



HAL
open science

Contributions à la conception de réseau de service en transport

Susann Schrenk

► **To cite this version:**

Susann Schrenk. Contributions à la conception de réseau de service en transport. Sciences de l'ingénieur [physics]. Institut National Polytechnique de Grenoble - INPG, 2010. Français. NNT : . tel-00539269

HAL Id: tel-00539269

<https://theses.hal.science/tel-00539269v1>

Submitted on 24 Nov 2010

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

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

UNIVERSITE DE GRENOBLE
INSTITUT POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--	--

THESE

pour obtenir le grade de

DOCTEUR DE L'Université de Grenoble
Délivré par l'Institut Polytechnique de Grenoble

Spécialité : "Génie Industriel : conception et production"

préparée au laboratoire G-SCOP (Grenoble - Sciences pour la Conception, l'Optimisation et la Production)

dans le cadre de l'**Ecole Doctorale I-MEP² "Ingénierie - Matériaux Mécanique Energétique Environnement Procédés Production"**)

présentée et soutenue publiquement

par

Susann SCHRENK

le 23 septembre 2010

Contributions à la conception de réseau de service
en transport

Directeur de thèse : Van-Dat CUNG

Co-encadrant : Gerd FINKE

JURY

M. Teodor Gabriel CRAINIC
M. Christian ARTIGUES
M. Marino WIDMER
M. Van-Dat CUNG
M. Gerd FINKE
M. Alexandre DOLGUI

CIRRELT Montréal
LAAS-CNRS Toulouse
Université de Fribourg
Grenoble-INP
Université Joseph Fourier
Ecole des Mines de Saint-Etienne

Président
Rapporteur
Rapporteur
Directeur de thèse
Co-encadrant
Examinateur

Remerciements

Je souhaite adresser mes plus sincères remerciements à tous ceux qui ont contribué à ce travail et qui m'ont accompagné durant ces trois années de thèse.

Je tiens tout particulièrement à remercier mes directeurs de thèse Van-Dat Cung et Gerd Finke. Un grand merci à toi, Van-Dat d'avoir pris le risque de me prendre en thèse, je t'en suis très reconnaissante. Ton encadrement, tes encouragements et tout ce que tu as fait pour moi et pour construire, bien au-delà de la thèse, une véritable amitié. Merci Gerd, pour votre disponibilité et votre culture qui ont permis de rendre agréable voir drôle des problèmes très durs. Merci pour ce plaisir que j'ai eu à travailler avec vous et les souvenirs de votre jeunesse.

Je remercie sincèrement Christian Artigues et Marino Widmer d'avoir accepté d'être les rapporteurs de cette thèse. Merci Marino, vos remarques et votre rapport me sont d'une grande valeur. Doublement merci Christian, personne n'aura lu mon manuscrit avec autant de minutie que vous, et merci pour votre implication en tant que membre organisateur du challenge ROADEF 2009.

J'adresse mes remerciements à M. Teodor Gabriel Crainic d'avoir présidé mon jury de thèse et d'avoir collaboré avec moi. Ses conseils, sa proximité et son professionnalisme me vont droit au cœur.

Mes remerciements vont ensuite à M. Alexandre Dolgui qui m'a fait l'honneur d'examiner mon travail.

Enfin, je voudrai remercier de manière générale toute l'équipe du laboratoire G-SCOP : Yannick Frein son directeur qui m'a laissé une grande liberté pour animer la vie des doctorants. L'ensemble de l'équipe des chercheurs, enseignant-chercheurs et enseignants pour les échanges d'égal à égal. Tous les doctorants, stagiaires et post-doc et particulièrement les membres de l'A-DOC. Je suis reconnaissante envers toute l'équipe technique et administrative du laboratoire. Je tiens également à remercier le cluster GOSPI de la région Rhône-Alpes pour le financement de ma thèse.

Je remercie mes amis qui m'ont encouragé pendant ces trois années et qui ont toujours été à mes côtés. Je suis particulièrement reconnaissante envers mes formateurs, en particulier toute l'équipe de Valoridoc qui m'a aidée à préparer l'après thèse en toute confiance.

Pour conclure, je ne serai jamais assez reconnaissant envers ma famille, qui s'est agrandie au fil des théorèmes. Louise, ma louloute, tu as l'âge de ma thèse, merci d'avoir fait vite pour dormir la nuit. Sacha, mon pépère, merci de m'avoir redonné des forces pour rédiger ce manuscrit. Xavier, merci pour tout.

