i_c j_c k_c}^1(C)$ est le coût total cumulé minimal tout au long du chemin C du nœud

(I_0, J_0, K_0) jusqu'au nœud (i_c, j_c, k_c) si le carburant consommé sur ce même chemin jusqu'à ce même nœud est $R_{i_c j_c k_c}^2(C)$.

3.1.4 Données, paramètres et ensembles

Cette sous-section décrit tous les paramètres du problème qui sont utiles pour modéliser le problème.

Éléments du graphe

- $\begin{bmatrix} a_{i_c j_c k_c}^1 & b_{i_c j_c k_c}^1 \\ a_{i_c j_c k_c}^2 & b_{i_c j_c k_c}^2 \end{bmatrix}$: matrice des contraintes sur le nœud (i_c, j_c, k_c) . La première ligne donne l'intervalle permis pour le coût et la deuxième, pour le carburant. Ainsi, la consommation de carburant au nœud (i_c, j_c, k_c) doit être d'au moins $a_{i_c j_c k_c}^2$ et d'au plus $b_{i_c j_c k_c}^2$. La valeur de $b_{i_c j_c k_c}^2$ sera calculée pour s'assurer de respecter la réglementation sur le carburant (voir la section 1.4.3) et le scénario critique de l'EDTO (voir la section 1.4.4). Remarquons que la quantité de carburant disponible pour la portion restante du vol est $b_{i_A j_A k_A}^2$, c'est-à-dire la consommation maximale au nœud de l'aéroport d'arrivée.
- \mathcal{D} : ensemble des aéroports de décollage pour le vol.
- (I_d, J_d, K_d) : coordonnées de l'aéroport $d \in \mathcal{D}$. La position des aéroports de décollage est nécessaire pour calculer les quantités réglementaires de carburant.

Zones tarifées

$s(i_c, j_c, i_s, j_s)$: coût de survol (en dollars) du segment qui joint (i_c, j_c, k_c) et (i_s, j_s, k_s) . Une fonction développée par un membre de l'équipe, Charles Pestieau, permet de calculer ce coût.

Carburant

- $r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u)$: coût (en kilogrammes de carburant) pour passer du point (i_c, j_c, k_c) au point (i_s, j_s, k_s) en u secondes si l'avion transporte $R_{i_c j_c k_c}^2(C)$ kilogrammes de carburant. Le calcul de ce coût est résumé à l'annexe B. Remarquons que le coût de carburant pour un arc du réseau n'est pas uniquement dépendant de la distance entre les deux nœuds qu'il relie. D'abord, la météo est prise en compte, donc la consommation dépend de la localisation des deux points sur la grille. De plus, la consommation de carburant dépend de la masse de l'avion et donc de la quantité de carburant qu'il transporte. Les implications de ces particularités du problème de recherche sont détaillées à la section 3.2.
- $\underline{R}_{i_c j_c k_c}^2$: coût (en kilogrammes de carburant) pour parcourir le chemin qui consomme le moins de carburant entre le nœud (i_c, j_c, k_c) et le nœud (I_A, J_A, K_A) .
- $\underbrace{R}_{i_c j_c k_c}^2$: coût (en kilogrammes de carburant) pour parcourir le chemin le plus rapide entre le nœud (i_c, j_c, k_c) et un aéroport de décollage.
- p : prix d'un kilogramme de carburant (en dollars)

Temps et délai

$d(t)$: coût d'un délai t .

Altitude

- FL_{max} : altitude maximale permise pour le vol par la réglementation.
- FL_{min} : altitude minimale permise pour le vol par la réglementation.
Les paramètres FL_{min} et FL_{max} seront utilisés pour la construction de la grille.
- $FL(i_c, j_c, i_s, j_s)$: ensemble des niveaux de vol atteignables selon la direction, en accord avec la réglementation présentée à la section 1.4.2. Pour un vol transatlantique ou transpacifique, on pourra se contenter de ne construire que la moitié des niveaux de vol, puisque l'ensemble des niveaux de vol ne varie pas en croisière.

Domaine de vol

- v : vitesse de l'avion.
- v_{max} : vitesse maximale permise selon l'enveloppe de vol.
- v_{min} : vitesse minimale permise selon l'enveloppe de vol.
- $V(i_c, j_c, k_c, R_{i_c j_c k_c}^2(C), v)$: ensemble des combinaisons (niveau de vol, masse, vitesse) permis par l'enveloppe de vol de l'avion. Tout dépendant de la masse et de l'altitude de l'avion, les vitesses possibles varient. L'enveloppe de vol dépend aussi de la météo.

EDTO

- $SEDTO$: seuil maximal de l'EDTO de l'État de l'opérateur (en minutes), déterminé selon le modèle de l'appareil.
- v_{EDTO} : vitesse à utiliser pour le calcul de l'EDTO. Cette vitesse varie selon l'appareil.
- $t_{EDTO}(i_c, j_c, k_c)$: durée minimale de vol entre le point (i_c, j_c, k_c) et l'aéroport de décollage le plus proche

Zones interdites

$\mathcal{Z}(i_c, j_c, k_c)$: ensemble des points atteignables depuis (i_c, j_c, k_c) ne faisant pas partie d'une zone interdite.

3.1.5 Fonctions d'extension de ressource

Une fonction d'extension de ressource est requise pour décrire la variation de chaque ressource. Deux fonctions sont donc présentées dans cette sous-section.

La première ressource est le coût. Le coût total comprend le coût du carburant et le coût de survol (3.1). Le coût initial du vol correspond au coût du délai (3.2).

$$f_{i_c j_c k_c i_s j_s k_s u}^1(R_{i_c j_c k_c}(C)) = R_{i_c j_c k_c}^1(C) + p \cdot r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) + s(i_c, j_c, i_s, j_s) \quad (3.1)$$

$$R_{I_0 J_0 K_0}^1(C) = d(t) \quad (3.2)$$

Utiliser la fonction $d(t)$ requiert une quantité énorme de calculs puisque le graphe en entier doit être résolu pour plusieurs valeurs de t . On introduit donc le CI, qui convertit le temps en kilogrammes de carburant. Cette conversion permet d'intégrer le coût du temps dans la fonction d'extension de ressource (3.3).

$$f_{i_c j_c k_c i_s j_s k_s u}^1(R_{i_c j_c k_c}(C)) = R_{i_c j_c k_c}^1(C) + p \cdot \left(r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) + CI \cdot u \right) + s(i_c, j_c, i_s, j_s) \quad (3.3)$$

Le coût initial du vol est donc de 0 et la condition initiale s'écrit comme en (3.4).

$$R_{I_0 J_0 K_0}^1(C) = 0 \quad (3.4)$$

Concrètement, dans la fonction qui calcule la consommation de carburant (voir annexe B), le CI intervient dans le choix de la vitesse optimale et détermine en conséquence le temps de parcours u . Un avantage du CI est qu'il est déjà utilisé par les opérateurs d'avions commerciaux. Cette estimation permet donc une bonne compatibilité avec les pratiques de l'industrie.

À la section 1.3.3, on remarquait que la valeur du CI est corrélée à l'heure d'arrivée. Pour cette recherche, nous faisons l'hypothèse que le CI et l'heure d'arrivée sont cohérents puisque le choix du CI est le sujet de recherche d'un autre membre de l'équipe, Mootez Ben Nasr.

La deuxième ressource est le carburant. La consommation en carburant sur le segment inclut la consommation requise pour passer d'un point à l'autre ainsi que la réserve associée. En vérifiant la contrainte (3.15) sur chaque nœud visité, on pourra s'assurer d'éliminer les chemins qui consomment plus que le carburant disponible.

$$f_{i_c j_c k_c i_s j_s k_s u}^2(R_{i_c j_c k_c}(C)) = R_{i_c j_c k_c}^2(C) + r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) \quad (3.5)$$

$$R_{I_0 J_0 K_0}^2(C) = 0 \quad (3.6)$$

On considérera que la consommation au départ est de 0, car on s'intéresse à la consommation du carburant disponible au moment de la mise à jour.

3.1.6 Formulation des contraintes

L'objectif final est d'éliminer toutes les étiquettes associées à des chemins qui ne respectent pas les contraintes. On peut distinguer deux types de contraintes. Dans le premier ensemble de contraintes, on choisit les successeurs qui sont inatteignables, peu importe le chemin emprunté auparavant. On élimine donc de visiter les nœuds qui ne permettront jamais de construire un chemin faisable. Dans le deuxième ensemble de contraintes, on choisit les chemins qui sont faisables. La qualité *faisable* d'un chemin est dépendante du carburant consommé auparavant. Ainsi, l'ensemble des chemins faisables pourrait ne pas retenir toutes les étiquettes d'un nœud atteignable.

Successeurs atteignables

Les conditions pour qu'un successeur fasse partie de l'ensemble des successeurs atteignables sont décrites ci-dessous. Parmi les chemins se terminant à ces successeurs atteignables, les chemins faisables seront choisis.

De toute évidence, le segment que l'on choisit de parcourir doit faire partie du graphe. En d'autres mots, le successeur choisi est l'un de ceux faisant partie de la configuration présentée à la figure 3.4.

$$\{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E \quad (3.7)$$

L'altitude doit correspondre aux niveaux de vol permis par l'OACI. Imaginons un avion se situant au niveau de vol FL 150. Les niveaux minimum et maximum pourraient être FL 140 et FL 200.

$$FL_{min} \leq k_s \leq FL_{max} \quad (3.8)$$

$$k_s \in FL(i_c, j_c, i_s, j_s) \quad (3.9)$$

La vitesse de l'avion doit se situer dans les vitesses possibles pour un vol. La vitesse ne fait pas partie des variables de décision. Cependant, les positions d'arrivée et de départ ainsi que le temps de parcours choisis permettent de la calculer.

$$v_{min} \leq v \leq v_{max} \quad (3.10)$$

L'avion ne doit pas se trouver dans une zone interdite.

$$(i_s, j_s, k_s) \in \mathcal{Z}(i_c, j_c, k_c) \quad (3.11)$$

L'avion ne doit pas dépasser le seuil maximal de l'EDTO

$$t_{EDTO}(i_s, j_s, k_s) \leq \frac{S_{EDTO}}{v_{EDTO}} \quad (3.12)$$

Définissons l'ensemble $\mathcal{A}(i_c, j_c, k_c)$, qui contient l'ensemble des nœuds atteignables pour un nœud (i_c, j_c, k_c) :

$$\begin{aligned} \mathcal{A}(i_c, j_c, k_c) = \{ & \{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E \\ & k_s : FL_{min} \leq k_s \leq FL_{max} \text{ et } k_s \in FL(i_c, j_c, i_s, j_s) \\ & (i_s, j_s, k_s) \in Z(i_c, j_c, k_c) \\ & (i_s, j_s, k_s) : t_{EDTO}(i_s, j_s, k_s) \leq \frac{S_{EDTO}}{v_{EDTO}} \} \end{aligned} \quad (3.13)$$

Chemins faisables

Certaines contraintes dépendent du chemin C emprunté pour arriver au point (i_c, j_c, k_c) et du successeur (i_s, j_s, k_s) choisi pour étendre le chemin C . Le nouveau chemin C' se terminant en (i_s, j_s, k_s) est nécessairement une extension d'un chemin C faisable. De plus, le successeur choisi doit être atteignable.

L'altitude et la vitesse choisies doivent être parmi les altitudes permises par l'enveloppe de vol.

$$(i_s, j_s, k_s, R_{i_c j_c k_c}^2(C)) \in V(i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), v) \quad (3.14)$$

La quantité de carburant doit se trouver dans les quantités permises. Une borne supérieure sera définie par la réserve requise par la réglementation et la quantité disponible.

$$R_{i_c j_c k_c}^2(C) + r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) \leq b_{i_s j_s k_s}^2 \quad (3.15)$$

Ainsi, pour qu'un chemin C' , extension d'un chemin C , se terminant en (i_s, j_s, k_s) soit faisable il doit faire partie de $\mathcal{F}((I_0 J_0 K_0), (i_s j_s k_s))$.

$$\begin{aligned} \mathcal{F}((I_0 J_0 K_0), (i_s j_s k_s)) = \{ & C \in \mathcal{F}((I_0 J_0 K_0), (i_c j_c k_c)) \\ & (i_s, j_s, k_s) \in \mathcal{A}(i_c, j_c, k_c) \\ & (i_s, j_s, k_s, R_{i_c j_c k_c}^2(C)) \in V(i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), v) \\ & a_{i_s j_s k_s}^2 \leq R_{i_c j_c k_c}^2(C) + r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) \leq b_{i_s j_s k_s}^2 \} \end{aligned} \quad (3.16)$$

3.2 Algorithme de résolution

Dans cette section, le programme qui permettra de résoudre le problème de la mise à jour de trajectoire pour un avion est décrit dans les grandes lignes. L'algorithme 3.1 résume toutes les étapes du programme. Il s'inspire de l'algorithme générique d'Irnich et Desaulniers (2004) présenté à la section 2.5.1. L'algorithme débute au nœud du réseau qui représente l'aéroport de départ. L'objectif est de trouver le chemin le moins cher jusqu'à l'arrivée sans dépenser plus de carburant que la quantité dont on dispose. Pour ce faire, on étend des chemins en leur ajoutant un segment jusqu'à ce qu'un chemin complet et satisfaisant soit trouvé. Comment sélectionner le chemin adéquat? On ne peut pas décider de ne conserver que les chemins les moins chers : on courrait le risque de trouver une solution qui consomme trop de carburant. On ne peut pas non plus décider de conserver que les chemins qui consomment peu de carburant : on trouverait probablement une solution beaucoup trop coûteuse. On retardera alors le plus possible le moment où il faudra choisir un chemin plutôt qu'un autre. Ainsi, plusieurs chemins différents pourraient se terminer au même nœud, sans toutefois que l'un de ces chemins soit clairement plus avantageux que les autres. Un des objectifs de la méthode est de calculer la solution *rapidement*. On veut donc générer le moins de chemins possible, tout en s'assurant de ne pas éliminer la solution optimale. On gardera donc pour chaque nœud le nombre minimum de chemins tout en s'assurant de garder tous ceux qui pourraient se trouver dans le chemin optimal. Les prochains paragraphes formalisent cette méthode de résolution.

Algorithme 3.1 Algorithme pour la résolution du problème de la mise à jour de trajectoire

```

1: Construire le réseau
2: Calculer les bornes  $[a_{i_c j_c k_c}, b_{i_c j_c k_c}]$  pour tous les  $(i_c, j_c, k_c) \in V$ 
3: Initialiser  $(I_0, J_0, K_0)$ 
4:  $\mathcal{U} \leftarrow (I_0, J_0, K_0)$ 
5:  $\mathcal{C} \leftarrow \emptyset$ 
6: tant que  $\mathcal{U} \neq \emptyset$  faire
7:   Choisir un nœud  $(i_c, j_c, k_c) \in \mathcal{U}$ 
8:   pour tout  $\{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E$  faire
9:     si  $(i_s, j_s, k_s) \in \mathcal{A}(i_c, j_c, k_c)$  alors
10:      pour tout chemin  $C \in (\mathcal{U} \cap \mathcal{F}((I_0, J_0, K_0), (i_c, j_c, k_c)))$  faire
11:         $\mathcal{U} = \mathcal{U} - C$ 
12:        Calculer  $r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_D - R_{i_c j_c k_c}^2(C))$   $\triangleright$  Voir annexe B pour le calcul
13:        si  $(C, (i_s, j_s, k_s)) \in \mathcal{F}((I_0, J_0, K_0), (i_s, j_s, k_s))$  alors
14:          Calculer  $R_{i_s j_s k_s}(C)$   $\triangleright$  Selon les fonction présentées à la section 3.1.5
15:           $\mathcal{C} = \mathcal{C} \cup (C, (i_s, j_s, k_s))$ 
16:        fin si
17:      fin pour
18:      Élagage par dominance au nœud  $(i_s, j_s, k_s)$   $\triangleright$  Voir annexe E pour l'algorithme
19:    fin si
20:  fin pour
21: fin tant que
22: Choisir un chemin  $C$  t.q.  $\min_{C \in \mathcal{C}} R_{I_A J_A K_A}^1(C)$ 

```

En premier lieu, le réseau doit être construit, ce qui est fait à la ligne 1. Seuls les nœuds sont créés, puisque le coût des arcs varie en fonction de la masse de l'avion. On pourrait discrétiser la masse de l'avion et générer un arc pour chaque valeur, mais le temps requis pour générer un

tel nombre d'arcs serait excessif. Puisque les valeurs des arcs ne sont pas connues, la relaxation lagrangienne ne pourra être utilisée pour valider la réponse. Le programme qui crée cette grille a été développé par un membre de l'équipe, Charles Pestieau. Les grandes étapes sont énumérées ici. À l'aide de la position des aéroports de départ et d'arrivée, les paramètres a et b de l'ellipse sont calculés. Des nœuds sont créés suivant la procédure de discrétisation présentée à la section 3.1.1. À chaque nœud sont assignés une position géographique en trois dimensions, un triplet (i, j, k) ainsi que des données météorologiques. La météo est interpolée linéairement à partir de fichiers GRIB, un standard aéronautique. Les nœuds qui sont au-dessus de FL_{max} ou sous FL_{min} ne sont pas créés. Les nœuds faisant partie d'une zone interdite peuvent être supprimés à cette étape.