Sommaire

Introduction	1
--------------	---

Partie I Flot, transport, réseaux - quelques problèmes combinatoires

1 Problèmes dans les réseaux	11
1.1 Problèmes de flots dans les réseaux	11
1.1.1 Problème de flot général	12
1.1.2 Problème de flot de coût minimal	13
1.1.3 Problème de transport	13
1.2 Réseau bien équilibré et paradoxe de transport	15
1.2.1 Définition d'un réseau bien équilibré	16
1.2.2 Une caractérisation	17
1.2.3 Critère de dualité	19
1.3 Réseau et changement d'outils	22
1.3.1 Modélisation flot borné	23
1.3.2 Cas particulier du changement d'outils	24
2 Complexité de quelques problèmes combinatoires fondamentaux	29
2.1 Coût fixe et dégénérescence	29
2.1.1 Problèmes de transport avec coût fixe (FCTP)	30
2.1.2 Dégénérescence dans le FCTP	31
2.1.3 Plus grand degré de dégénérescence dans le FCTP	33
2.2 Complexité due aux contraintes spécifiques de gestion de véhicules	39
2.2.1 Rotation de véhicules : un problème de décomposition de graphe	40
2.2.2 De la complexité de l'équilibrage des charges entre véhicules	42
Synthèse	42

Partie II Conception de réseau de service pour le transport de fret

3	Conception de réseau de service pour le transport de fret	47
3.1	Description du problème	48
3.1.1	Données	48
3.1.2	Contraintes	51
3.1.3	Fonction d'optimisation	52
3.1.4	Solution	52
3.2	Complexité du problème	54
3.2.1	Un problème \mathcal{NP} ?	55
3.2.2	Un problème au moins \mathcal{NP} -difficile	55
3.2.3	Complexité théorique et pratique due aux contraintes spécifiques	58
3.3	Etat de l'art	59
3.3.1	Systèmes de transport de marchandises	59
3.3.2	Flot multicommodités	62
3.3.3	Tournées de véhicules	63
3.3.4	Conception de réseau	64
3.3.5	Conception de réseau de service	67
3.3.6	Positionnement de notre problème de transport de fret	69
4	Etude comparée de différentes formulations	71
4.1	Quatre formulations du problème	72
4.1.1	Formulation <i>arc-arc</i>	72
4.1.2	Formulation <i>cycle-arc</i>	74
4.1.3	Formulation <i>arc-chemin</i>	76
4.1.4	Formulation <i>cycle-chemin</i>	78
4.2	Comparaison théorique des formulations	79
4.2.1	Taille des formulations	79
4.2.2	Comparaison théorique des formulations	80
4.3	Comparaison expérimentale des modèles	82
4.3.1	Instances	82
4.3.2	Résultats sur nos instances	83
5	Résolution par génération de colonnes	87
5.1	La génération de colonnes	88
5.1.1	Principe pour résoudre un PL	88
5.1.2	Principe pour résoudre un PLNE	89
5.2	Algorithme de génération de colonnes	90
5.2.1	Problème maître restreint	91
5.2.2	Sous-problème de génération de cycles pour les véhicules	92
5.2.3	Sous-problème de génération de chemins pour les marchandises	93

5.3	Résultats numériques	93
5.4	Résolution d'instances industrielles	96
5.4.1	Instances de type transport express	97
5.4.2	Instances de type transport lent	97
	Synthèse	98

Partie III Gestion de perturbation dans le milieu aérien

6	Gestion de perturbation dans le milieu aérien	103
6.1	Description du problème	104
6.1.1	Données	104
6.1.2	Contraintes	110
6.1.3	Fonction d'optimisation	112
6.1.4	Structure d'une solution	113
6.2	Etat de l'art	116
6.2.1	Planification dans le domaine aérien : processus et modèles	116
6.2.2	Recouvrement d'une seule ressource	120
6.2.3	Recouvrement intégrant plusieurs ressources	121
6.3	Les méthodes proposées par les finalistes	122
7	Notre approche de résolution	127
7.1	Structure de l'algorithme de résolution	127
7.2	Construction d'un nouveau plan de vols	128
7.2.1	Schéma général et modélisation	129
7.2.2	Décomposition en paquets.	133
7.2.3	Création de nouveaux vols pour les maintenances.	133
7.2.4	Coûts de réaffectation	135
7.3	Réaccommodation des passagers	136
7.3.1	Itinéraires alternatifs pour la réaccommodation	136
7.3.2	Modélisation	137
8	Résultats expérimentaux	139
8.1	Instances	139
8.2	Temps de calcul	140
8.3	Qualité des solutions obtenues	141
	Synthèse	143
	Conclusions et perspectives	145

A Notations pour le problème de conception de réseau de service	147
B Notations pour le problème de gestion de perturbation dans le milieu aérien	149
Bibliographie	151
Résumé	157

Introduction

Enjeux du transport

Le transport dans une chaîne logistique peut représenter jusqu'à 30% des coûts, et ces coûts ne cessent d'augmenter ces dernières années à cause notamment des productions en juste-à-temps et de la mondialisation des échanges. Par exemple, d'après la fédération nationale des transports routiers, les coûts du transport routier ont augmenté de plus de 7% en 2008 par rapport à 2007. La réduction des coûts s'est opérée sur la production et le stock au détriment du transport. L'optimisation du maillon transport d'une chaîne logistique est donc une source de réduction non négligeable de coûts et de pollutions. Limiter les pollutions et réduire toutes les nuisances faites à l'environnement est un enjeu majeur. En effet, réduire son impact carbone est intéressant non seulement financièrement mais également pour l'image de marque d'une entreprise.

En avril 2010, un nuage de cendres volcaniques venu d'Islande et se propageant sur le ciel européen oblige plus d'une vingtaine de pays à fermer tout ou partie de leur espace aérien. Seuls moins de 10 000 vols sur 30 000 prévus ont pu être opérés lors de la "journée noire" du samedi 17 avril 2010. L'association internationale du transport aérien (IATA) estime à plus de 200 millions de dollars par jour le coût de ces perturbations pour les compagnies aériennes. Gérer au mieux la reprise du trafic et rapatrier rapidement les passagers bloqués sont donc des priorités pour limiter les conséquences de telles perturbations.

Voici deux exemples qui montrent les enjeux de l'optimisation dans le domaine du transport. Le contexte économique étant caractérisé par une concurrence féroce, des marges réduites et une complexité croissante des décisions de gestion, la Recherche Opérationnelle devient souvent un outil d'aide à la décision indispensable pour améliorer la performance des entreprises et notamment de leurs chaînes logistiques. Nous nous plaçons dans ce contexte, et abordons ces problèmes avec un regard d'aideur à la décision fondé sur les techniques et outils de Recherche Opérationnelle.

Deux problèmes de transport

Dans cette thèse nous nous intéressons à deux applications industrielles particulières dans le domaine du transport. La première concerne le transport de fret, la seconde le transport aérien de passagers.

La première application est un problème de conception de réseau de service avec gestion de véhicules pour un transport de fret. Notre positionnement est celui d'un transporteur avec consolidation. Le problème de transport de fret étudié est issu d'une collaboration industrielle antérieure entre le laboratoire G-SCOP et un industriel. Le but est d'apporter une aide à la décision pour le transporteur en répondant aux questions suivantes : Quelles demandes accepter totalement, partiellement ou refuser ? Comment acheminer les marchandises à travers les réseaux physiques et de services ? Sur quels parcours doit-on fournir le service ? Quel type de service doit-on utiliser ? A quelle fréquence doit-on offrir le service et selon quels horaires ? Le but est d'obtenir un plan de transport qui maximise le profit du transporteur, tout en s'assurant qu'un certain nombre de contraintes spécifiques à l'application industrielle soit vérifié (quotas d'utilisation des véhicules, fenêtre de temps, cyclage des véhicules, ...).

Dans un premier temps, des travaux [Teypez, 2008] ont été réalisés pour répondre à la question : "Quelle serait la flotte idéale pour maximiser les profits ?", et pour proposer le plan de transport correspondant. Ces travaux ont fait l'objet d'un article publié [Teypez *et al.*, 2010].

La nouvelle contrainte forte que nous considérons est que le transporteur a une flotte fixée. Une réponse au problème consiste alors à déterminer un ensemble de plannings (itinéraires, horaires et opérations de transport effectuées – chargement et déchargement) pour les véhicules et pour la période de planification considérée. Ces plannings doivent être choisis de sorte que le transporteur maximise son profit et doivent respecter les contraintes spécifiques suivantes. D'une part, le transporteur planifie un transport régulier, il souhaite donc répéter les plannings à l'identique d'une période sur l'autre. Il faut donc assurer que la planification faite sur une seule période permette qu'une rotation des véhicules s'opère de période en période afin de réduire au maximum le voyage à vide des véhicules. D'autre part, le transporteur souhaite utiliser ses véhicules de manière équitable, il faut donc garantir qu'en fin de période aucun véhicule ne soit ni sous- ni sur-exploité. Plus précisément, le temps d'utilisation d'un véhicule est borné par des quotas minimum et maximum.

Dans la littérature on fait référence à ce problème avec le terme "Conception de réseau de service avec gestion de ressources", *Service Network Design with Asset management* [Andersen *et al.*, 2009b].

La seconde application concerne la gestion de perturbation dans le domaine aérien. Il s'agit du sujet du challenge ROADEF'2009. Cette année là le sujet était proposé par Amadeus¹ et concernait la gestion de perturbations dans le domaine aérien². Il s'agit, en cas de perturbations affectant le plan de vol initial d'une compagnie aérienne, de reconstruire ce plan de vol à l'intérieur d'une période de recouvrement en réaffectant de façon simultanée la flotte d'appareils et les passagers. Il s'agit de trouver des solutions performantes permettant d'absorber la perturbation en un temps minimum et, de fait, d'en minimiser l'impact.