À la ligne 2, les bornes de carburant sont calculées. Cette opération est réalisée en deux étapes. Dans un premier temps, les valeurs de consommation de carburant possibles à chaque nœud sont restreintes pour s'assurer de respecter la réglementation présentée aux sections 1.4.3 et 1.4.4. Cette étape élimine tous les points qui sont trop éloignés des aéroports de décollage. La réglementation requiert que l'avion transporte une quantité minimale de carburant à certains points de la trajectoire. Une formulation transformée permet de comprendre comment cette situation est modélisée : en limitant la quantité de carburant que peut consommer l'avion, on s'assure qu'il transportera au moins la quantité minimale de carburant requise pour respecter les contraintes des sections 1.4.3 et 1.4.4. Cette situation est illustrée dans la partie gauche de la figure 3.5. De la quantité maximale de carburant disponible pour le vol est soustrait le minimum requis pour satisfaire la réglementation (en bleu), ce qui réduit la consommation possible au nœud aux valeurs se situant entre a_{ijk}^2 et \underbrace{R}_{ijk}^2 . L'algorithme utilisé pour calculer ces bornes est présenté à l'annexe F. Dans un deuxième temps, la stratégie d'élagage par infaisabilité présentée à la section 2.4.1 sera mise en œuvre. Pour ce faire, un algorithme de plus court chemin minimisant la consommation de carburant sera réalisé dans le sens inverse du calcul de la trajectoire (voir annexe G pour le calcul). Ce calcul permettra de stocker en chaque nœud la quantité minimale de carburant requise pour se rendre à destination. Ce calcul est équivalent à une planification initiale avec un CI et une réserve à l'arrivée de zéro. Un CI de zéro minimise la consommation de carburant sans tenir compte du temps. Une réserve à l'arrivée nulle permet de réaliser les calculs pour un avion qui vole avec une masse minimale et qui a donc une consommation de carburant minimale. La contrainte produite est illustrée dans la partie droite de la figure 3.5. Le minimum requis pour se rendre à destination (en rouge) est soustrait du carburant total disponible. La consommation devra donc se situer entre a_{ijk}^2 et \underline{R}_{ijk}^2 . Deux éléments sont donc considérés dans le calcul des bornes : la réglementation et l'élagage. Chaque élément réduit la consommation possible au nœud. La borne b_{ijk}^2 prendra la valeur la plus faible de consommation possible pour chaque nœud ou, en d'autres mots, prendra la valeur calculée pour la situation la plus contraignante. Dans l'exemple de la figure 3.5, l'élagage par infaisabilité est le plus contraignant, ce qui nous mène à $b_{ijk}^2 = \underline{R}_{ijk}^2$.

À la ligne 3, le nœud de départ est initialisé. Quel est ce point de départ ? Pour le déterminer, on doit déterminer les informations dont nous disposons et les informations requises pour le calcul. En tout point, pour lancer un calcul de consommation de carburant, il est nécessaire de connaître

- la masse de l'avion ;
- la vitesse de l'avion ;
- le point de destination et de départ du segment à parcourir.

Pour la planification initiale, le calcul est effectué dans le sens inverse du vol, car la seule masse connue est celle de l'atterrissage. Effectivement, le pilote décide de la réserve de carburant qu'il

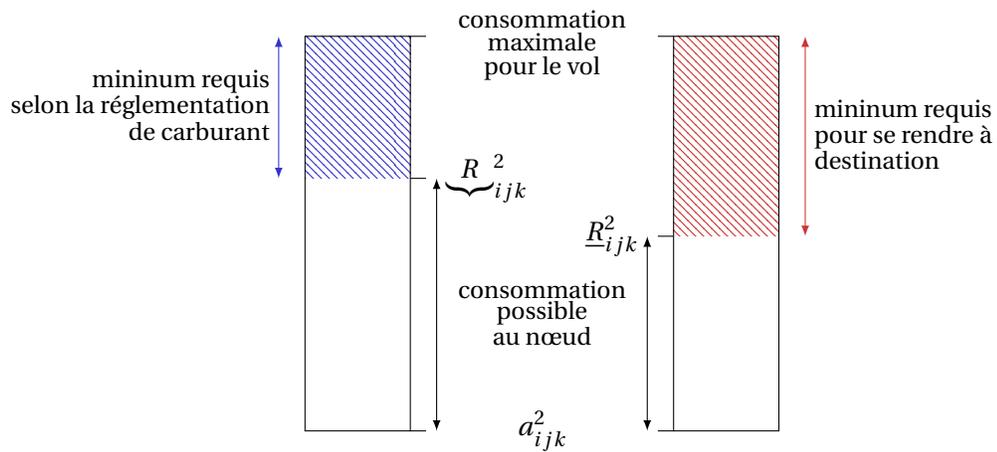


FIGURE 3.5: Illustration de la quantité maximale consommée à un nœud

désire avoir à l'arrivée, ce qui permet de calculer la masse de l'avion. À la figure 3.6, les cercles indiquent les endroits auxquels les trois informations requises sont connues au lancement du calcul. Le cercle de la figure 3.6a est placé à l'arrivée. En calculant le carburant requis à rebours de l'arrivée jusqu'au départ, dans une direction indiquée pour quelques points par les flèches bleues, on peut trouver ensuite la trajectoire que parcourra l'avion (flèches noires). Pour la mise à jour, le calcul doit être fait dans le sens du vol, car l'endroit où la mise à jour est lancée est le seul où la masse de l'avion est connue. La masse à l'arrivée n'est pas connue puisque l'imprévu qui force à mettre à jour la trajectoire peut modifier la masse d'arrivée choisie lors de la planification du vol. Les flèches rouges de la figure 3.6b permettent de voir que le calcul se fait dans le même sens que le vol (en noir). Ainsi, l'initialisation de la ligne 3 consiste à assigner au nœud de départ la masse et la vitesse actuelles de l'avion.

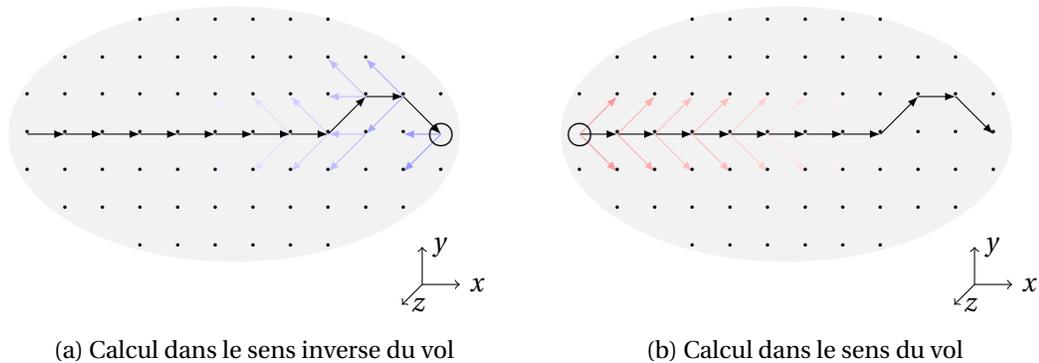


FIGURE 3.6: Illustration des deux sens possibles pour le calcul

Les lignes 4 et 5 de l'algorithme initialisent les ensembles de nœuds. L'ensemble \mathcal{C} contient les chemins faisables qui ont été étendus. En pratique, l'ensemble \mathcal{C} ne sera pas créé puisqu'il n'est pas nécessaire. Tout chemin stocké dans un nœud sera considéré comme faisant partie de l'ensemble \mathcal{C} . L'ensemble \mathcal{U} ne sera pas créé pour tous les algorithmes. L'utilisation de cet ensemble dépend de l'algorithme choisi. Pour l'algorithme inspiré de Bellman-Ford, les nœuds sont visités dans un ordre séquentiel, ils n'ont donc pas besoin d'être stockés dans un ensemble.

La boucle de la ligne 6 se terminera quand le nœud de destination sera visité ou quand aucun des nœuds restants à évaluer ne sera associé à un chemin faisable.

C'est à la ligne 7, dans le choix du nœud à visiter, que l'algorithme de résolution choisi joue son rôle. L'algorithme se base sur le principe de l'algorithme 2.2, mais est ajusté pour tenir compte de la contrainte de ressource. Les nœuds sont choisis dans une séquence connue à l'avance. Les algorithmes à fixation d'étiquettes ne sont pas vraiment une option dans ce cas-ci, puisqu'ils ne permettent pas toujours de trouver une trajectoire faisable.

Concernant l'ensemble E des arcs composant le graphe, il contient cinq arcs sortants pour chaque nœud selon la configuration présentée à la figure 3.4. Ainsi, à partir de chaque nœud, cinq options sont possibles :

- Avancer d'une unité selon x ;
- Avancer d'une unité selon x et se déplacer de +1 unité en y ;
- Avancer d'une unité selon x et se déplacer de -1 unité en y ;
- Avancer de deux unités selon x et monter d'une unité en z ;
- Avancer de deux unités selon x et descendre d'une unité en z .

La boucle de la ligne 8 traite ces cinq options de nœud successeur. L'intérieur de cette boucle consiste à vérifier si le successeur choisi fait partie d'un chemin faisable. La vérification de la ligne 9 est équivalente à déterminer si le successeur est atteignable. L'information est tirée de l'intervalle $[a_{ijk}^2, b_{ijk}^2]$ stocké dans le nœud successeur. Un intervalle pour lequel $b_{ijk}^2 < a_{ijk}^2$ ou $b_{ijk}^2 = a_{ijk}^2 = 0$ indique que le nœud successeur est trop éloigné d'un aéroport de dégagement ($t_{EDTO} > \frac{SEDTO}{VEDTO}$) ou qu'il se situe dans une zone interdite ($(i_s, j_s, k_s) \notin \mathcal{Z}(i_c, j_c, k_c)$).

Si la condition est vraie, la faisabilité des chemins sera évaluée. À l'intérieur de la boucle de la ligne 10 on tente d'étendre vers le nœud successeur atteignable tous les chemins arrivant au nœud (i_c, j_c, k_c) . La consommation de carburant entre le nœud visité et le nœud successeur est calculée. Elle permet de vérifier si l'intervalle $[a_{i_s j_s k_s}^2, b_{i_s j_s k_s}^2]$ est respecté. Elle permet aussi de calculer la masse et de s'assurer que l'avion n'entame pas de montée s'il est trop lourd. La fonction qui calcule la consommation de carburant choisit toujours une vitesse faisant partie de l'enveloppe de vol.

Si le chemin étendu est faisable, une étiquette est stockée dans le nœud successeur. Avant de passer à un prochain successeur, l'algorithme de dominance est lancé, de manière à ce que tout nœud visité ne soit associé qu'à un ensemble d'étiquettes non dominées. Le pseudo-code de l'algorithme de dominance est présenté à l'annexe E.

Si une solution existe, le chemin optimal sera stocké dans le nœud de destination à la fin de l'exécution de l'algorithme.

Chapitre 4

Résultats

Ce chapitre explore les résultats obtenus suite à la mise en œuvre de l'algorithme décrit au chapitre 3. En premier lieu, les différences entre l'algorithme de mise à jour et l'algorithme de planification initiale seront exposées. En se concentrant ensuite uniquement sur l'algorithme de mise à jour, la solution fournie par l'algorithme sera expliquée. Les deux sous-sections suivantes décrivent la manière dont la solution est affectée d'abord par les contraintes et ensuite par les coûts. Finalement, l'effet des différents paramètres (précision de la grille, CI et carburant disponible) sur la solution et le temps de calcul est étudié.

4.1 Planification initiale et mise à jour

Depuis l'introduction, il a été mis au clair que le problème de planification initiale et le problème de mise à jour sont deux problèmes différents. L'algorithme de planification initiale en direct (algorithme 4.1) visite tous les nœuds dans un ordre prédéterminé et choisit le chemin le moins coûteux. Il est similaire à l'algorithme de Bellman-Ford (algorithme 2.2) présenté en 2.5.2. Effectivement, les nœuds sont visités dans un ordre prédéterminé. Cependant, comme le graphe est acyclique, chaque nœud n'est visité qu'une fois. L'algorithme de mise à jour de la trajectoire a été présenté à la section 3.2 et choisit le chemin le moins coûteux pour une quantité de carburant donnée.

La figure 4.1 met en évidence la différence entre les deux algorithmes en comparant la consommation de trajectoires en croisière complètes. L'algorithme utilisé pour générer les données de la planification initiale est très similaire à l'algorithme développé par un autre membre de l'équipe, Charles Pestieau, mais celui qui est utilisé dans ce mémoire à des fins de comparaison réalise le calcul du départ vers l'arrivée plutôt que de l'arrivée vers le départ. Cela permet de réaliser le calcul dans exactement les mêmes conditions et ainsi d'avoir une comparaison plus juste. L'algorithme utilisé pour générer les données de mise à jour est le même que celui présenté à la section 3.2 et sera utilisé pour le reste de ce chapitre.

La ligne pointillée de la figure 4.1 indique la quantité de carburant disponible, c'est-à-dire la quantité de carburant embarquée dans l'avion qui peut être consommée sur le trajet. Même si seulement une partie du carburant est consommée, tout le carburant, celui qui est consommé et celui qui n'est pas consommé, doit être transporté jusqu'à destination. Les chemins représentés par un point se situant dans la zone grise au-dessus de la ligne pointillée sont infaisables en pratique car ils consomment plus de carburant que la quantité disponible. Les croix illustrent les consommations de carburant pour les trajectoires déterminées avec l'algorithme de

Algorithme 4.1 Algorithme de planification initiale en direct de la trajectoire

```

1: Construire le réseau
2: Éliminer les nœuds trop éloignés des aéroports de décollage
3: Initialiser  $(I_0, J_0, K_0)$ 
4:  $\mathcal{U} \leftarrow (I_0, J_0, K_0)$ 
5: tant que  $\mathcal{U} \neq \emptyset$  faire
6:   Choisir un nœud  $(i_c, j_c, k_c) \in \mathcal{U}$  dans l'ordre topologique
7:    $\mathcal{U} = \mathcal{U} - (i_c, j_c, k_c)$ 
8:   pour tout  $\{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E$  faire
9:     si  $(i_s, j_s, k_s) \in \mathcal{A}(i_c, j_c, k_c)$  alors
10:       Calculer  $r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_D - R_{i_c j_c k_c}^2(C))$   $\triangleright$  Voir annexe B pour le calcul
11:       si  $f_{i_c j_c k_c i_s j_s k_s}^1(R_{i_c j_c k_c}(C)) < R_{i_s j_s k_s}^1$  alors
12:          $\mathcal{U} = \mathcal{U} \cup (i_s, j_s, k_s)$ 
13:         Calculer  $R_{i_s j_s k_s}(C)$   $\triangleright$  Selon les fonction présentées à la section 3.1.5
14:       fin si
15:     fin si
16:   fin pour
17: fin tant que

```

planification initiale et les cercles, les consommations pour les trajectoires déterminées avec l'algorithme de mise à jour.

La figure 4.1 permet de mettre en relief quelques éléments. D'abord, elle permet de constater que la consommation calculée par les deux algorithmes augmente avec la quantité disponible. La dépense additionnelle en carburant permet de parcourir la trajectoire sans augmenter la durée du vol bien que la masse de l'avion soit augmentée par l'ajout de carburant. C'est donc que pour les conditions dans lesquelles le calcul a été lancé, il est moins coûteux de dépenser un peu plus de carburant que de retarder l'atterrissage dans le but de maintenir la consommation à un niveau inférieur. Ensuite, les deux algorithmes calculent la même consommation en cinq occurrences. Le phénomène se produit lorsque la quantité de carburant disponible est au moins égale à celle qui est consommée par une trajectoire calculée par l'algorithme de planification initiale. Dans ce cas, la quantité consommée, la trajectoire et la vitesse à laquelle est parcourue sont identiques pour les deux vols. Le tableau 4.1 recense les coûts et les durées de trajectoires calculées par les deux algorithmes. Lorsque le carburant disponible est supérieur ou égal à la consommation de carburant de la planification initiale, le coût et la durée des trajectoires calculées par les deux algorithmes sont exactement les mêmes. L'algorithme de planification initiale choisit le chemin le plus économique sans tenir compte de la quantité de carburant disponible. Si la quantité de carburant requise par la planification initiale ne viole pas la contrainte, le chemin calculé est aussi la solution de la mise à jour. Cependant, lorsque la quantité disponible est moindre que celle requise par l'algorithme de planification initiale, les consommations sont très différentes, car l'algorithme de mise à jour propose une trajectoire différente qui tient compte de la limitation de carburant. L'algorithme de mise à jour ne peut trouver de chemin faisable avec un CI de 100, par exemple, si seulement 41 000 kilogrammes de carburant sont disponibles. Il trouvera cependant un chemin consommant 41 510 kilogrammes de carburant si sa réserve est de 42 000 kilogrammes de carburant. L'algorithme de planification initiale calcule pour sa part une trajectoire qui consomme 43 500 kilogrammes.

Ce nouveau chemin faisable est différent du premier, tel qu'illustré à la figure 4.2. Le trait noir indique la trajectoire de l'avion qui a au moins 43 600 kilogrammes de carburant dis-

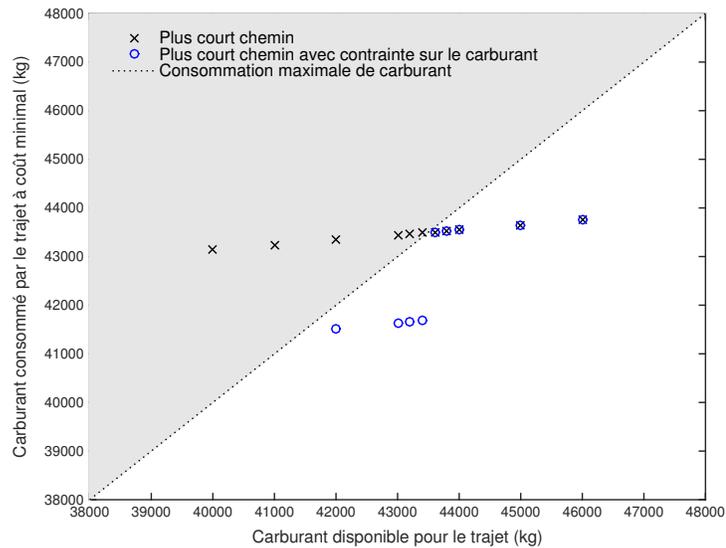


FIGURE 4.1: Comparaison de la consommation de trajectoires calculées par les algorithmes de planification initiale et de mise à jour. $CI = 100$, $p = 1$

ponible. Le trait bleu montre la trajectoire lorsque cette quantité est diminuée à 43 400 kilogrammes. Le profil horizontal des deux vols est le même, mais le vol s'effectue à une altitude plus élevée, car la consommation de carburant à cette altitude est réduite.