1. <http://www.amadeus.com/>

2. <http://challenge.roadef.org/2009/index.fr.htm>

Dans le cadre du challenge, on connaît le programme de vol initial prévu, conçu afin d'optimiser les revenus générés et qui respecte les contraintes opérationnelles et de maintenance. On envisage différentes perturbations extérieures se traduisant par des retards ou des annulations de vols, des indisponibilités d'appareils ou des réduction des capacités aéroportuaires de décollage et d'atterrissage. De façon traditionnelle, dans de tels cas, les ressources sont réaffectées au cours d'un processus séquentiel selon l'ordre hiérarchique naturel : flotte d'appareils, équipages, passagers. Cependant, une modification du plan de vol entraîne souvent une impossibilité de correspondance pour un équipage ou un ensemble de passagers. Idéalement, il faudrait réaffecter toutes les ressources simultanément. Le but du challenge est de réaffecter de façon simultanée deux de ces ressources : les passagers et la flotte d'appareils. Il faut donc reconstruire un plan de vol à l'intérieur d'une période de recouvrement et minimiser non seulement les coûts opérationnels (liés principalement aux appareils), mais aussi le désagrément subi par les passagers. Les contraintes sur les appareils, notamment les contraintes de maintenance conjointement avec les contraintes de capacité aéroportuaire rendent le problème riche et difficile à résoudre.

Le sujet est particulièrement novateur, ardu et difficile à formaliser. L'objectif est de fournir un programme qui résout ce problème en moins de 10 minutes !

Méthodologie générale

Les deux problèmes que nous étudions contiennent une dimension temporelle forte. Dans les deux cas, il faut établir des plannings avec des horaires précis. De plus, il faut synchroniser les différents véhicules pour permettre des opérations de transbordements de marchandises, ou pour garantir les correspondances des passagers. De ce fait, il est nécessaire d'inclure le temps de façon explicite dans les modèles. On se place alors traditionnellement dans un réseau espace-temps (voir figure 1). Chaque nœud de ce réseau

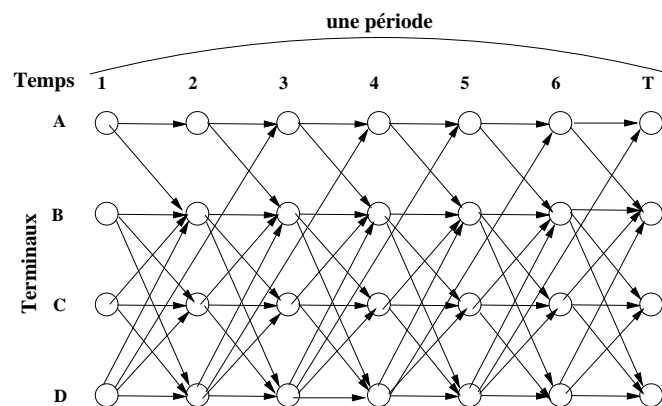


FIGURE 1 – Réseau espace-temps.

correspond à un terminal à un horaire donné. Les arcs du réseau correspondent à des arcs de services (vols) potentiels avec horaires intégrés.

Nos deux problèmes industriels ont une structure commune représentée dans la figure 2. Il s'agit de faire transiter un flot, soit de passagers, soit de marchandises, via un réseau de transport comportant une dimension temporelle. Dans les deux cas les connexions du réseau de transport sont à opérer par des véhicules du transporteur. On retrouve donc deux flux dans un réseau espace-temps :

- les flux de passagers ou de marchandises,
- les flux de véhicules qui permettent de définir les plans de transport ou les rotations des véhicules.

Dans les deux cas, ces flux (multiflots) sont liés par une contrainte forçante ou liante: rien ni personne ne peut transiter via un trajet non opéré. De plus, le flot de marchandises et de personnes est limité par la capacité des véhicules. Un certain nombre de contraintes plus spécifiques sur les véhicules ou les passagers/marchandises viennent compléter ces modèles. Les fonctions d'optimisation sont différentes mais, dans les deux cas, il s'agit de fonction de coût concave, faites d'une combinaison de coûts fixes et de coûts variables linéaires.

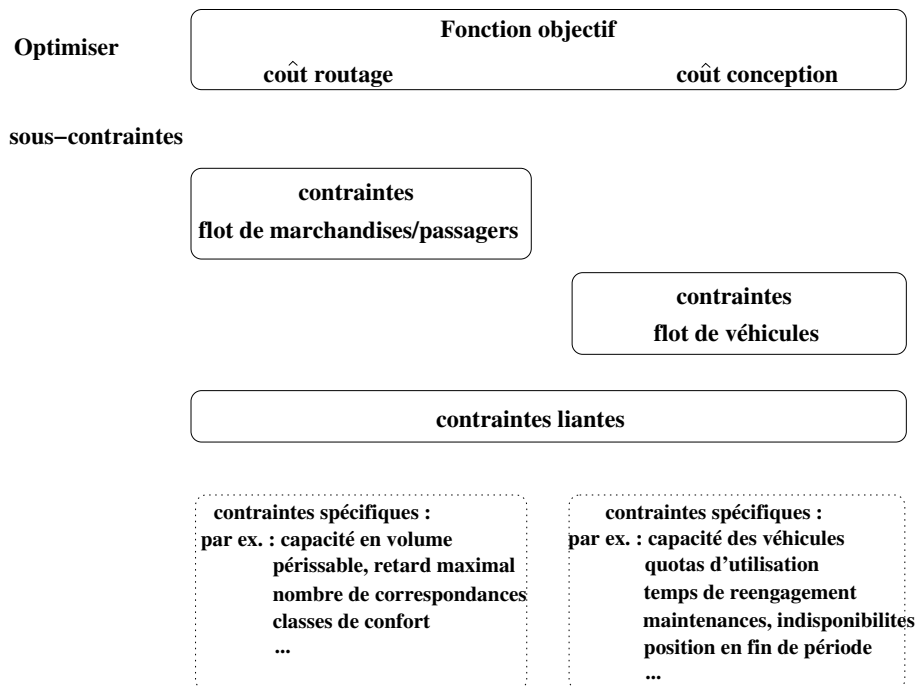


FIGURE 2 – Schéma de modélisation générique des deux problèmes.

Plan de travail et contributions

Le présent travail se découpe en trois parties, qui correspondent à différents niveaux

d'abstraction. Dans un premier temps, nous nous intéressons à des problèmes combinatoires purs. Ensuite, nous étudions différentes modélisations et résolutions du problème de conception de réseau de service avec gestion de véhicules. Pour finir, nous présentons le travail réalisé pour répondre au challenge ROADEF'2009 sur la gestion de perturbation dans le domaine aérien.

Première partie

La première partie est consacrée à l'étude de problèmes fondamentaux. Nous y présentons quelques sous-problèmes combinatoires de nos problèmes industriels.

Dans le premier chapitre, nous abordons les problèmes de flot dans les réseaux et plus particulièrement le problème de transport classique de la littérature. Etant donnée une quantité d'un produit uniforme disponible à m origines, des demandes de ce produit à n destinations, et la connaissance des coûts de transport origine-destination, le problème de transport consiste à trouver une solution d'expédition qui satisfait toutes les demandes et à moindre coût. Dans un premier temps, nous supposons que les coûts de transport sont linéaires. Dans ce cas trouver la solution optimale du problème de transport est équivalent à résoudre un problème de flot, il s'agit d'un problème facile.

Mais, il se peut qu'augmenter les quantités des offres et des demandes permette de réduire les coûts de transport ! Il s'agit du paradoxe de transport, i.e. le réseau n'est pas bien équilibré. L'étude et la caractérisation des réseaux bien équilibrés, ainsi qu'une méthode polynomiale de levée du paradoxe sont des extensions étudiées dans ce chapitre. Les problèmes étudiés dans ce cadre sont polynomiaux.

Le second chapitre rassemble les résultats concernant la complexité de quelques problèmes combinatoires. Nous commençons par étudier la complexité engendrée par les coûts fixes. Pour cela nous considérons le problème de transport avec un coût fixe constant identique sur toutes les liaisons de transport. Nous démontrons que ce problème est équivalent à trouver un arbre de degré de dégénérescence maximal et qu'il est \mathcal{NP} -difficile. Nous démontrons que ce problème reste \mathcal{NP} -difficile même lorsqu'une grande partie de la combinatoire sous-jacente est donnée. Ce résultat nouveau a fait l'objet d'un article soumis à *Mathematical and Computer Modelling*. Pour finir, nous étudions la complexité engendrée par quelques contraintes spécifiques aux applications industrielles : contraintes de rotations de véhicules et contraintes de quotas d'utilisation. Nous interprétons ces problèmes comme des problèmes de partitionnement de graphes. Nous démontrons que dans la plupart des cas ces problèmes sont \mathcal{NP} -difficiles et exhibons quelques problèmes dont la complexité est encore inconnue.

Deuxième partie

Cette partie porte sur le problème de conception de réseau de service.

En guise d'introduction, le troisième chapitre présente notre problème de conception de réseau de service pour le transport de fret. Nous démontrons qu'il s'agit d'un problème

au moins \mathcal{NP} -difficile. Finalement nous donnons un aperçu de la littérature, nous citons en particulier les travaux de Crainic et al. et de Barnhart et al. sur le sujet.

Le quatrième chapitre est dédié à l'étude de différentes formulations en programme linéaire en nombres entiers de notre problème. Ces modèles sont des modèles de conception de réseau de service avec gestion de flotte. Ils comportent deux types de variables : les variables de service et les variables pour le flot de marchandises.

Les variables de services définissent le parcours des véhicules sur le réseau. On peut les définir soit directement sur les arcs du réseau espace-temps, soit sur les parcours potentiels des véhicules. Ces parcours sont des chemins, morceaux de cycles, dans le réseau espace-temps de longueurs égales à la longueur de la période.

De même les variables concernant les trajets des marchandises à travers le réseau peuvent être formulées sur les arcs du réseau espace-temps, ou bien sur les chemins potentiels des marchandises de leur origine à leur destination. Dans les deux cas ces variables correspondent à la quantité de marchandises qui transite via le réseau.

Ces différents choix nous amènent à quatre modélisations différentes selon que les variables de service sont exprimées sur les arcs ou les parcours (morceaux de cycles), et selon que les variables concernant le flot de marchandises sont exprimées sur les arcs ou les chemins. Les quatre formulations *arc-arc*, *cycle-arc*, *arc-chemin* et *cycle-chemin* du problème ont toutes la même structure générale, représentée dans le diagramme de la figure 2.