Bien que la consommation de carburant diminue, le coût du vol ne diminue pas pour autant. L'ajout d'une contrainte à un problème d'optimisation ne permet jamais d'améliorer le coût de la solution, car la contrainte peut rendre infaisable la solution la moins coûteuse. Dans la situation illustrée à la figure 4.2, le fait de resserrer la contrainte de carburant, c'est-à-dire de diminuer la quantité de carburant disponible, fait voler l'avion à plus haute altitude, car cela permet d'économiser du carburant. Cependant, la durée du vol s'allonge et le coût augmente en conséquence. Le tableau 4.1 met ce phénomène en évidence. Lorsque la quantité de carburant disponible est de 43 000 kilogrammes, l'algorithme de mise à jour calcule un chemin qui coûte 9 332 \$ et requiert environ 3 minutes de plus celui calculé avec 44 000 kilogrammes disponibles. Cette différence est le résultat de la modification de trajectoire observée à la figure 4.2.

4.2 Étude de la solution

Cette section explique les résultats obtenus par l'exécution de l'algorithme 3.1. Dans un premier temps, un chemin obtenu dans un environnement sans contraintes sera présentée. Ensuite, les bornes de carburant seront ajoutées, ce qui permettra de constater l'effet de ces bornes sur la trajectoire et son coût. Finalement, les zones tarifées seront introduites, ce qui fournira un portrait complet des trajectoires générées par l'algorithme.

4.2.1 Trajectoire sans contraintes

La solution de l'algorithme de planification initiale n'est pas une trajectoire qui relie le départ à l'arrivée. Elle s'apparente plutôt à une stratégie ou une politique qui décrit comment se déplacer du nœud de départ vers chacun des nœuds du réseau. La solution de l'algorithme de

TABLEAU 4.1: Comparaison de temps de vol et de coût entre l'algorithme de planification initiale et l'algorithme de mise à jour. CI = 100, $p = 1$

Carburant disponible (kg)	Planification initiale			Mise à jour		
	Carburant (kg)	Coût (\$)	Durée (s)	Carburant (kg)	Coût (\$)	Durée (s)
40 000	43 171	2 188 873	21 457	–	–	–
41 000	43 273	2 188 971	21 457	–	–	–
42 000	43 374	2 189 070	21 457	41 591	2 197 885	21 563
43 000	43 477	2 189 168	21 457	41 713	2 198 022	21 563
43 200	43 497	2 189 188	21 457	41 746	2 198 049	21 563
43 400	43 518	2 189 208	21 457	41 771	2 198 076	21 563
43 600	43 538	2 189 228	21 457	43 538	2 198 220	21 457
43 800	43 559	2 189 248	21 457	43 559	2 198 247	21 457
44 000	43 579	2 189 267	21 457	43 579	2 189 267	21 457
45 000	43 682	2 189 375	21 457	43 682	2 189 375	21 457
46 000	43 786	2 189 467	21 457	43 786	2 189 467	21 457

mise à jour ressemble à celle de la planification initiale, mais elle peut proposer plusieurs chemins différents entre le nœud de départ et un autre nœud du réseau qui dépendent de la quantité de carburant disponible. À la figure 4.3, les trois trajectoires obtenues par l'exécution de l'algorithme 3.1 qui relie le départ à l'arrivée ont été dessinées. La première option (en bleu) est la moins chère à 2 189 625 \$, mais c'est aussi celle qui consomme le plus de carburant (43 654 kg). L'option la plus chère (en rouge) est aussi celle qui consomme le moins de carburant.

Ainsi, la solution de l'algorithme de plus court chemin avec contraintes peut fournir plusieurs options de chemin à chaque nœud. Seuls les chemins dont les étiquettes sont non inférieures (voir définition 2.2 à la page 23) sont conservés.

4.2.2 Effet des bornes de la consommation de carburant

Le calcul des bornes de consommation de carburant se fait en deux étapes. D'abord, les contraintes de carburant sont calculées dans l'ensemble du réseau. Ensuite, le dual est calculé et combiné avec les contraintes de carburant. Cette section examine l'impact des bornes sur la forme de la solution et le temps de calcul.

Contraintes de carburant

Pour montrer la différence entre la mise à jour et la planification ainsi que pour expliquer la forme de la solution, l'espace aérien utilisé ne tenait pas en compte les contraintes de carburant. Elles sont donc introduites ici. Notons que le calcul de ces contraintes peut aussi être utilisé pour la planification initiale, bien que l'algorithme qui les calcule ait été développé dans le cadre du projet sur lequel porte ce mémoire. Le calcul peut être réalisé à l'avance, car il ne dépend pas de la trajectoire ou de la masse actuelle de l'avion, mais seulement de la météo et de la masse de l'avion sans carburant. Ce calcul ne sera donc pas inclus dans le temps de calcul de la mise à jour. La figure 4.4 illustre les contraintes de carburant pour une altitude donnée. Les aéroports de dégagement sont indiqués par des points noirs. À proximité de ceux-ci, la réserve requise est faible, comme l'indiquent les données colorées en jaune. En s'éloignant des aéroports de dégagement, la réserve requise augmente et la couleur des points tend de plus en plus vers l'indigo. La région gris pâle est au-delà du seuil maximal de l'EDTO et est donc interdite.

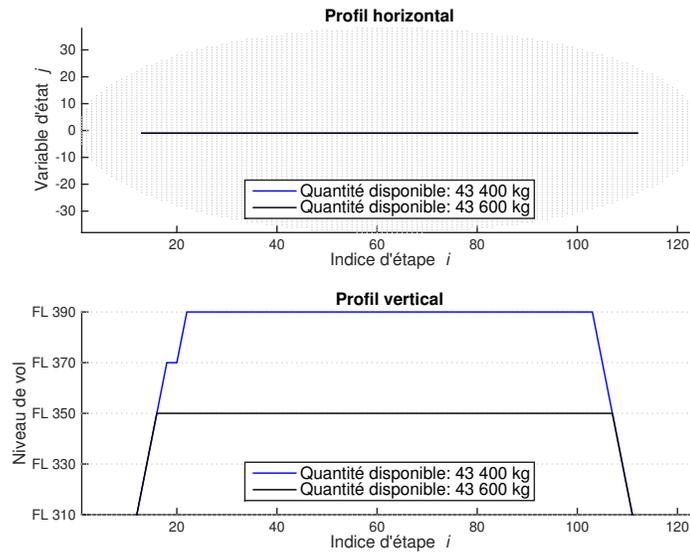


FIGURE 4.2: Comparaison de deux trajectoires avec des quantités de carburant disponibles différentes. $CI = 100$, $p = 1$

L'avion peut donc voler dans la région colorée tant qu'il transporte les réserves de carburant nécessaires. Si un chemin ne permet pas de transporter les réserves suffisantes en tout point, il est éliminé.

En plus de varier en fonction de la distance avec l'aéroport de décollage, la quantité de carburant requise varie aussi avec l'altitude. Le tableau 4.2 indique que pour se rendre au nœud (25,56,16), l'avion ne peut avoir consommé plus de 37 621 kilogrammes de carburant et que pour se rendre au nœud (25,56,19), il ne peut avoir consommé plus de 32 207 kilogrammes. La quantité requise au nœud (25,56,19) est supérieure de 5 414 kilogrammes à celle du nœud (25,56,16). On peut interpréter cette relation de la manière suivante : il faut 5 414 kilogrammes de carburant de plus pour se rendre à l'aéroport de décollage à partir du nœud (25,26,19) qu'il n'en aurait fallu pour s'y rendre depuis (25,26,16). Cette différence est due au fait que la descente vers l'aéroport de décollage est plus longue lorsque l'altitude de départ est plus élevée. L'algorithme cherche donc une trajectoire optimale qui présente un équilibre entre avoir besoin de plus de carburant dans le réservoir pour voler à plus haute altitude et voler à plus haute altitude pour faire des économies en carburant.

TABLEAU 4.2: Consommation de carburant maximale en fonction de l'altitude

Niveau de vol	FL 310	FL 330	FL 350	FL 370	FL 390
k_c	15	16	17	18	19
$b_{25,56,k_c}^2$ (kg)	38 699	37 621	36 443	34 706	32 207

Avant d'évaluer comment les contraintes de l'EDTO affectent la trajectoire, deux remarques doivent être faites sur la forme des contraintes. D'abord, la variation de la couleur, et donc de la contrainte, devrait être lisse autour des aéroports de décollage. La manière dont le calcul est réalisé introduit des variations brusques dans la contrainte. Les trajets parallèles aux axes, notamment, sont plus avantageux que ceux qui en dévient légèrement. Plutôt que de voir des cercles concentriques de couleur autour des aéroports de décollage, on observe plutôt des

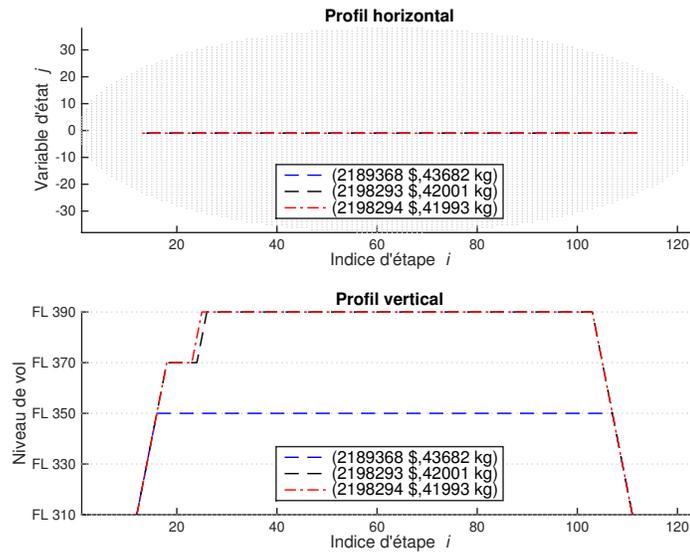


FIGURE 4.3: Comparaison des chemins possibles entre le départ et l'arrivée. $b_{IAJA}^2 = 45000$, $CI = 100$, $p = 1$

losanges. Cette déformation a un effet sur la qualité des trajectoires en les rendant moins lisses. En outre, déterminer les réserves d'une manière en tout point conforme à la réglementation (voir OACI (2014b)) est un problème d'optimisation complexe qui tient compte de la masse de l'avion. Le calcul développé dans le cadre de ce travail est donc une bonne approximation, mais il pourrait être perfectionné.

La figure 4.5 permet d'illustrer un des impacts de l'EDTO. Le profil horizontal de chaque chemin doit être compris dans la zone de la couleur qui lui est associée. Par exemple, la zone et le chemin bleus sont associés à $s_{EDTO} = 90$. Lorsque le seuil maximal est de 90 minutes, le profil horizontal du vol est le même que celui d'un vol sans contrainte de carburant. Or, lorsque le seuil maximal réduit, la trajectoire doit se rapprocher des aéroports de décollage. D'une part, lorsque la zone EDTO réduit, la taille de la zone interdite augmente et la ligne centrale de l'ellipse est obstruée. Le nouveau chemin doit donc contourner l'obstacle. D'autre part, les réserves requises diminuent lorsque l'avion s'approche des aéroports de décollage, ce qui peut expliquer pourquoi la trajectoire ne suit pas exactement le contour de l'obstacle.

Lorsque le profil horizontal change, le coût, la consommation de carburant et la durée du vol changent aussi nécessairement. Les vols illustrés à la figure 4.5 correspondent aux données des deux dernières colonnes du tableau 4.3. Lorsque le seuil maximal est au-delà de 150 minutes, la trajectoire reste la même, car la zone EDTO est suffisamment grande pour contenir la trajectoire optimale. Lorsque le seuil maximal diminue et que le profil horizontal du vol se déplace vers les aéroports de décollage, le coût et la durée du vol augmentent. La trajectoire bleue de la figure 4.5 requiert 26 minutes de plus que la trajectoire noire. La quantité de carburant disponible est un facteur important à prendre en considération. Par exemple, il est impossible de trouver un vol avec 50 000 kilogrammes de carburant disponible si le seuil maximal est de 90 minutes et le CI, de 100.

Lorsque la quantité de carburant disponible diminue, le coût du vol peut augmenter. La figure 4.6 montre deux trajectoires différentes pour un même seuil maximal. Le scénario illustré en bleu dispose de moins de carburant. Il vole donc à plus haute altitude en début de vol pour

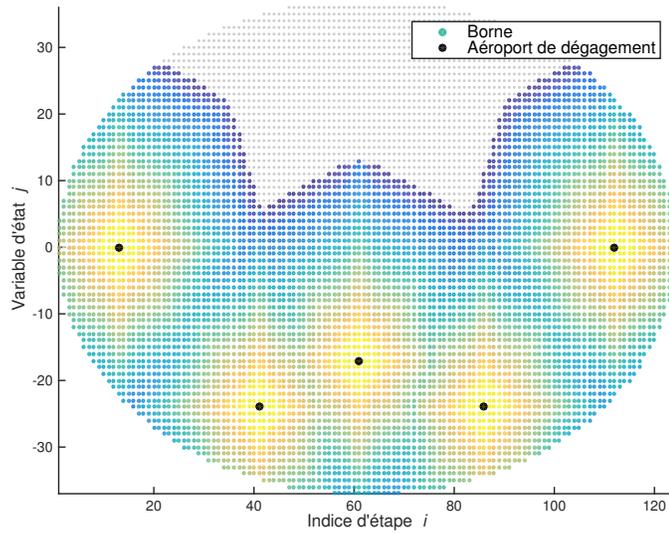


FIGURE 4.4: Bornes supérieures de consommation calculées avec les contraintes de carburant. $SEDTO = 150$

TABLEAU 4.3: Coût et durée d'un vol en fonction du seuil maximal et du carburant disponible. $CI = 100, p = 1$

$SEDTO$ (min)	Carburant disponible (kg)					
	50 000		53000		56000	
	Coût (\$)	Durée (s)	Coût (\$)	Durée (s)	Coût (\$)	Durée (s)
90	–	–	2 359 500	23 115	2 351 100	23 032
120	2 216 300	21 724	2 209 400	21 644	2 209 400	21 640
150	2 192 300	21 477	2 190 500	21 460	2 190 800	21 460
180	2 192 300	21 477	2 190 500	21 460	2 190 800	21 460

économiser du carburant puis effectue une descente par palier jusqu'à FL 310 pour profiter du fait que les réserves requises sont plus faibles et que la vitesse de croisière peut être plus élevée à plus basse altitude. En comparant les quantités de carburant disponibles utilisées dans cette section à celles utilisées à la section 4.1, on constate qu'elles sont plus élevées. Effectivement, pour pouvoir satisfaire les contraintes de carburant, l'avion a davantage de carburant à sa disposition.

Élagage par infaisabilité

Afin d'accélérer la résolution, il était proposé d'introduire l'élagage par infaisabilité dont il a été question en premier à la section 2.4.1. Le dual de la ressource est calculé en déterminant le plus court chemin qui minimise la consommation de carburant de l'aéroport d'arrivée vers l'aéroport de départ (voir annexe G pour l'algorithme). Le plus court chemin est calculé uniquement dans la zone permise par la réglementation de l'EDTO ce qui permet de prendre en compte les obstacles dans la consommation minimale de carburant. Il vaut la peine de men-

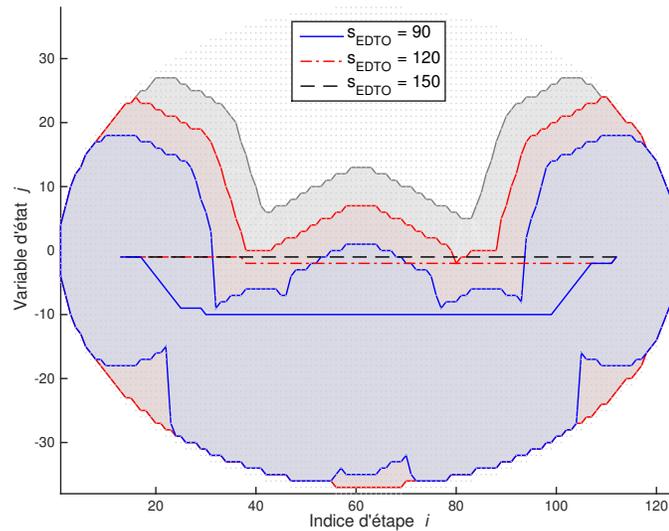


FIGURE 4.5: Comparaison de trajectoires avec trois seuils de l'EDTO différents. $b_{I_A J_A K_A}^2 = 56000$, $CI = 100$, $p = 1$

tionner que l'ajout du dual permet d'éliminer des chemins infaisables, mais sans modifier la solution.

Une illustration du résultat de ce calcul du dual est donné à la figure 4.7. Autour de l'aéroport d'arrivée, la quantité requise pour se rendre à l'arrivée est faible. Les points sont donc colorés en jaune. Autour de l'aéroport de départ, les couleurs bleu et indigo indiquent que la quantité requise dans le réservoir est importante. Notons quelques différences avec la figure 4.4. À la figure 4.7, la contrainte est plus serrée à mesure que l'on s'éloigne de l'aéroport d'arrivée alors qu'à la figure 4.4 la contrainte se resserre lorsqu'on s'éloigne d'un aéroport de dégagement, peu importe lequel. De plus, la zone interdite est plus étendue à la figure 4.7. Elle permet donc d'éviter de visiter un plus grand nombre de nœuds. L'espace a subi deux coupes en diagonales qui se croisent à l'aéroport d'arrivée ainsi qu'une autre coupe au-dessus de l'aéroport de départ. Ces coupes sont le résultat de la configuration des successeurs (présentée à la figure 3.4) qui limite l'amplitude des virages.