Ces quatre modélisations sont implémentées et comparées expérimentalement sur des jeux de petites instances générées à partir d'un générateur d'instances. Pour ces instances, les morceaux de cycle et les chemins pour les marchandises sont tous calculés a priori. Nous faisons ressortir la dominance des modèles à base de cycles, et démontrons également cette dominance analytiquement.

Le chapitre cinq propose une ébauche d'algorithme à base de génération de colonnes pour résoudre des instances de plus grande taille. Nous proposons une heuristique simple à base de génération de colonnes. Sa performance est estimée sur les petites instances pour lesquelles nous connaissons les solutions optimales. Des tests permettent de cerner la limite de l'heuristique.

Troisième partie

Cette dernière partie présente le travail que j'ai réalisé avec trois autres doctorants du laboratoire G-SCOP pour répondre au challenge ROADEF'2009 proposé par la société Amadeus qui portait sur la gestion de perturbation dans le domaine aérien, challenge auquel nous avons remporté le deuxième prix de la catégorie Junior et cinquième au classement général.

Le chapitre six est introductif. Nous présentons le problème industriel de gestion de perturbation, décrivons une instance du problème et les solutions attendues. Notre état de l'art traite des problèmes de recouvrement d'une seule ressource, ou de plusieurs ressources

(ce qui est le cas du challenge). Nous proposons également une revue des méthodes utilisées par les autres équipes finalistes.

Le septième chapitre porte sur notre approche de résolution. Les instances à résoudre sont de grande taille (1000 vols, 85 avions, passagers) et le problème comporte un grand nombre de contraintes spécifiques. Dans le cadre du challenge, ce qui importe, c'est de parvenir à proposer des solutions réalisables de bonne qualité en moins de 10 minutes de temps de calcul sur un ordinateur de bureau. Nous proposons une approche heuristique fondée sur une décomposition du problème. En premier lieu, nous calculons un nouveau plan de vols proche du plan de vol initial. Ce calcul est réalisé par morceaux. On traite d'un nombre limité d'avions à chaque fois, et on reconstruit un plan de vols pour ces avions en résolvant un modèle de multiflot dans un réseau (les arcs du réseau correspondent aux vols) avec contraintes additionnelles. En priorité, on détermine les rotations des appareils ayant une maintenance, avec si besoin création de vols. Ensuite, dans une seconde étape, on réacommode les passagers sur le nouveau plan de vols, en résolvant un problème de multiflot. Les différents modèles en nombre entiers sont résolus directement par IBM ILOG Cplex.

Le huitième et dernier chapitre donne les résultats numériques obtenus après une légère modification de notre code qui garantit la réalisabilité des solutions pour tous les jeux d'instance. Cette modification nous aurait permis d'atteindre la première place de la catégorie junior et d'être troisième au classement général.

Première partie

Flot, transport, réseaux - quelques problèmes combinatoires

Le but de cette première partie est d'étudier des aspects théoriques de quelques problèmes fondamentaux, sous-problèmes au cœur de nos problèmes industriels. Il s'agit des sous-problèmes les plus simplifiés possibles mais qui sont déjà d'une très grande richesse d'un point de vue combinatoire. Nous nous intéressons à deux axes de recherche.

D'une part, dans le chapitre 1, nous étudions les problèmes polynomiaux de flots dans les réseaux de distribution. En particulier nous étudions le problème de transport, cas spécifique du problème de flot. Nous proposons quelques extensions, notamment autour du paradoxe de transport. Nous modélisons également un problème de repositionnement de véhicules via un flot dans un réseau.

D'autre part, dans le chapitre 2, nous étudions la complexité de sous-problèmes. Ce chapitre vise notamment à comprendre d'où vient la difficulté des problèmes. Ainsi, Nous nous intéressons au problème de transport avec coût fixe, pour montrer que dans le cas le plus simple ce problème est déjà difficile. Nous discutons également de la difficulté engendrée par les contraintes pratiques de cyclage de véhicules.

Les résultats concernant le problème de transport ont été soumis fin 2009 dans le journal international *Mathematical and Computer Modelling*. On y retrouve la définition du problème de transport (section 1.1) et du problème de transport avec coût fixe. Une des contributions majeures est l'étude de la complexité du problème de transport avec coût fixe (section 2.1). Dans cet article apparaît également une note sur le paradoxe de transport (section 1.2).

Chapitre 1

Problèmes dans les réseaux

Sommaire

1.1	Problèmes de flots dans les réseaux	11
1.1.1	Problème de flot général	12
1.1.2	Problème de flot de coût minimal	13
1.1.3	Problème de transport	13
1.2	Réseau bien équilibré et paradoxe de transport	15
1.2.1	Définition d'un réseau bien équilibré	16
1.2.2	Une caractérisation	17
1.2.3	Critère de dualité	19
1.3	Réseau et changement d'outils	22
1.3.1	Modélisation flot borné	23
1.3.2	Cas particulier du changement d'outils	24

Ce chapitre a pour objet l'étude des problèmes de distribution dans les réseaux. Dans la section 1.1, nous présentons le problème général de flot dans un réseau de distribution, ainsi que deux cas particuliers : le problème de flot de coût minimal dans un réseau avec coûts linéaires et le problème de transport. Dans la section 1.2, nous définissons et caractérisons les réseaux bien équilibrés. Cette partie est illustrée avec des problèmes de transport à coût linéaire. Finalement, en section 1.3, nous proposons une modélisation d'un problème pratique sous forme de flot dans un réseau. Ceci illustre la puissance des modèles de flot.