La prochaine étape consiste à combiner les contraintes de carburant et le dual pour obtenir en chaque nœud une borne supérieure unique pour la consommation. À chaque nœud, la borne la plus contraignante est choisie : celle qui est prescrite par les règles de carburant ou celle qui est calculée par le plus court chemin. Par exemple, si les règles de carburant requièrent que la consommation maximale ne soit pas de plus de 37 000 kilogrammes (4.1) et le dual, 36 000 kilogrammes, (4.2) alors la valeur de la contrainte sera 36 000 kilogrammes (4.3).

$$R_{ijk}^2(C) \leq \underbrace{R}_{ijk}^2 = 37000 \quad (4.1)$$

$$R_{ijk}^2(C) \leq b_{I_A J_A K_A}^2 - \underline{R}_{ijk}^2 = 36000 \quad (4.2)$$

$$R_{ijk}^2(C) \leq 36000 \quad (4.3)$$

Le résultat de la combinaison se trouve à la figure 4.8. L'influence des contraintes imposées par les règles de carburant est visible autour de l'aéroport d'arrivée, alors que le dual est

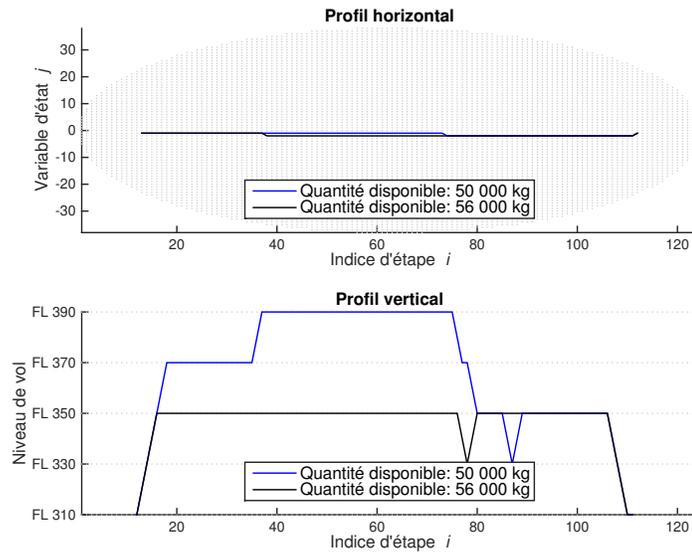


FIGURE 4.6: Comparaison de chemins avec deux quantités de carburant disponibles. $s_{EDTO} = 150$, $CI = 100$, $p = 1$

prépondérant plutôt dans la partie de gauche de l'ellipse. Ce résultat est cohérent avec les éléments présentés à la section 1.4.4 où il était mentionné que les zones à surveiller pour s'assurer de respecter les normes de l'EDTO sont celles se trouvant à la fin du vol entre les aéroports de décollage. C'est ce que l'on peut observer à la figure 4.8 autour de la valeur d'indice d'étape 100 où se situe une petite zone où les points prennent une couleur tendant vers le vert.

La figure 4.9 montre l'impact de l'utilisation de l'élagage par infaisabilité sur le temps de calcul. La figure 4.9a montre la relation entre le nombre de fois où le calcul de consommation $r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_D - R_{i_c j_c k_c}^2(C))$ est requis et le CI. Le dual permet de réduire le nombre de calculs peu importe le CI. Ce nombre de calculs est important pour deux raisons. D'abord, le calcul de $r^2(\cdot)$ en lui-même requiert du temps. Ainsi, une réduction d'environ 180 000 calculs comme on peut l'observer autour de $CI = 120$ peut permettre une réduction intéressante du temps de calcul. Si, par exemple, le calcul de $r^2(\cdot)$ requiert en moyenne 1 μs , la réduction espérée pour le calcul complet de la trajectoire est de 180 ms. Ensuite, les étiquettes générées doivent être comparées par l'algorithme de dominance (voir annexe E). La complexité de cet algorithme est de $\mathcal{O}(m + n)$ où m et n sont la longueur des listes d'étiquettes à comparer. En diminuant le nombre d'étiquettes générées, le temps de calcul requis par l'algorithme de dominance est donc aussi réduit. La figure 4.9b montre que la réduction du nombre de calculs effectués n'a pas d'effet positif sur le temps de calcul, car le temps de calcul du dual dépasse les économies réalisées par l'élagage. Pour certaines valeurs de CI, on peut observer un temps de calcul de presque 0 ms. Le temps de calcul est très faible car l'algorithme ne trouve pas de chemin pour ces valeurs.

La figure 4.10 illustre l'impact de la quantité de carburant disponible sur l'efficacité de l'élagage par infaisabilité. Chaque courbe est associée à une quantité de carburant disponible. Deux phénomènes se produisent : certaines courbes sont incomplètes et la portion d'étiquettes éliminées varie selon la quantité de carburant disponible.

Si trois courbes de la figure 4.10 sont incomplètes, c'est que pour la combinaison donnée des contraintes de carburant, du CI et de la quantité de carburant choisies, aucune trajectoire

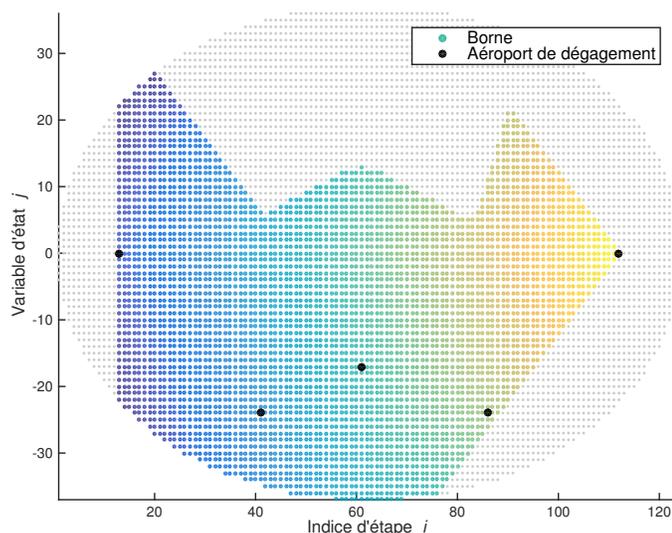


FIGURE 4.7: Bornes supérieures de consommation pour l'élagage par infaisabilité. $s_{EDTO} = 150$

faisable n'existe dans le réseau. Plus la quantité de carburant disponible diminue, plus l'intervalle de valeurs de CI sur lequel il n'y a pas de chemin faisable s'élargit. On peut identifier deux facteurs qui contribuent à ce phénomène. D'une part, le fait de diminuer la quantité de carburant disponible rend le respect des contraintes de carburant plus difficile, car les réserves pour se rendre aux aéroports de décollage sont diminuées. On constate cependant que pour des valeurs faibles de CI, l'algorithme trouve un chemin. Quand le CI est faible, le problème est presque équivalent à minimiser la consommation de carburant sur la trajectoire. Ainsi, la vitesse choisie pour chaque segment sera celle qui minimise la consommation de carburant sur le segment, ce qui assure qu'une trajectoire faisable sera déterminée si elle existe. Dès que le CI commence à augmenter, la vitesse choisie sur les segments augmente, entraînant une augmentation de la consommation de carburant et en conséquence une diminution des réserves. D'autre part, lorsque le CI est faible, l'avion tend à voler à plus haute altitude pour minimiser sa consommation. Or, il a été observé à la section 4.2.2 que les réserves requises à haute altitude sont plus faibles qu'à plus basse altitude. Ainsi, la combinaison d'une quantité de carburant faible, d'une vitesse de croisière requérant une consommation légèrement plus élevée et de réserves requises élevées ne permet pas de trouver une trajectoire faisable.

Lorsque la quantité de carburant disponible est la plus faible, la proportion d'étiquettes éliminées est la plus élevée. Ce phénomène illustre la remarque de Lozano et Medaglia (2013) qui indiquait que lorsque la contrainte de ressource est serrée, l'élagage par infaisabilité est particulièrement avantageux. Avec 50 000 kilogrammes, on peut considérer la contrainte comme serrée, car pour cette configuration d'aéroports de décollage avec $s_{EDTO} = 150$ et $CI = 100$, il faut au minimum 49 000 kilogrammes de carburant pour trouver une trajectoire faisable. Remarquons qu'il en faut peu pour que l'élagage par infaisabilité perde de son efficacité. Avec 53 000 kilogrammes de carburant, la proportion d'étiquettes éliminées chute à environ 20%, soit environ la même proportion que pour une quantité de carburant disponible de 56 000 kilogrammes. Cette chute s'explique par le fait que l'élagage par infaisabilité élimine le plus rapidement les chemins infaisables, mais que ces chemins deviennent de plus en plus rares au fur et à mesure que la quantité de carburant disponible augmente. Cependant, la proportion d'éti-

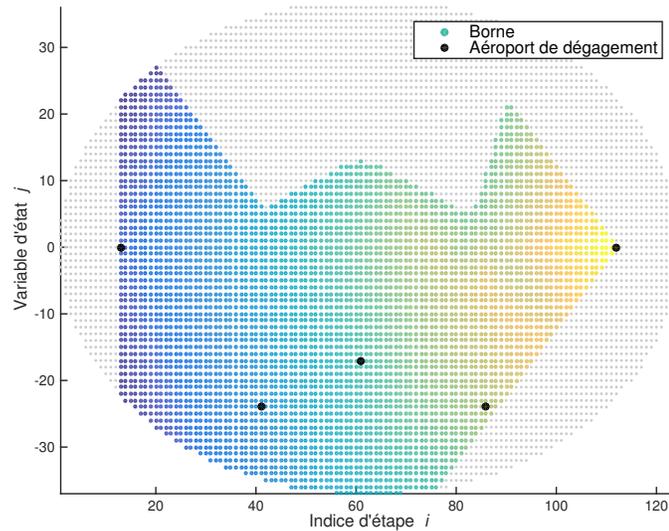


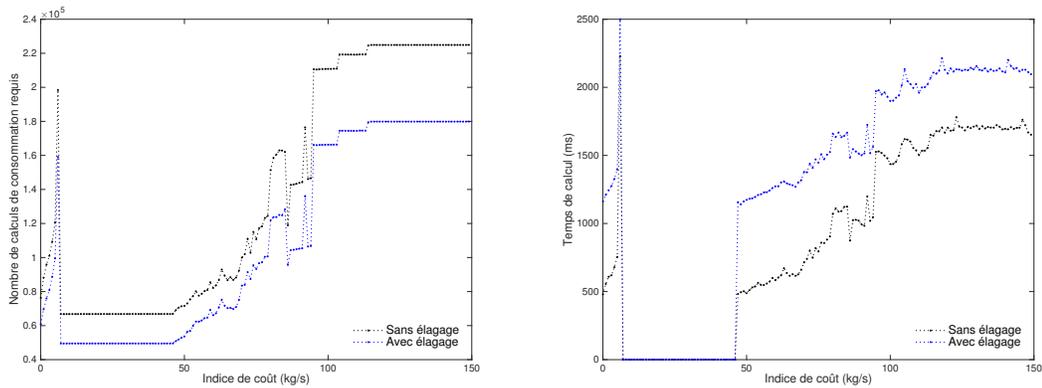
FIGURE 4.8: Bornes supérieures de consommation calculées avec les contraintes de carburant combinées aux bornes de l'élagage par infaisabilité. $SEDTO = 150$, $b_{IAKA}^2 = 56000$

quettes éliminées ne continue pas à diminuer au fur et à mesure que la quantité de carburant disponible augmente.

La figure 4.11 explique partiellement ce phénomène. Chacune des sous-figures montre où sont éliminées les étiquettes pour une quantité de carburant disponible donnée. En 4.11a, les étiquettes éliminées sont associées à des nœuds éloignés de la ligne centrale de l'ellipse. Avec 50 000 kilogrammes disponibles, il y a donc peu, sinon pas, de marge de manœuvre pour trouver un chemin si la trajectoire s'éloigne de la ligne de centre de l'ellipse. À la sous-figure 4.11d, les étiquettes sont éliminées dans une zone limitée par une pointe triangulaire dont le sommet est l'aéroport d'arrivée. Cette zone est en grande partie la même que la zone interdite qui a été présentée à la figure 4.7. Les figures 4.11b et 4.11c montrent le déplacement des étiquettes éliminées du bas de l'ellipse vers la zone interdite à mesure que la quantité de carburant disponible augmente.

Il semble donc qu'une bonne partie de l'amélioration provienne des coupes qui sont réalisées par le dual plutôt que par l'élimination des chemins infaisables. Il s'avère donc que le temps investi dans le calcul de la consommation de carburant dans l'algorithme de l'annexe G est mal investi. Il serait suffisant de seulement connaître les nœuds qui sont visités lors du calcul du dual pour éliminer un grand nombre d'étiquettes. L'algorithme 4.2 réalise ces coupes en éliminant les nœuds qui ne sont pas visités pendant le calcul du dual. Il visite donc les mêmes nœuds que l'algorithme de l'annexe G sans lancer le calcul de consommation, ce qui permet de diminuer considérablement le temps de calcul.

La figure 4.12 permet de comparer sa performance avec celle de l'algorithme sans élagage et de l'algorithme avec l'élagage par le calcul du dual. Le nombre d'étiquettes pour les CI de moins de 60 est plus faible que sans élagage, mais plus élevé que pour l'élagage complet, ce qui est cohérent avec les résultats de la figure 4.11. Au fur et à mesure que le CI augmente, le nombre d'étiquettes se rapproche de celui obtenu par l'élagage par infaisabilité. Du côté du temps de calcul, cet algorithme surpasse tous les autres. Le temps requis pour augmenter la taille de la zone interdite est largement inférieur au temps économisé lors de la résolution de l'algorithme

(a) Nombre de fois où le calcul de $r^2(\cdot)$ est effectué

(b) Temps requis pour le calcul

FIGURE 4.9: Effet de l'élagage par infaisabilité en fonction du CI. $s_{EDTO} = 150$, $b_{IAJAK_A}^2 = 56000$, $p = 1$

de mise à jour, ce qui permet d'obtenir un temps de calcul inférieur à celui obtenu par les deux autres algorithmes dans tous les cas. Cette technique d'élagage pourrait aussi être utilisée pour la planification initiale afin d'éliminer les nœuds qui ne peuvent jamais être atteints.

L'élagage par coupe est plus efficace que l'élagage complet et que le calcul sans élagage même quand la contrainte de carburant est serrée, comme le montre la figure 4.13. Le gain en temps est moins intéressant que pour des quantités de carburant disponibles plus élevées, car les étiquettes se trouvant au centre au bas de l'ellipse à la figure 4.11a ne sont pas éliminées par l'algorithme 4.2.

Cette sous-section a permis de constater que les contraintes de carburant requièrent que l'avion transporte plus de carburant que dans un espace aérien sans contrainte. Cela diminue donc significativement le nombre de trajectoires potentiellement infaisables, ce qui rend l'élagage par infaisabilité pratiquement inutile. Une simple réduction de l'espace permet d'améliorer le temps de calcul. Le lecteur a aussi pu observer que les contraintes calculées introduisent aussi des irrégularités dans les trajectoires, comme des montées et descentes en pleine phase de croisière.

4.2.3 Effet des coûts

Au chapitre précédent, il a été déterminé que le coût serait calculé de la manière suivante :

$$f_{i_c j_c k_c i_s j_s k_s u}^1(R_{i_c j_c k_c}(C)) = R_{i_c j_c k_c}^1(C) + p \cdot \left(r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u) + CI \cdot u \right) + s(i_c, j_c, i_s, j_s) \quad (4.4)$$

S'il n'y a pas de zone tarifée, une augmentation du prix p du carburant n'aura pas d'autres effets d'augmenter le coût de la trajectoire. Effectivement, puisque le CI convertit le temps en kilogrammes de carburant, le prix du carburant n'a pas d'impact sur l'importance du temps par rapport à celle du carburant. On pourrait aussi reformuler (4.4) comme en (4.5). Développer l'équation (4.5) permet d'obtenir (4.6) (où les indices a dénotent un nœud antérieur) ce qui permet de mettre en évidence que c'est le rapport entre le coût d'une zone tarifée et le coût du carburant qui est réellement important. Effectivement, si le coût du carburant augmente,

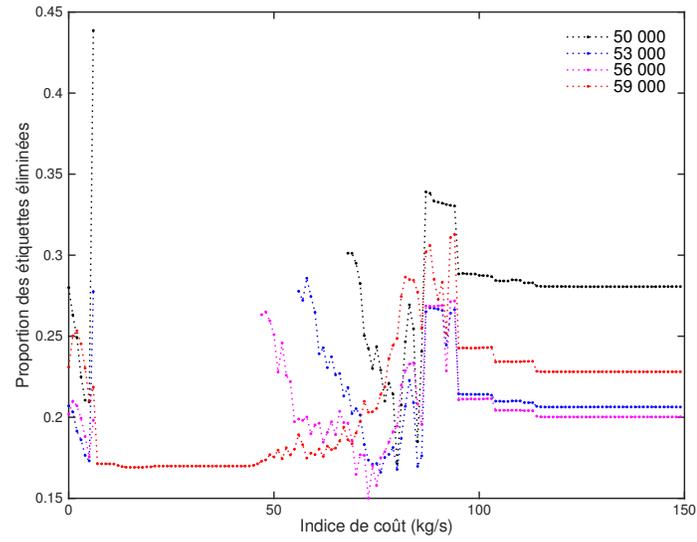


FIGURE 4.10: Proportion des étiquettes éliminées par le dual. $s_{EDTO} = 150$, $p = 1$, $CI = 60$

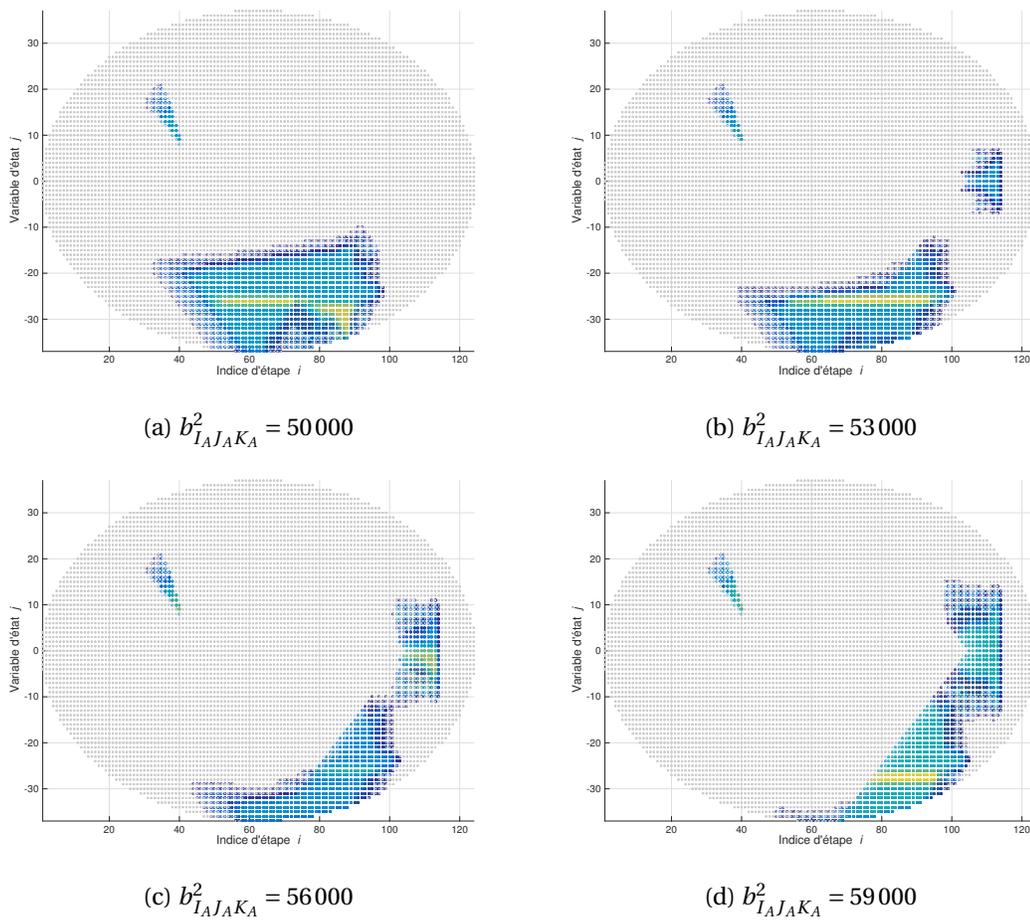
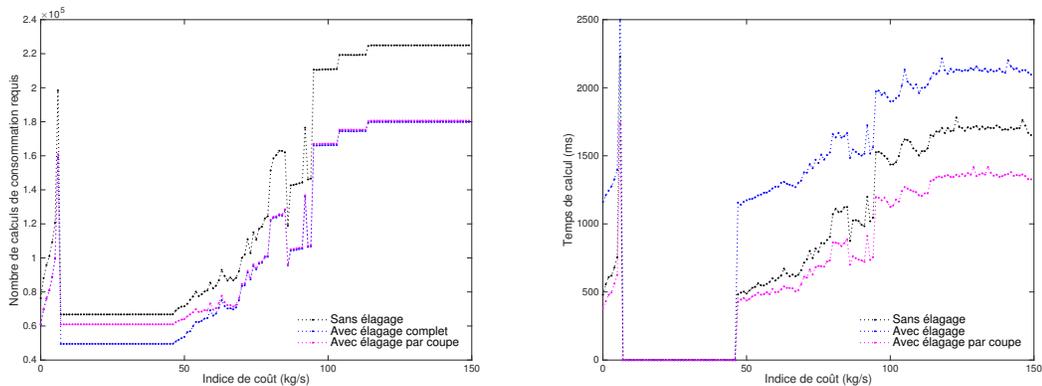


FIGURE 4.11: Impact de l'utilisation de l'élagage par infaisabilité en fonction de la quantité de carburant disponible. $s_{EDTO} = 150$, $CI = 60$, $p = 1$

Algorithme 4.2 Algorithme pour agrandir la zone interdite

- 1: Initialiser (I_A, J_A, K_A)
- 2: $\mathcal{U} \leftarrow (I_A, J_A, K_A)$
- 3: $\mathcal{C} \leftarrow (I_A, J_A, K_A)$
- 4: **tant que** $\mathcal{U} \neq \emptyset$ **faire**
- 5: Choisir un nœud $(i_s, j_s, k_s) \in \mathcal{U}$ dans l'ordre topologique
- 6: $\mathcal{U} = \mathcal{U} - C$
- 7: **pour tout** $\{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E$ **faire**
- 8: **si** $(i_s, j_s, k_s) \in \mathcal{A}(i_c, j_c, k_c)$ **alors**
- 9: $\mathcal{U} = \mathcal{U} \cup (i_c, j_c, k_c)$
- 10: $\mathcal{C} = \mathcal{C} \cup (i_c, j_c, k_c)$
- 11: **fin si**
- 12: **fin pour**
- 13: **fin tant que**
- 14: **pour tout** $((i_c, j_c, k_c) \notin C) \in E$ **faire**
- 15: $b_{i_c j_c k_c}^2 = 0$
- 16: **fin pour**

(a) Nombre de fois où le calcul de $r^2(\cdot)$ est effectué

(b) Temps requis pour le calcul

FIGURE 4.12: Effet de l'élagage par coupe en fonction du CI. $s_{EDTO} = 150$, $b_{I_A J_A K_A}^2 = 56000$, $p = 1$

il devient plus coûteux de contourner la zone tarifée et donc relativement moins coûteux de passer par ladite zone. Si le droit de passage dans une zone tarifée augmente, il vaudra mieux la contourner. Ce serait également le cas si le coût du carburant diminuait. Nous conserverons donc le prix du carburant à 1, mais ferons varier le droit de passage de la zone tarifée

$$\frac{f_{i_c j_c k_c i_s j_s k_s}^1(R_{i_c j_c k_c}(C))}{p} = \frac{R_{i_c j_c k_c}^1(C)}{p} + r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u_2) + CI \cdot u_2 + \frac{s(i_c, j_c, i_s, j_s)}{p} \quad (4.5)$$

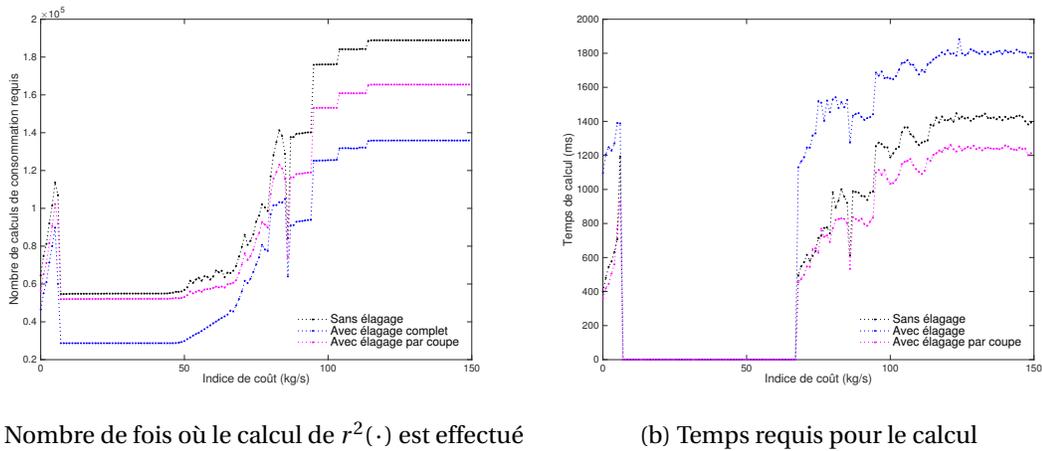


FIGURE 4.13: Effet de l'élagage par coupe en fonction du CI avec peu de carburant disponible. $s_{EDTO} = 150$, $b_{IAJAKA}^2 = 50000$, $p = 1$

$$\frac{f_{i_c j_c k_c i_s j_s k_s u}^1(R_{i_c j_c k_c}(C))}{p} = \frac{R_{i_a j_a k_a}^1(C)}{p} + r^2(i_a, j_a, k_a, i_c, j_c, k_c, R_{i_a j_a k_a}^2(C), u_1) + r^2(i_c, j_c, k_c, i_s, j_s, k_s, R_{i_c j_c k_c}^2(C), u_2) + CI \cdot (u_1 + u_2) + \frac{s(i_p, j_p, i_c, j_c) + s(i_c, j_c, i_s, j_s)}{p} \quad (4.6)$$

La figure 4.14 permet de voir l'effet de la quantité de carburant disponible lorsque vient le temps de décider de contourner ou non une zone tarifée. Lorsque moins de carburant est disponible, le contournement de la zone tarifée peut être problématique. Par exemple, on pourrait décider de la contourner, mais en conséquence devoir diminuer la vitesse de croisière, ce qui allonge la durée du vol, et en augmente le coût. Il est aussi possible que la quantité disponible ne permettent simplement pas le contournement. À la figure 4.14, la zone tarifée est colorée en rouge. Il en coûte 3 000 \$ pour chaque unité de i franchie dans cette zone. Dans la situation présentée à la figure 4.14a, il vaut mieux ne pas contourner la zone et payer le frais de 60 000 \$. Lorsque plus de carburant est disponible, comme à la figure 4.14b, on peut se permettre de s'écarter de la ligne centrale de l'ellipse, de payer le retard engendré par le détour et le carburant brûlé pour éviter les frais, ce qui revient tout de même à environ 58 000 \$.

4.3 Effets de la variation des paramètres

La figure 4.15 permet de voir l'effet de la variation du CI sur le temps de calcul et la consommation de carburant. De manière générale, le temps de calcul est assez faible pour les CI de 50 et moins puis augmente rapidement pour atteindre son maximum autour de CI= 125. Il redescend ensuite et se maintient pour la plupart des essais autour de 1 200 millisecondes. Lorsque le CI est faible, le nombre d'étiquettes est faible, car pour chacune, le coût du segment est environ égal à la consommation de carburant. Ainsi, si le coût d'une étiquette est plus élevé que celui d'une deuxième étiquette, la consommation l'est sans doute aussi. La deuxième étiquette domine alors la première, qui est éliminée. Lorsque le CI, et donc le coût du temps, augmente, cette

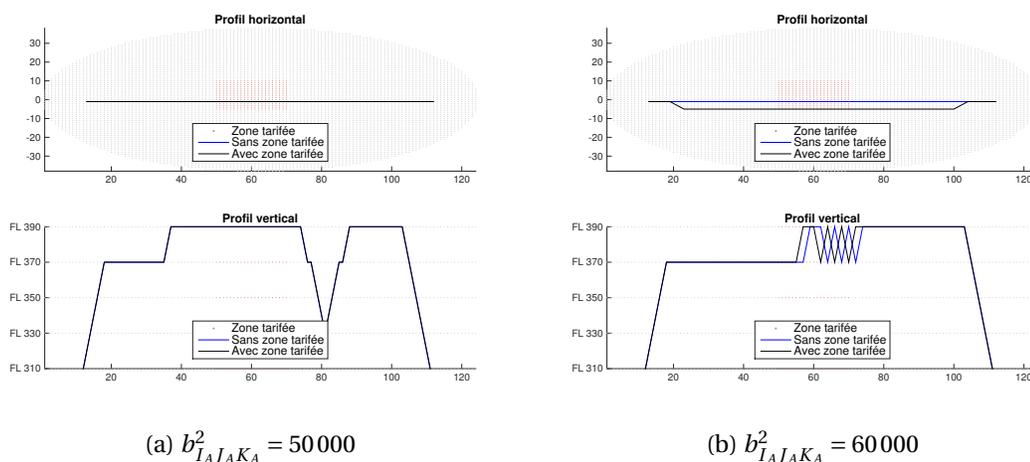
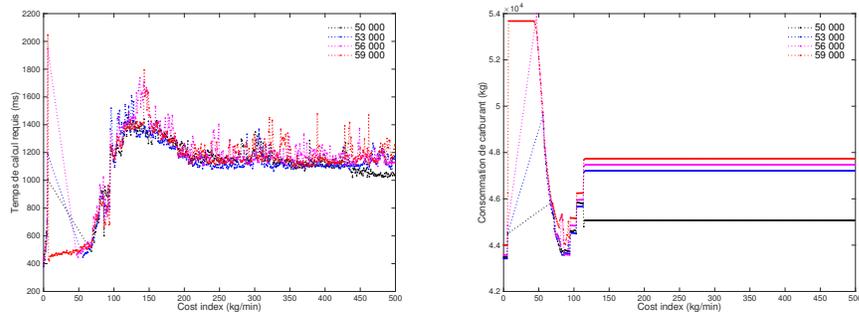


FIGURE 4.14: Effet de la zone tarifée en fonction du carburant disponible. $s_{EDTO} = 150$, $CI = 80$, $p = 1$

dominance s'estompe : une étiquette peut avoir un coût élevé et une consommation faible, ce qui ne permet pas d'éliminer les étiquettes concurrentes. On a donc en chaque nœud plusieurs choix de chemin et l'évaluation de tous ces chemins requiert du temps. Finalement, lorsque le temps devient réellement dispendieux, une petite différence de temps de vol produira une grande différence de coût, ce qui permet d'éliminer quelques étiquettes et de réduire légèrement le temps de calcul. De toute évidence, le CI affecte aussi la consommation de carburant, puisque la vitesse de l'avion est ajustée en fonction du coût du temps. La vitesse de l'avion étant limitée, au-delà d'un certain CI, même si celui-ci continue d'augmenter, la consommation de l'avion reste stable, comme on peut le constater à la figure 4.15b. Finalement, la figure 4.15c montre que lorsque le CI augmente, le coût du vol augmente de manière proportionnelle. Effectivement, lorsque la consommation se stabilise, l'heure d'atterrissage reste la même, mais chaque seconde du vol a un coût plus élevé.

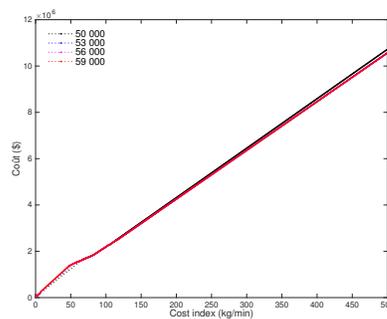
Pour donner une perspective différente, l'impact de la variation de la quantité de carburant disponible sera analysé. Du côté du temps de calcul, on observe entre les différentes courbes la même tendance qu'à la figure 4.15a. L'impact de la quantité de carburant est assez faible en comparaison avec celui du CI, sauf si le CI se trouve autour de 80, alors le temps de calcul diminue lorsque la quantité de carburant augmente. Lorsqu'on s'intéresse à la consommation de carburant, on remarque que certains phénomènes déjà observés sont présents. D'abord, lorsque le CI est faible, la quantité de carburant consommée augmente avec la quantité disponible, car la masse de l'avion est augmentée. C'est lorsque le CI dépasse 100 que la consommation de carburant a réellement un impact. Aussitôt qu'on rend plus de 51 000 kilogrammes de carburant disponible, l'avion change de trajectoire comme en témoigne l'augmentation rapide de la consommation. Ce changement de trajectoire a été analysé plus en détail à la section 4.1. Finalement, comme attendu, le coût du vol augmente aussi lorsque plus de carburant est disponible, car du carburant est brûlé pour transporter le carburant en surplus.

Dans chacune des trois figures, la quantité minimale de carburant requis pour trouver un chemin est de 49 000 kilogrammes, soit près de 7 000 kilogrammes de plus que pour un chemin qui ne tient pas compte des contraintes de carburant comme à la section 4.1. Les kilogrammes supplémentaires ne sont jamais dépensés complètement, comme le montre la figure 4.16b. Le



(a) Temps requis pour le calcul

(b) Consommation de carburant de la trajectoire optimale



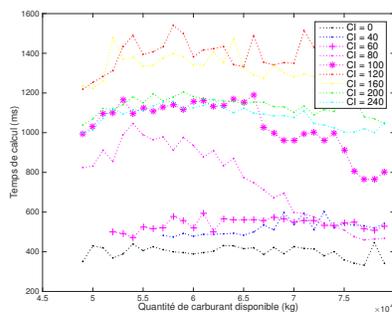
(c) Coût de la trajectoire optimale

FIGURE 4.15: Effet de la variation du CI sur le calcul du RCSP. $s_{EDTO} = 150$, $p = 1$

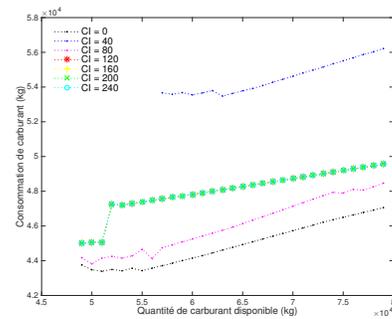
coût des contraintes de carburant est donc ce qu'il en coûte en carburant et en temps pour transporter ces kilogrammes.