1.1 Problèmes de flots dans les réseaux

Les problèmes de flot font partie des problèmes les plus fondamentaux dans les réseaux. Il s'agit de faire transiter à moindre coût une marchandise dans un réseau afin de satisfaire des demandes à certains terminaux à partir d'offres disponibles à d'autres terminaux. Pour plus de détails, des exemples d'applications, et une étude approfondie des problèmes de flot on renvoie le lecteur au livre [Ahuja *et al.*, 1993].

Nous commençons par présenter un modèle général de problème de flot sur un réseau quelconque avec une structure de coût quelconque (section 1.1.1). Puis nous présentons

deux cas particuliers, le problème de flot à coût minimal (section 1.1.2) et le problème de transport (section 1.1.3).

1.1.1 Problème de flot général

On se place dans un réseau \mathcal{R} constitué par un ensemble N de nœuds et un ensemble A d'arcs. Ce réseau est modélisé par le graphe orienté $G = (N, A)$. Pour un nœud $i \in N$, on note $\delta^+(i)$ l'ensemble de ces successeurs, $\delta^+(i) = \{j \in N | (i, j) \in A\}$. De même on note, $\delta^-(j)$ l'ensemble des prédecesseurs d'un nœud $j \in N$, $\delta^-(j) = \{i \in N | (i, j) \in A\}$.

On considère la distribution d'un seul produit sur \mathcal{R} . Chaque nœud de ce réseau est étiqueté par a_i , la quantité de produit disponible au nœud $i \in N$. L'ensemble des nœuds N est partitionné en trois sous-ensembles S , T et D :

S est l'ensemble des nœuds sources (usines, production) et $a_i > 0$ pour $i \in S$,

T est l'ensemble des nœuds de transfert ou transbordement (*transshipment*) $a_i = 0$ pour $i \in T$,

D est l'ensemble des nœuds destinations (client, demandes) $a_i < 0$ pour $i \in D$.

On suppose que les offres et demandes sont équilibrées, i.e. $\sum_{i \in S \cup T \cup D} a_i = 0$. Les arcs du

réseau peuvent avoir une capacité u_{ij} . On note x_{ij} les variables de flots. Les coûts de transport entre deux nœuds peuvent être linéaires CV_{ij} , fixes CF_{ij} , ou décrits par une fonction plus complexe, généralement concave, $C(x_{ij})$.

Le but est de déterminer les quantités x_{ij} à transporter sur chaque route (i, j) pour satisfaire toutes les demandes de transport à moindre coût. Le problème de "flot" du réseau de distribution \mathcal{R} , noté $\mathcal{R}(a, C)$, se modélise sous la forme du programme mathématique décrit en figure 1.1.

$$\begin{aligned} & \text{Minimiser : } \sum_{ij \in A} C(x_{ij}), & (1.1) \\ \text{s.c.} & & \\ & \sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = a_i, \quad \forall i \in N, & (1.2) \\ & 0 \leq x_{ij} (\leq u_{ij}), \quad \forall (ij) \in A. & (1.3) \end{aligned}$$

FIGURE 1.1 – Modèle mathématique du problème de flot dans un réseau de distribution.

La contrainte 1.2 est une écriture compacte de la contrainte de conservation du flot au niveau des nœuds du réseau. Le cas échéant on retrouve la contrainte de capacité sur les arcs (deuxième partie de la contrainte 1.3).

Ce modèle est très général. La difficulté et la nature du problème dépendent de deux paramètres : le réseau et la fonction coût. Si la fonction coût est linéaire, on retrouve le problème linéaire de flot de coût minimal (section 1.1.2) dans un réseau. Si le graphe est biparti, il s'agit d'un problème de transport (section 1.1.3). Si la fonction coût comporte

une partie fixe et une partie variable linéaire, on a un problème de conception de réseau (section 3.3.4).

1.1.2 Problème de flot de coût minimal

Le problème de flot de coût minimal, est le cas de flot le plus simple : flot d'un seul produit avec des coûts de transport linéaires. Une extension de ce problème simple est présentée dans la section 1.2. La puissance des modèles de flots de coût minimal est illustrée dans la section 1.3.

Le problème de flot à coût minimal se modélise sous la forme du programme linéaire donné figure 1.2.

$$\text{Minimiser : } \sum_{ij \in A} CV_{ij} x_{ij}, \quad (1.4)$$

s.c.

$$\sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = a_i, \quad \forall i \in N, \quad (1.5)$$

$$0 \leq x_{ij} (\leq u_{ij}), \quad \forall (ij) \in A. \quad (1.6)$$

FIGURE 1.2 – Modèle mathématique linéaire du problème de flot de coût minimal.