Examinons l'impact de la finesse de la grille. La figure 4.17a montre l'évolution du temps de calcul en fonction du CI. Chaque courbe est associée à une valeur de la légende. Cette valeur indique la proportion d'un carré d'un degré de côté qui est représentée par un nœud du réseau. Pour obtenir les données précédentes, un carré d'un demi-degré de côté et la courbe qui lui est associée est celle portant la valeur 0,25 dans la légende. Chaque nœud représente donc un carré dont le côté mesure un demi-degré. Tel qu'on s'y attend, plus la précision de la grille est élevée, plus le temps de calcul est long. On remarque aussi que si la surface du carré représentée est de moins de 0,25, l'algorithme trouve un chemin pour CI= 65. Pour les valeurs plus élevées, le CI doit être d'au moins 70.

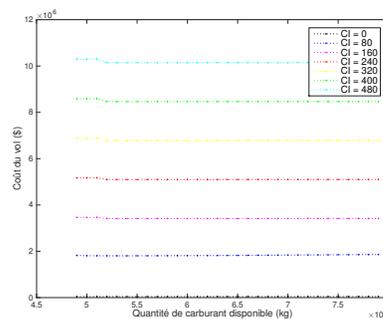
La figure 4.18 montre l'évolution du temps de calcul en fonction de la précision de la grille de calcul. Pour chaque combinaison de carburant disponible et de CI, le temps de calcul pour les différentes précisions de la grille est exprimé comme une proportion du temps de calcul de la trajectoire dans la grille ayant la précision la plus élevée. Par exemple, si le temps de calcul pour la précision la plus élevée est de 4 000 millisecondes et qu'il est de 800 millisecondes pour une précision plus faible, la proportion est 0,2. Les croix rouges indiquent la proportion moyenne du temps de calcul par rapport au temps le plus long pour les cas où l'algorithme trouve une trajectoire complète. Le temps de calcul augmente rapidement avec la précision. Dans le cas à l'étude, un vol Montréal-Paris, le temps de calcul se situe toujours à moins de 5000 millisecondes, mais si on cherchait à trouver une trajectoire reliant Montréal et Pékin, par exemple, la précision de la grille pourrait avoir un impact sérieux sur le temps de calcul. En uti-



(a) Temps requis pour le calcul



(b) Consommation de carburant de la trajectoire optimale



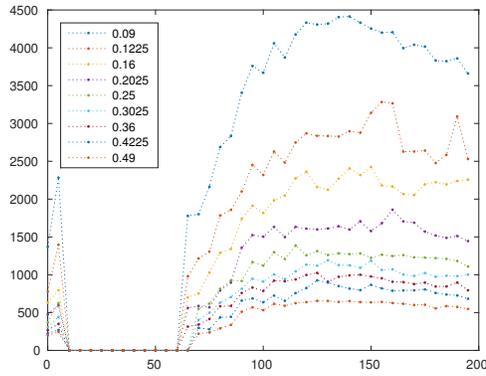
(c) Coût de la trajectoire optimale

FIGURE 4.16: Effet de la variation de la quantité de carburant disponible sur le calcul du RCSP. $s_{EDTO} = 150$, $p = 1$

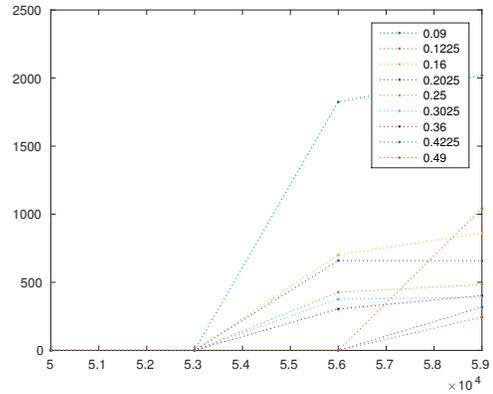
lisant un carré de 0,3 degré de côté, le temps de calcul moyen est près de cinq fois le temps de calcul moyen pour un carré de 0,6 degré de côté. Diminuer la longueur des arcs du réseau par 2 multiplie donc le temps de calcul moyen par un facteur plus élevé que 4.

La figure 4.17b montre que tout dépendant de la précision, lorsque $CI = 45$, la quantité de carburant requise pour trouver un chemin n'est pas toujours la même. Dans la grille la plus fine, un chemin est trouvé. Il semble que, si le nœud représente une proportion de 0,16 et de 0,36 d'un carré d'un degré de côté, l'algorithme trouve un chemin presque aussi souvent que la précision la plus élevée, peu importe le CI et la quantité de carburant disponible. Une précision faible (un nœud représente une proportion de 0,42 du carré de côté mesurant un degré) rend la recherche de chemin difficile, comme si les mailles du réseau étaient si larges que l'algorithme ne dispose pas de suffisamment de choix et ne peut trouver un chemin qui respecte les contraintes. Il semble que les précisions associées aux valeurs 0,4225 et 0,49 degré de côté soient inappropriées pour répondre aux objectifs. Celles-ci sont donc éliminées pour les tests ultérieurs.

La figure 4.19 montre la consommation de carburant en fonction du CI . Selon la précision, pour un même CI , la différence de consommation peut être de plus de 2 000 kilogrammes de carburant. Les proportions 0,16 et 0,36 se démarquent à nouveau du lot par leur comportement similaire. On peut dire que la précision la plus élevée permet de calculer des consommations qui ne suivent pas la moyenne des consommations toutes précisions confondues, mais sans données de vol pour les comparer, il est difficile de savoir si c'est parce qu'elles sont réellement



(a) En fonction du CI. $b^2_{IAJA KA} = 50000$



(b) En fonction de la quantité de carburant disponible. CI= 45

FIGURE 4.17: Effet de la variation de la précision de la grille sur le temps de calcul requis. $s_{EDTO} = 150, p = 1$

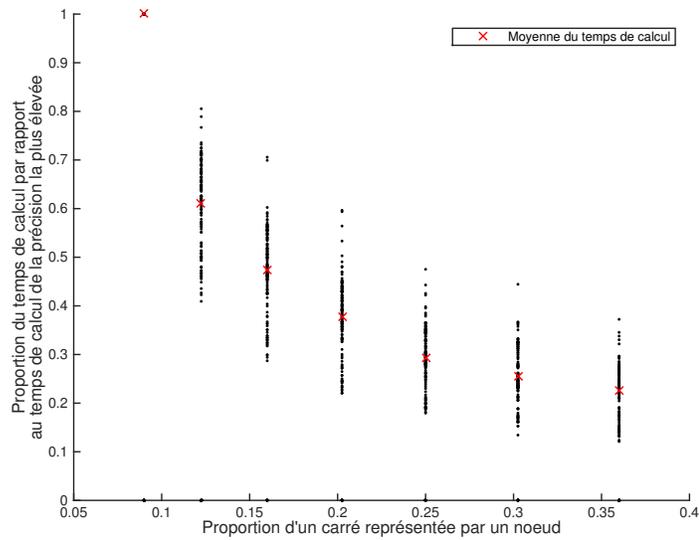


FIGURE 4.18: Temps de calcul en fonction de la précision de la grille. $s_{EDTO} = 150, p = 1$

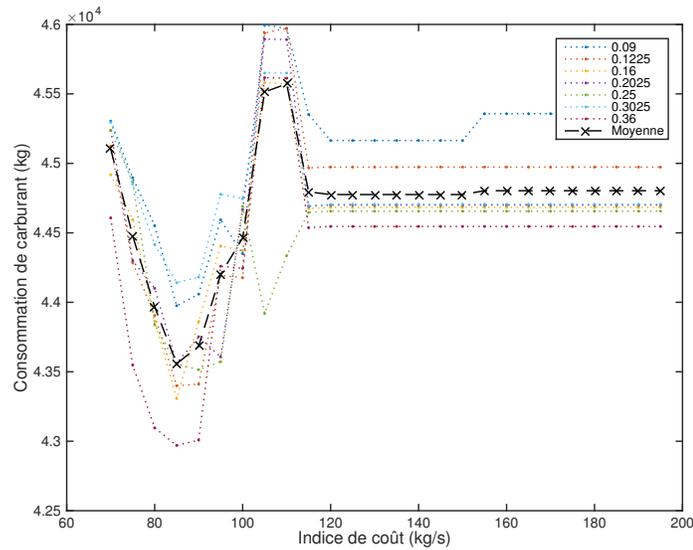


FIGURE 4.19: Consommation de carburant en fonction du CI dans des grilles de précisions différentes. $b_{IJA}^2 = 50000$, $s_{EDTO} = 150$, $p = 1$

plus précises ou si c'est parce que des erreurs numériques se présentent lorsque la trajectoire est composée d'une grande quantité d'arcs.

La figure 4.21 compare les différentes trajectoires générées. La trajectoire attendue pour les conditions de tests est une montée dès le début du vol et une descente qui se termine en même temps que le vol. Aucune des trajectoires ne suit exactement la trajectoire attendue. Deux éléments distinguent les trajectoires les unes des autres. Premièrement, la montée et la descente ne se font pas nécessairement au même moment. Les contraintes de carburant sont très peu serrées autour des aéroports, donc la différence doit provenir d'ailleurs. Examinons comment les descentes et les montées sont réalisées. La figure 4.20 montre comment les deux types d'arcs sont calculés. D'abord, l'avion accélère (décélère) puis exécute la montée (descente). L'avion vole ensuite à vitesse constante pour parcourir le reste de la distance au sol. Dans chacun, l'accélération (décélération) et la longueur du segment à vitesse constante vont varier la consommation de carburant. De plus, si l'accélération ou la montée ou les deux combinées requièrent une plus grande distance que la longueur de l'arc, l'arc est éliminé.

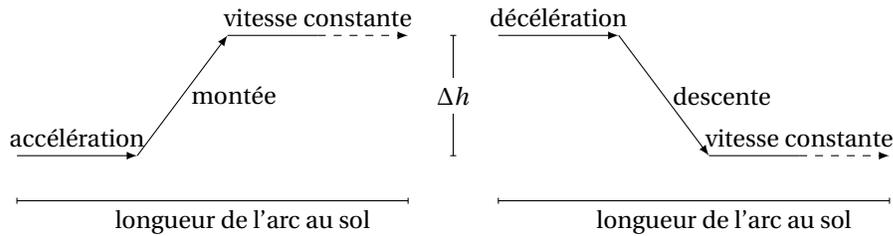


FIGURE 4.20: Illustration des manœuvres de montée et de descente

Le tableau 4.4 indique la longueur de l'arc au sol pour chaque précision. Peu importe la précision choisie, la hauteur de la montée (Δh) reste la même, car elle est déterminée par la réglementation (voir section 1.4.2). En revanche, la longueur de l'arc au sol varie en fonction de

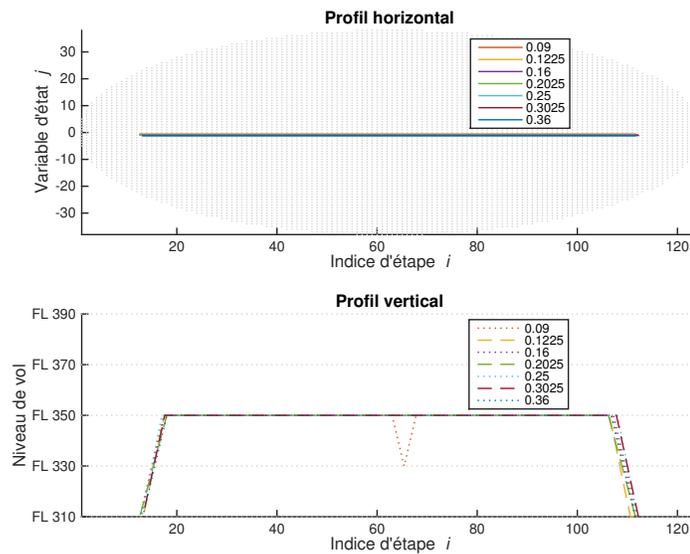


FIGURE 4.21: Comparaison de trajectoires dans des grilles de précisions différentes. $CI = 100$, $b_{IAJAK_A}^2 = 56000$, $s_{EDTO} = 150$

TABLEAU 4.4: Longueur du segment en fonction de la précision de la grille

Proportion du carré	0,09	0,1225	0,16	0,2025	0,25	0,3025	0,36
Δk	1						
Δi	4	3	3	3	2	2	2
Longueur de l'arc au sol (degré)	1,2	1,05	1,2	1,35	1	1,1	1,2

la précision et n'est pas exactement proportionnelle à celle-ci. La longueur est la même pour les nœuds représentant une proportion de 0,16 et de 0,36 du carré, ce qui pourrait expliquer pourquoi les trajectoires générées dans ces grilles se ressemblent. Cette variation dans la longueur de l'arc au sol pourrait expliquer pourquoi les montées et descentes ne sont pas toujours exécutées au même moment. Elles font varier le coût d'un arc de montée pour une raison étrangère à la dynamique de l'avion.

Deuxièmement, la trajectoire générée dans la grille avec la précision la plus élevée comporte une descente et une montée en milieu de croisière. C'est un exemple de l'effet des lacunes du calcul des réserves de carburant. Cette descente compense une augmentation abrupte des réserves de carburant et cette augmentation ne se trouve que dans la grille avec la précision la plus élevée.

Les résultats présentés dans cette sous-section permettent d'établir que le temps de calcul est influencé largement par la précision de la grille et en partie par le CI utilisé. La quantité de carburant disponible a peu d'influence, sauf pour quelques exceptions. La quantité de carburant disponible affecte la consommation, car elle la limite. Le CI influence aussi la consommation en faisant varier la vitesse et l'altitude de croisière. Sans données de vol avec lesquelles comparer les trajectoires générées, il est difficile de dire quelle précision devrait être utilisée. Puisque son influence est importante sur le temps de calcul, elle pourrait être ajustée en fonction de l'application. Par exemple, si plusieurs milliers de trajectoires doivent être planifiées, une précision plus faible pourrait être utilisée pour diminuer le temps de calcul. Cependant,

chaque nœud devrait représenter tout au plus un carré de 0,6 degré de côté, car une précision plus faible ne permet pas de trouver une solution avec une qualité satisfaisante.

La première et la deuxième partie de ce chapitre ont permis de montrer que l'algorithme de mise à jour trouvera une ou plusieurs trajectoires pour une quantité de carburant donnée si une solution existe. La différence entre les multiples trajectoires calculées, s'il y a lieu, a été expliquée. Ensuite, l'effet des contraintes de carburant a été exploré.

Conclusion

En conclusion, l'algorithme de résolution proposé pour le problème de la mise à jour permet de trouver assez rapidement une trajectoire avec une quantité de carburant limitée, si une trajectoire existe. Dans un premier temps, la description du problème a permis d'établir les coûts à prendre en compte : le coût du carburant, le coût du temps et le coût de survol des zones tarifées. Les contraintes considérées sont les règles de l'espace aérien, qui régissent la variation permise de vitesse, les altitudes permises, la quantité de carburant à emporter, l'éloignement des aéroports de décollage et les zones interdites. La dynamique de l'avion limite le nombre de triplets vitesse-altitude-masse possibles. Les limitations sur les altitudes et les virages ont été intégrées dans la construction de la grille. À chaque nœud, une contrainte sur le carburant permet de tenir compte de la quantité de carburant à emporter, des aéroports de décollage et des zones interdites. La revue de littérature a permis d'identifier la programmation dynamique comme une approche détenant beaucoup de potentiel pour résoudre le problème de la mise à jour de plan de vol. La formulation mathématique utilisée, le plus court chemin avec contraintes de ressource, permet d'optimiser une trajectoire en 3D en tenant compte de la météo, des zones tarifées et du coût du temps. Parmi les méthodes de mise à jour disponibles au public, c'est, à notre connaissance, le seul algorithme qui possède cette capacité. Pour pouvoir calculer une trajectoire tenant compte des contraintes de carburant liées à la réglementation, un algorithme qui calcule de telles réserves a été développé. Ce calcul, de même que la stratégie utilisée pour accélérer le calcul de la mise à jour, peuvent être utilisés par les membres de l'équipe qui travaillent sur le problème de la planification initiale. Le temps de calcul est influencé surtout par la précision de la grille, mais aussi par le coût du temps choisi. Une trajectoire reliant Montréal et Paris peut être calculée en moins de cinq secondes pour toutes les précisions testées. La précision de la grille affecte le calcul de la consommation de carburant, notamment en raison de la gestion des montées et des descentes.

Le modèle proposé a ses limites. En termes de résultats, on a pu constater que le calcul des réserves réglementaires introduit beaucoup de montées et de descentes dans les trajectoires. En pratique, pour que la trajectoire soit réalisable elle doit être plus lisse. La solution simple est d'ajouter un coût artificiel à une trajectoire lorsqu'une montée ou une descente est effectuée, mais cette solution aurait d'autres impacts que le lissage de la trajectoire, ce qui n'est pas nécessairement désirable. Une autre approche serait d'améliorer le calcul des réserves de carburant pour éviter d'introduire des montées et descentes inutiles. En outre, les tests faisant varier la précision de la grille ont permis de mettre en lumière des anomalies lors des montées et des descentes. En ajustant la précision ou la longueur des arcs ou les deux, on pourrait estomper ces anomalies. Concernant la justesse de la solution, puisque la méthode a été développée dans un espace discret, pour un CI donné, l'algorithme choisit le chemin optimal à travers le réseau. Ce chemin optimal n'est probablement pas la trajectoire optimale dans l'espace puisque la trajectoire doit passer par des points déterminés à l'avance. Girardet (2014) propose une manière de contourner le problème de l'espace discret en interpolant la trajectoire entre les nœuds

du réseau. Puisque dans l'algorithme présenté plusieurs consommations peuvent être stockées dans chaque nœud, du travail supplémentaire est requis pour que la méthode puisse être utilisée pour avec l'algorithme présenté dans ce mémoire. En outre, l'algorithme de mise à jour est n'est pas compatible avec n'importe quel type de modélisation de l'avion. L'algorithme proposé fonctionne avec un modèle qui a lui-même ses limites. Seules trois types de manœuvres sont tenues en compte : le segment direct, la montée et la descente. Le cap de l'avion et le roulis ne sont pas tenus en compte. Si le modèle de l'avion était modifié pour ajouter des paramètres, par exemple le cap de l'avion, le modèle de mise à jour pourrait ne pas pouvoir en tenir compte.

Afin que l'algorithme puisse être utilisé dans une application de contrôle aérien, il vaudrait sans doute la peine de travailler à réduire le temps de calcul encore davantage. Les résultats ont permis de déterminer que la réduction de la grille de calcul est efficace, mais que l'élagage par infaisabilité ne l'est pas. La prochaine étape pourrait consister à voir si l'on peut utiliser un chemin déjà planifié, par exemple lors de la planification initiale, pour réaliser une meilleure initialisation de la grille (voir Bertsekas (1998)) ou un élagage par borne sur le coût du vol. Un modèle probabiliste permettrait d'ouvrir un grand nombre de possibilités puisque l'environnement dans lequel évolue l'avion change constamment. Par exemple, Delgado et al. (2013) prennent en compte la probabilité que l'heure d'atterrissage d'un vol soit modifiée et suggèrent une stratégie en conséquence. Nilim et El Ghaoui (2004) et Yoon et al. (2012) proposent des algorithmes pour contourner des obstacles météo qui se dissipent à un moment inconnu. Afin de décider quelle option étudier en premier, il serait utile de soumettre les trajectoires générées à des professionnels du mode de l'aviation. Ceux-ci pourraient évaluer si les trajectoires générées par l'algorithme de mise à jour sont faisables ou si des fonctionnalités importantes ont besoin d'être ajoutées.

Annexes

Annexe A

International Standard Atmosphere

L'approximation utilisée par l'OACI (1993) consiste à diviser l'espace aérien en huit et à attribuer des caractéristiques à chacune des divisions. Ainsi, les caractéristiques de l'ISA varient avec l'altitude. L'espace aérien n'est pas contrôlé pour des altitudes de plus de 18000 mètres (FAA, 2014), donc nous nous en tiendrons aux altitudes de moins de 20000 mètres. Les données utilisées seront donc celles des divisions qui concernent les altitudes entre 0 et 11000 mètres et entre 11000 et 20000 mètres.

L'air qui constitue l'ISA est considéré comme un gaz parfait, les équations des gaz parfaits sont donc la base des calculs de l'ISA.

En utilisant une altitude en mètres et une température en degrés Kelvin, on peut utiliser la formule générale (voir (A.1)) en utilisant les paramètres de la limite inférieure de la division observée (notés i). Les paramètres notés 0 sont ceux du niveau de la mer et sont présentés à la table A.1.

$$T = T_i + \beta \frac{(h - h_i)}{1000} \quad (\text{A.1})$$

Les données de OACI 1993, permettent de calculer la température comme en (A.2) pour les altitudes sous 11000 mètres et comme en (A.3) sinon.

$$T_{<11000} = T_0 - 6,5 \frac{h}{1000} \quad (\text{A.2})$$

$$T_{>11000} = T_0 - 6,5 \frac{11000}{1000} - 0 \frac{(h - 11000)}{1000} = 216,65 \quad (\text{A.3})$$

Le même principe est utilisé pour calculer la pression, dont la formule générale, en admettant une variation linéaire de la température, est donnée en (A.4) pour $\beta \neq 0$ et en (A.5) pour $\beta = 0$.

$$p = p_i \left(1 + \frac{\beta}{T_i} \frac{(h - h_i)}{1000} \right)^{-\frac{g_0}{\beta R}}, \beta \neq 0 \quad (\text{A.4})$$

$$p = p_i e^{-\frac{g_0}{TR} \frac{(h - h_i)}{1000}}, \beta = 0 \quad (\text{A.5})$$

À nouveau, on utilise les données de OACI 1993 pour les calculs. Si l'altitude est de moins de 11000 mètres, on calcule la pression comme en (A.6), sinon on la calcule comme en (A.7).

$$p = p_0 \left(1 - \frac{6,5}{T_0} \frac{h}{1000} \right)^{\frac{g_0 x_0}{6,5R}} \quad (\text{A.6})$$

$$p = p_i e^{-\frac{g_0}{TR} \frac{(h-11000)}{1000}} = p_0 \left(1 - \frac{71,5}{T_0}\right)^{\frac{g_0}{6,5R}} e^{-\frac{g_0}{(T_0-71,5)R} \frac{(h-11000)}{1000}} \quad (\text{A.7})$$

Avec la pression et la température calculées grâce aux équations (A.2) et (A.6) ou (A.3) et (A.7), on peut calculer le paramètre restant grâce à (A.8)

$$\rho = \frac{p}{RT} \quad (\text{A.8})$$

Les paramètres au niveau de la mer sont présentés au tableau A.1.

TABLEAU A.1: Valeurs des paramètres pour les calculs de l'ISA (Source : OACI, 1993)

paramètre	valeur	unité
g_0	9,806	m/s ²
p_0	1 013,25	Pa
T_0	288,15	K
R	287,05	J/(K · kg)

Annexe B

Calcul de la consommation de carburant

Les équations du calcul de la consommation de carburant sont tirées de Maazoun (2015). Un calcul numérique détermine une solution aux équations différentielles de la dynamique du vol. La fonction $r^2(i, j, k, x, y, z, R_i^2(C), u)$ qui effectue ce calcul a été développée par Abderrazak Moutassim et Charles Pestieau. Les variables utilisées dans cette section sont décrites brièvement aux annexes C et D.

Le principe à la base du calcul est que la seule variation de la masse de l'avion est celle engendrée par la combustion du carburant. Ainsi, si on nomme le débit massique - taux de consommation de carburant - FF , on sait que

$$\frac{dm}{dt} = -FF \quad (\text{B.1})$$

avec

- m , la masse de l'avion (kg) ;
- t , le temps (s).

Le débit massique en croisière est à la base de tous les calculs de consommation de carburant. Ce débit massique est fonction de la poussée T que fournissent les moteurs. Cette poussée peut être calculée à l'aide de l'équation (B.2) qui dérive de la seconde loi de Newton et des principes d'aérodynamique.

$$(T \cos \alpha - D)V = mg \frac{dh}{dt} + mV \frac{dV}{dt} \quad (\text{B.2})$$

avec

- T , la poussée longitudinale (parallèle à la trajectoire) (N) ;
- α , l'angle d'incidence (radian) ;
- D , traînée aérodynamique (N) ;
- V , la vitesse de l'avion par rapport à l'air (m/s) ;
- h , l'altitude (m) ;
- g , la constante gravitationnelle (9,806 m/s²).

La traînée D est donnée en B.3.

$$D = \frac{1}{2} C_D \rho V^2 S \quad (\text{B.3})$$

où S est la surface des ailes (m²).

Le coefficient de traînée C_D est donné par

$$C_D = (C_{D0} + C_{D2}C_L^2)(1 + C_{M16}M^{16}) \quad (\text{B.4})$$

où C_{D0} , C_{D2} et C_{M16} sont des coefficients propres à l'avion (détaillés à l'annexe D) et M , le nombre de Mach (sans unités).

Le coefficient de portance C_L peut être approximé par

$$C_L = \frac{2\pi}{1 + 2/\lambda}(\alpha - \alpha_0) \quad (\text{B.5})$$

$$\lambda = \frac{b^2}{S} \quad (\text{B.6})$$

où b est l'envergure des ailes (m).

L'angle d'incidence α peut être calculé avec la masse, la vitesse et l'altitude de l'avion (B.8).

$$\alpha = \frac{mg(1 + 2/\lambda)}{\pi\rho V^2 S} + \alpha_0 \quad (\text{B.7})$$

$$\alpha_0 = -\frac{C_{L0}}{C_{L\alpha}} \quad (\text{B.8})$$

Par des méthodes d'intégration numérique, on obtient donc la variation de la masse de l'avion en fonction du temps.

Le programme de Charles Pestieau et Abderrazak Moutassim utilise une méthode de Runge-Kutta d'ordre 4 pour résoudre ces équations.

Annexe C

Notions d'aéronautique

Les quatre forces qui agissent sur l'avion sont la traînée, la portance, la poussée et le poids (voir figure C.1). Le poids ne dépend que de la masse de l'avion. La poussée est fournie par les moteurs. La portance dépend des caractéristiques des ailes, de la vitesse de l'avion et de la densité de l'air. La portance dépend entre autres de l'angle d'incidence α (voir figure C.2), de la surface S et de l'envergure b des ailes (voir figure C.4). En modifiant la portance de l'une des deux ailes, on peut faire varier l'angle de roulis ϕ (voir figure C.3). La traînée a plusieurs composantes. Elle est causée entre autres par le frottement de l'air sur l'avion et les tourbillons induits par la portance

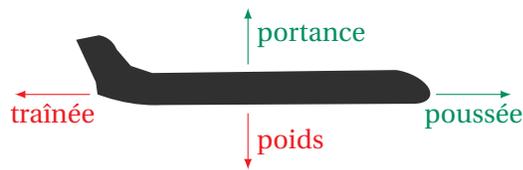


FIGURE C.1: Diagramme de corps libre de l'avion



FIGURE C.2: Angle d'incidence



FIGURE C.3: Angle de roulis

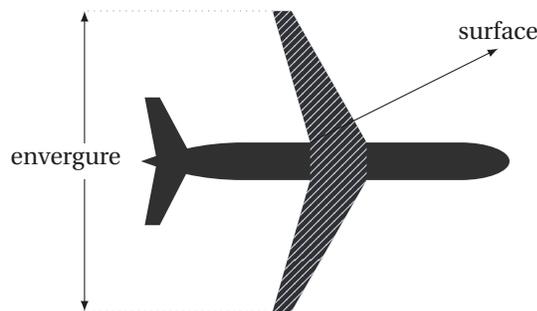


FIGURE C.4: Surface et envergure des ailes d'un avion

Annexe D

Modèle BADA et paramètres

Le modèle BADA d'Eurocontrol utilise des coefficients pour modéliser les différentes forces appliquées sur l'avion (introduites à l'annexe C). Les coefficients utilisés dans le modèle de consommation utilisé pour ce projet sont décrits plus bas.

C_D : coefficient de traînée (1/s)

C_{D0} : coefficient de traînée à portance nulle (1/s)

C_{D2} : coefficient de traînée induite par la portance (1/s)

C_L : coefficient de portance (sans unité)

C_{M16} : coefficient de traînée due à la compressibilité (sans unité)

C_{f1} : coefficient de carburant (kg/sN)

C_{f2} : coefficient de carburant (kg/sN)

C_{f3} : coefficient de carburant (m/s)

C_{f4} : coefficient de carburant (kg/s)

C_{fcr} : coefficient de correction du débit massique en croisière (sans unité)

Annexe E

Algorithme de dominance

On possède une liste d'étiquettes A à insérer dans une liste B . Toutes les étiquettes doivent être non inférieures. Les étiquettes de la liste B respectent l'intervalle $[a_i, b_i]$ du nœud. Les deux listes sont classées en ordre croissant de coût, donc la consommation de ressource est en ordre décroissant (voir exemples aux tables a et b de la figure E.1). Le résultat de l'algorithme de dominance pour ces deux listes la figure est la table c de la figure E.1.

coût	ressource	coût	ressource	coût	ressource
1	8	4	9	1	8
3	7	6	6	3	7
8	4	9	5	6	6
10	3	10	2	8	4
15	1			10	2

(a) Liste d'étiquettes A

(b) Liste d'étiquettes B

(c) Liste résultat

FIGURE E.1: Entrées et résultat de l'algorithme de dominance

L'algorithme débute en haut de la liste A . Il examine la consommation de ressource de la première étiquette. Si elle dépasse la consommation maximale b_i , l'algorithme passe à la prochaine étiquette jusqu'à trouver une étiquette dont la consommation de ressource est au plus b_i . Il compare donc cette étiquette de la liste A avec la première étiquette de B .

Si l'étiquette de la liste A domine l'étiquette de la liste B , l'étiquette de la liste B est *éliminée*. L'étiquette de la liste A n'est pas insérée tout de suite, car elle pourrait dominer la prochaine étiquette de la liste B . Elle doit donc être comparée à nouveau avant d'être insérée. L'algorithme passe ainsi à la prochaine étiquette de la liste B . Si les deux étiquettes sont non inférieures et que le coût de l'étiquette de la liste A est inférieur à celui de l'étiquette de la liste B , l'étiquette de A est *insérée* avant l'étiquette de B . L'algorithme passe à la prochaine étiquette de la liste A . Si les deux étiquettes sont non inférieures, mais que celle de la liste B est la moins coûteuse, l'algorithme passe à la prochaine étiquette de la liste B . L'étiquette de la liste A n'est pas insérée, car la liste doit être classée en ordre croissant de coût et le coût de la prochaine étiquette de la liste B est inconnu. Si l'étiquette de B domine l'étiquette de A , l'algorithme passe à la prochaine étiquette de la liste A . L'étiquette de A est *sans effet*.

Dès que la consommation de ressource d'une étiquette de la liste A tombe sous le seuil a_i , l'algorithme cesse de parcourir la liste A et ajoute les étiquettes restantes de la liste B à la fin de la liste créée.

Algorithme E.1 Algorithme de dominance pour RCSP à un coût et une ressource

Require: La liste B de longueur m qui est contenue dans le nœud satisfait l'intervalle de ressource $[a, b]$ au nœud. Les listes A et B ne contiennent que des étiquettes non inférieures et sont classées en ordre croissant de coût. A est de longueur n . Elle possède n_a étiquettes $\leq a$ et n_b étiquettes $\geq b$

```

1:  $l_A = 1$            ▷ La variable indique à quelle ligne du tableau d'étiquettes A on se trouve
2:  $l_B = 1$            ▷ La variable indique à quelle ligne du tableau d'étiquettes B on se trouve
3:           ▷ Ignorer toutes les étiquettes de A qui dépassent la consommation maximale [ $\mathcal{O}(n_a)$ ] :
4: tant que consommation de  $A[l_A] > b$  faire
5:      $l_A = l_A + 1$ 
6: fin tant que
7:   ▷ Vérifier la dominance des étiquettes tant qu'on peut les comparer [ $\mathcal{O}(n - n_a - n_b + m)$ ] :
8: tant que fin des listes A et B pas atteintes ET consommation de  $A[l_A] \geq a$  faire
9:   si coût de  $A[l_A] <$  coût de  $B[l_B]$  alors
10:    si consommation de  $A[l_A] \leq$  consommation de  $B[l_B]$  alors
11:      ▷ L'étiquette de la liste A domine celle de la liste B :  $B[l_B]$  éliminée
12:      Éliminer  $B[l_B]$ ,  $l_B = l_B + 1$ 
13:    else           ▷ Les deux étiquettes sont non inférieures :  $A[l_A]$  insérée
14:      Insérer  $A[l_A]$  avant  $B[l_B]$ ,  $l_A = l_A + 1$ 
15:    fin si
16:  else si coût de  $A[l_A] =$  coût de  $B[l_B]$  alors
17:    ▷ L'étiquette de la liste A domine celle de la liste B :  $B[l_B]$  éliminée
18:    si consommation de  $A[l_A] <$  consommation de  $B[l_B]$  alors
19:      Éliminer  $B[l_B]$ ,  $l_B = l_B + 1$ 
20:    else           ▷ L'étiquette de la liste B domine celle de la liste A :  $A[l_A]$  sans effet
21:       $l_A = l_A + 1$ 
22:    fin si
23:  else
24:    ▷ Les deux étiquettes sont non inférieures, mais on ne pas insérer tout se suite
25:    si consommation de  $A[l_A] <$  consommation de  $B[l_B]$  alors
26:       $l_B = l_B + 1$ 
27:    else           ▷ L'étiquette de la liste B domine celle de la liste A :  $A[l_A]$  sans effet
28:       $l_A = l_A + 1$ 
29:    fin si
30:  fin si
31: fin tant que
32:   ▷ Si la fin d'une des deux listes est atteinte, ajouter les étiquettes restantes [ $\mathcal{O}(n_b)$ ]
33: si fin de la liste B atteinte ET fin de la liste A pas atteinte alors
34:   tant que consommation de  $A[l_A] \geq a$  faire
35:     Insérer  $A[l_A]$  à la fin de la liste,  $l_A = l_A + 1$ 
36:   fin tant que
37: else si fin de la liste A atteinte ET fin de la liste B pas atteinte alors
38:   Aller à la fin de la liste B
39: fin si

```

Annexe F

Algorithme de calcul du carburant à transporter

L'algorithme F1 décrit la manière dont la valeur de $R^2_{i_c j_c k_c}$ est calculée. Les nœuds sont visités en choisissant toujours celui dont le temps EDTO est le plus faible. À chaque nœud, les nœuds à la même altitude qui permettent d'agrandir la zone EDTO sont visités. Des segments qui descendent vers chaque successeur sont calculés. Chaque fois que le temps EDTO est plus faible que celui qui était déjà stocké dans le nœud, le temps et la consommation EDTO sont remplacés, ce qui assure que dans chaque nœud le temps le plus court vers un aéroport de décollage est stocké.

Algorithme F.1 Algorithme de calcul des réserves requises pour l'EDTO

```

1: pour tout les scénarios de l'EDTO faire
2:   pour tout  $d \in \mathcal{D}$  faire
3:      $t_{EDTO}(I_d J_d K_d) = 0$ 
4:      $\underbrace{R^2}_{I_d J_d K_d} = 0$ 
5:      $\mathcal{U} = \mathcal{U} \cup (I_d, J_d, K_d)$ 
6:   fin pour
7:   tant que  $\mathcal{U} \neq \emptyset$  faire
8:     Choisir le nœud  $(i_s, j_s, k_s) \in \mathcal{U}$  pour lequel  $t_{EDTO}(I_d J_d K_d)$  est le plus petit
9:     pour tout  $\{(i_c, j_c, k_s), (i_s, j_s, k_s)\}$  qui permet d'agrandir la zone EDTO faire
10:      Calculer  $r^2(i_c, j_c, k_s, i_s, j_s, k_s, \underbrace{R^2}_{i_s j_s k_s})$   $\triangleright$  Vitesse fixée par le scénario de l'EDTO
11:      si  $t_{EDTO}(i_s j_s k_s) + u < t_{EDTO}(i_c j_c k_s)$  alors
12:         $t_{EDTO}(i_c j_c k_s) = t_{EDTO}(i_s j_s k_s) + u$ 
13:         $\underbrace{R^2}_{i_c j_c k_s} = \underbrace{R^2}_{i_s j_s k_s} + r^2(i_c, j_c, k_s, i_s, j_s, k_s, \underbrace{R^2}_{i_s j_s k_s})$ 
14:         $\mathcal{U} = \mathcal{U} \cup (i_c, j_c, k_s)$ 
15:      fin si
16:      pour tout faire  $\{(i_c, j_c, k_s), (i_h, j_h, k_h)\}$  qui permet d'agrandir la zone EDTO et  $k_h >$ 
14
17:        Calculer  $r^2(i_h, j_h, k_h, i_c, j_c, k_s, \underbrace{R^2}_{i_c j_c k_s})$   $\triangleright$  Descente selon le scénario de
18:        l'EDTO
19:        si  $t_{EDTO}(i_h j_h k_h) + u < t_{EDTO}(i_c j_c k_s)$  alors
20:           $t_{EDTO}(i_h j_h k_h) = t_{EDTO}(i_c j_c k_s) + u$ 
21:           $\underbrace{R^2}_{i_h j_h k_h} = \underbrace{R^2}_{i_c j_c k_s} + r^2(i_h, j_h, k_h, i_c, j_c, k_s, \underbrace{R^2}_{i_c j_c k_s})$ 
22:        fin si
23:      fin pour
24:    fin tant que
25:  fin pour

```

Annexe G

Algorithme de calcul du dual

L'algorithme G.1 décrit la manière dont la valeur de $\underline{R}_{i_c j_c k_c}^2$ est calculée. Les nœuds sont visités dans l'ordre topologique inverse, donc du nœud d'arrivée vers le départ. À chaque nœud, tous les successeurs sont visités. Pour chacun, la consommation la plus faible jusqu'au nœud d'arrivée est stockée.

Algorithme G.1 Algorithme pour le calcul du dual

```
1: Initialiser  $(I_A, J_A, K_A)$ 
2:  $\mathcal{U} \leftarrow (I_A, J_A, K_A)$ 
3: tant que  $\mathcal{U} \neq \emptyset$  faire
4:   Choisir un nœud  $(i_s, j_s, k_s) \in \mathcal{U}$  dans l'ordre topologique inverse
5:    $\mathcal{U} = \mathcal{U} - C$ 
6:   pour tout  $\{(i_c, j_c, k_c), (i_s, j_s, k_s)\} \in E$  faire
7:      $b_{i_s j_s k_s}^2 = \min(b_{i_s j_s k_s}^2, b_{i_A j_A k_A}^2 - \underline{R}_{i_s j_s k_s}^2)$ 
8:     si  $(i_s, j_s, k_s) \in \mathcal{A}(i_c, j_c, k_c)$  alors
9:       Calculer  $r^2(i_c, j_c, k_c, i_s, j_s, k_s, \underline{R}_{i_s j_s k_s}^2)$  ▷ Voir annexe B pour le calcul
10:      si  $f_{i_c j_c k_c i_s j_s k_s}^2(\underline{R}_{i_s j_s k_s}^2) < \underline{R}_{i_c j_c k_c}^2$  alors
11:         $\mathcal{U} = \mathcal{U} \cup (i_c, j_c, k_c)$ 
12:        Calculer  $\underline{R}_{i_c j_c k_c}^2$  ▷ Selon les fonction présentées en 3.1.5
13:      fin si
14:    fin si
15:  fin pour
16: fin tant que
```

Liste des références

- ADAMS, D. R. et MCKENZIE, R. E. (1989). System and method for optimizing aircraft flight path.
- AGAM, L. et AGAM, D. (2009). System for producing a flight plan.
- AHUJA, R. K., MAGNANTI, T. L. et ORLIN, J. B. (1993). *Network Flows : Theory, Algorithms and Applications*. Prentice Hall, Engewood Cliffs, NJ.
- AIRBUS (2014). Airbus Global Market Forecast 2014-2033.
- AIRBUS FLIGHT OPERATIONS SUPPORT & LINE ASSISTANCE (1998). Getting to Grips with Cost Index.
- ALTUS, S. (2009). Effective Flight Plans Can Help Airlines Economize. *AERO*, 35:27–30.
- ANEJA, Y. P., AGGARWAL, V. et NAIR, K. P. K. (1983). Shortest Chain Subject to Side Constraints. *Networks*, 13(2):295–302.
- BEASLEY, J. et CHRISTOFIDES, N. (1989). An algorithm for the Resource Constrained Shortest Path Problem. *Networks*, 19(4):379–394.
- BELLMAN, R. (1956). On a routing problem. Rapport technique, The Rand Corporation.
- BERTSEKAS, D. P. (1998). *Network Optimization : Continuous and Discrete Models*. Athena Scientific.
- BERTSEKAS, D. P. (2005). *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts, 3 édition.
- BETTS, J. T. (1998). Survey of Numerical Methods for Trajectory Optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207.
- BOROUJERDI, A. et UHLMANN, J. (1998). An efficient algorithm for computing least cost paths with turn constraints. *Information Processing Letters*, 67(6):317–321.
- BOROUJERDI, A. et UHLMANN, J. K. (2002). Efficient computation of least cost paths with hard constraints.
- Bureau of Transportation Statistics* (2015). Airline On-Time Statistics and Delay Causes : Causes of National Aviation System Delays, National : January - December, 2014.
- CABOS, R. (2013). Onboard Flight Planning System.
- CARLYLE, W. M., ROYSET, J. O. et KEVIN WOOD, R. (2008). Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks*, 52(4):256–270.

- CHAKRAVARTY, A. J. M. (1995). Time-responsive flight optimization system.
- COOK, A., TANNER, G., ANDERSON, S. et EUROCONTROL (2004). Evaluating the true cost to airlines of one minute of airborne or ground delay. Rapport technique, University of Westminster, London.
- COOK, A., TANNER, G., WILLIAMS, V. et MEISE, G. (2007). Dynamic Cost Indexing. *In 6th EUROCONTROL Innovative Research Workshops & Exhibition*, pages 1–12, Brétigny sur Orge, France. EUROCONTROL.
- COOK, A., TANNER, G., WILLIAMS, V. et MEISE, G. (2009). Dynamic cost indexing – Managing airline delay costs. *Journal of Air Transport Management*, 15(1):26–35.
- COULMEAU, F. (2005). Procédé d’optimisation en cours de vol de la consommation en carburant d’un aéronef.
- COULMEAU, F., DEKER, G. et GOUTELARD, H. (2007). Procédé d’optimisation d’un plan de vol.
- DELGADO, L., PRATS, X. et SRIDHAR, B. (2013). Cruise speed reduction for ground delay programs : A case study for San Francisco International Airport arrivals. *Transportation Research Part C : Emerging Technologies*, 36:83–96.
- DESROCHERS, M. (1988). *An algorithm for the shortest path problem with resource constraint*. Les Cahiers du GERAD, G-88-27. Montréal, QC.
- DESROCHERS, M. et SOUMIS, F. (1985a). *Un algorithme de marquage permanent généralisé pour le problème de plus court chemin avec contraintes d’horaire*.
- DESROCHERS, M. et SOUMIS, F. (1985b). *Un algorithme de réoptimisation pour le problème de plus court chemin avec contraintes d’horaire*. Montréal, QC.
- DESROSIERS, J., PELLETIER, P. et SOUMIS, F. (1983). Plus court chemin avec contraintes d’horaire. *Revue française d’automatique, d’informatique et de recherche opérationnelle. Recherche opérationnelle*, 17(4):357–377.
- DEVULAPALLI, R. (2012). *An Efficient Algorithm for Commercial Aircraft Trajectory Optimization in the Air Traffic System*. Thèse de doctorat, University of Minnesota.
- DI PUGLIA PUGLIESE, L. et GUERRIERO, F. (2013). A survey of resource constrained shortest path problems : Exact solution approaches. *Networks*, 62(3):183–200.
- DIAL, R. B. (1969). Algorithm 360 : Shortest path forest with topological ordering. *Communications of the ACM*, 12(11):632–633.
- DIJKSTRA, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.
- DORIGO, M., STÜTZLE, T., GENDREAU, M. et POTVIN, J.-Y. (2010). Chapter 8 : Ant Colony Optimization Overview and Recent Advances. *In Handbook of Metaheuristics*, page 648. Springer, New York, 2e édition.
- DOUGUI, N., DELAHAYE, D., PUECHMOREL, S. et MONGEAU, M. (2013). A light-propagation model for aircraft trajectory planning. *Journal of Global Optimization*, 56(3):873–895.

- DUMITRESCU, I. et BOLAND, N. (2003). Improved preprocessing, labeling and scaling algorithms for the Weight-Constrained Shortest Path Problem. *Networks*, 42(3):135–153.
- FEDERAL AVIATION ADMINISTRATION (FAA) (2014). Aeronautical Information Manual.
- FERGUSON, D., KALRA, N. et STENTZ, A. (2006). Replanning with RRTs. In *IEEE International Conference on Robotics and Automation*, pages 1243–1248, Orlando, Florida. IEEE.
- FORBES, S. J. (2008). The effect of air traffic delays on airline prices. *International Journal of Industrial Organization*, 26(5):1218–1232.
- FREDMAN, M. L. et TARJAN, R. E. (1984). Fibonacci heaps and their uses in improved network optimization algorithms. In *25th Annual Symposium on Foundations of Computer Science*, pages 338–346. IEEE.
- GIRARDET, B. (2014). *Trafic Aérien : Détermination optimale et globale des trajectoires d'avions en présence de vent*. Thèse de doctorat, Institut National des Sciences Appliquées de Toulouse, Toulouse, France.
- GOUVERNEMENT DU CANADA (1996). *Règlement de l'aviation canadien*. Dors/96-43 édition.
- HANDLER, G. Y. et ZANG, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–310.
- HART, P. E., NILSSON, N. J. et RAPHAEL, B. (1968). Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.
- HUANG, G., LU, Y. et NAN, Y. (2012). A survey of numerical algorithms for trajectory optimization of flight vehicles. *Science China Technological Sciences*, 55(9):2538–2560.
- INTERNATIONAL AIR TRANSPORT ASSOCIATION (2015). New IATA Passenger Forecast Reveals Fast-Growing Markets of the Future.
- IRNICH, S. et DESAULNIERS, G. (2004). *Shortest path problems with resource constraints*. Les Cahiers du GERAD, G-2004-11. Montréal, QC.
- JOKSCH, H. (1966). The shortest route problem with constraints. *Journal of Mathematical analysis and applications*, 14(2):191–197.
- JÜTTNER, A., SZVIATOVSKI, B., MÉCS, I. et RAJKÓ, Z. (2001). Lagrange relaxation based method for the QoS routing problem. In *20th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, volume 2, pages 859–868. IEEE.
- KOENIG, S. et LIKHACHEV, M. (2002). Improved Fast Replanning for Robot Navigation in Unknown Terrain. In *IEEE International Conference on Robotics & Automation*, pages 968–975, Washington D.C. IEEE.
- LOZANO, L. et MEDAGLIA, A. L. (2013). On an exact method for the constrained shortest path problem. *Computers & Operations Research*, 40(1):378–384.
- MAAZOUN, W. (2015). *Conception et analyse d'un système d'optimisation de plans de vol pour les avions*. Thèse de doctorat, École Polytechnique de Montréal.

- MINOUX, M. (1975). Plus court chemin avec contraintes : Algorithmes et applications. *Annales des télécommunications*, 30(11-12):1–12.
- MUHANDIRAMGE, R. et BOLAND, N. (2009). Simultaneous Solution of Lagrangean Dual Problems Interleaved with Preprocessing for the Weight. 53(4):358–381.
- MYERS, W. L. (2000). System for determination of optimal travel path in a multidimensional space.
- NAV CANADA (2014). Completion of Canadian Flight Plan/Flight Itinerary and ICAO Flight Plan.
- NILIM, A. et EL GHAOUI, L. (2004). Algorithms for air traffic flow management under stochastic environments. *In Proceedings of the 2004 American Control Conference*, volume 4, pages 3429–3434.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (1993). Manuel de l'atmosphère type OACI élargie jusqu'à 80 kilomètres (262 500 pieds).
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2002). Manual on Implementation of a 300 m (1000 ft) Vertical Separation Minimum Between FL 290 and FL 410 Inclusive.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2005). Annex 2 to the Convention on International Civil Aviation : Rules of the Air.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2010). Annex 6 to the Convention on International Civil Aviation : Operation of Aircraft.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2014a). EDTO Workshop Module 6 - Flight Operations Considerations.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2014b). EDTO Workshop Module 7 - Implementing EDTO Regulations.
- ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE (OACI) (2014c). Plan mondial de navigation aérienne.
- REEVES, C. R., GENDREAU, M. et POTVIN, J.-Y. (2010). Chapter 5 : Genetic Algorithms. *In Handbook of Metaheuristics*. Springer, New York, 2e édition.
- REINHARDT, L. B. et PISINGER, D. (2011). Multi-objective and multi-constrained non-additive shortest path problems. *Computers & Operations Research*, 38(3):605–616.
- ROSENBERGER, J. M., SCHAEFER, A. J., GOLDSMAN, D., JOHNSON, E. L., KLEYWEGT, A. J. et NEMHAUSER, G. L. (2002). A stochastic model of airline operations. *Transportation Science*, 36(4):357.
- SANTOS, J. L. et PAIXA, J. M. (2013). Chapter 46 : Labeling Methods for the General Case of the Multi-objective Shortest Path Problem - A Computational Study. *In MADUREIRA, A., REIS, C. et MARQUES, V., éditeurs : Computational Intelligence and Decision Making : Trends and Applications*, volume 61 de *Intelligent Systems, Control and Automation : Science and Engineering*, chapitre 46, pages 489–502. Springer Netherlands, Dordrecht.

- SANTOS, L., COUTINHO-RODRIGUES, J. a. et CURRENT, J. R. (2007). An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B : Methodological*, 41(7):756–771.
- SCHULTZ, R. L. et SHANER, D. A. (2001). Multi-dimensional route optimizer.
- SCHULTZ, R. L., SHANER, D. A. et PRATT, S. G. (2003). Four-dimensional route planner.
- SESAAR CONSORTIUM (2012). The Roadmap for Sustainable Air Traffic Management : European ATM Master Plan.
- SIDIBÉ, S. (2014). *Optimisation des trajectoires d'un système de gestion de vol d'avions pour la réduction des coûts de vol*. Mémoire de maîtrise, École de technologie supérieure.
- SOLER, M., OLIVARES, A., STAFFETTI, E. et BONAMI, P. (2011). En-Route Optimal Flight Planning Constrained to Pass Through Waypoints using MINLP Optimal Horizontal Motion of a Commercial Aircraft Constrained to Pass Through Waypoints using MINLP. In *Ninth USA/Europe Air Traffic Management Research and Development Seminar*, Berlin.
- STENTZ, A. (1994). Optimal and efficient path planning for partially-known environments. In *IEEE International Conference on Robotics and Automation*, pages 3310–3317. IEEE.
- STENTZ, A. (1995). The focussed D* algorithm for real-time replanning. In *14th International Joint Conference on Artificial intelligence*, numéro August, pages 1652–1659.
- TALBI, E.-G. (2009). Chapter 1 : Common Concepts for Metaheuristics. In JOHN WILEY & SONS, éditeur : *Metaheuristics : From Design to Implementation*, page 625.
- TANG, B. et MUKHERJEE, A. (2000). System and method for generating optimal flight plans for airline operations control.
- TERROVITIS, M., BAKIRAS, S., PAPADIAS, D. et MOURATIDIS, K. (2005). Constrained Shortest Path Computation. In *Advances in Spatial and Temporal Databases*, pages 181–199. Angra dos Reis, Brazil.
- TRANSPORT CANADA (2007). Safety Criteria for Approval of Extended Range Twin-Engine Operations (ETOPS).
- VERLUT, G. et LOHMILLER, W. (2010). Computation-Time-Optimized Route Planning for Aircraft.
- VON STRYK, O. et BULIRSCH, R. (1992). Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37(1):357–373.
- WU, D. (2012). *Models, Optimal Performances and Sensitivities of Commercial Flight Trajectory in the Air Traffic System*. Thèse de doctorat, University of Minnesota.
- WZOREK, M., KVARNSTRÖM, J. et DOHERTY, P. (2010). Choosing Path Replanning Strategies for Unmanned Aircraft Systems. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling*, pages 193–200. Association for the Advancement of Artificial Intelligence.
- YANG, J., QU, Z. et HULL, R. A. (2009). Real-Time Optimized Trajectory Planning for a Fixed-Wing Vehicle Flying in a Dynamic Environment. *Journal of Aerospace Engineering*, 22(4):331–341.

-
- YOON, Y., HANSEN, M. et BALL, M. O. (2012). Optimal route decision with a geometric ground-airborne hybrid model under weather uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):34–49.
- ZHANG, W., KAMGARPOUR, M., SUN, D. et TOMLIN, C. J. (2011). Decentralized Flight Path Planning for Air Traffic Management. *In American Control Conference, 2011*, pages 2137–2142, San Francisco, CA.
- ZHU, X. et WILHELM, W. E. (2012). A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation. *Computers & Operations Research*, 39(2):164–178.