Ce modèle (figure 1.2) est un programme linéaire : la fonction objectif et les contraintes sont linéaires. Le problème de flot de coût minimal est polynomial. Il reste polynomial même si l'on demande d'avoir une solution entière lorsque toutes les données (offres, demandes, capacités) sont entières. Plus précisément, la matrice de ce programme linéaire est totalement unimodulaire, par conséquent toutes les solutions de base sont entières ([Schrijver, 1998]). C'est une des raisons pour lesquelles les modèles de flot sont très utilisés.

Si l'on considère le transport de plusieurs produits avec des coûts linéaires sur le réseau \mathcal{R} avec capacité, on a alors un problème de multiflot (*multicommodity network flow problem*). Dans ce cas on introduit un indice supplémentaire k pour différencier les différentes marchandises. Ce problème sera présenté dans la section 3.3.2. La limitation de la capacité sur les arcs lie les différentes marchandises entre elles, et rend le problème difficile si l'on souhaite avoir un flot entier.

1.1.3 Problème de transport

Si l'on se place dans le cas particulier du graphe biparti complet ayant pour sommets $S \cup D$ on est alors dans le cas du problème de transport, *transportation problem (TP)*. Traditionnellement on note dans ce cas $a = \{a_1, a_2, \dots, a_m\}$, $a_i \geq 0$ les offres, et $b = \{b_1, b_2, \dots, b_n\}$, $b_j \geq 0$, les demandes. L'équilibre entre offre et demande s'écrit $\sum_{i=1}^m a_i =$

$\sum_{j=1}^n b_j$. Si C désigne la fonction coût, on note $\mathcal{TP}(a, b, C)$ une instance du problème de transport. Un réseau de ce cas particulier est représenté figure 1.3, la formulation du problème est donnée figure 1.4.

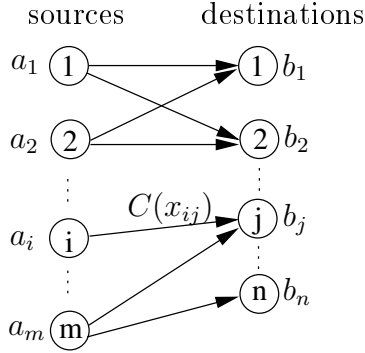


FIGURE 1.3 – Graphe biparti complet du TP.

$$\begin{aligned} & \text{Minimiser : } \sum_{ij \in A} C(x_{ij}), & (1.7) \\ & \text{s.c.} \\ & \sum_{j=1}^n x_{ij} = a_i, \quad \forall i = 1..m, & (1.8) \\ & \sum_{i=1}^m x_{ij} = b_j, \quad \forall j = 1..n, & (1.9) \\ & 0 \leq x_{ij} (\leq u_{ij}), \quad \forall i, j. & (1.10) \end{aligned}$$

FIGURE 1.4 – Modélisation du TP linéaire.

Le problème de transport avec coûts linéaires a été formulé par [Hitchcock, 1941]. En 1951, [Dantzig, 1951] donne la formulation standard en programme linéaire et applique l'algorithme du simplexe. Dès lors, le problème de transport est largement présenté dans de nombreux livres de Recherche Opérationnelle.

Remarque : Étant donné que les offres et les demandes sont équilibrées ($\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$) une des contraintes est redondante. Une solution de base contient donc $m + n - 1$ variables.

Proposition 1.1.1. Si la fonction coût est concave, alors il existe une solution de base qui est optimale. Cette solution est un arbre. Plus précisément, à chaque X solution réalisable de TP , on associe un graphe biparti G_x défini comme suit : G_x contient l'arc (i, j) si et seulement si $x_{ij} > 0$. On a alors une équivalence entre X est une arête du polytope de TP et G_x est une forêt. Si X est non dégénérée alors G_x est un arbre, sinon on peut relier la forêt G_x par des variables de base à valeur zéro pour obtenir un arbre dégénéré.

Définition. Une solution du TP est dégénérée si au moins une variable de base a la valeur zéro.

Dans l'exemple figure 1.5, nous nous familiarisons avec la notation de tableau de transport, *transportation tableau*. Un tableau décrit de façon synthétique les offres, les demandes et une solution d'un problème de transport. Pour l'instance donnée, nous donnons trois

1.2 Réseau bien équilibré et paradoxe de transport

solutions différentes. Ces solutions sont des arbres. La première est non dégénérée, la deuxième a une dégénérescence, la dernière a deux dégénérescences.

Exemple.

Offre $a = (4, 3, 5, 5)$

Demande $b = (4, 3, 6, 4)$

$b_j \rightarrow$	4	3	6	4	4	3	6	4	4	3	6	4
a_i	4	3		1	4	4		0	4	4		0
\downarrow	3			3	3	3		3	3	3		0
	5			5	5		5		5		5	
	5	1	3	1	5	3	1	1	5	5	1	4

FIGURE 1.5 – Problème de transport et solution avec zéro, une et deux dégénérescences.

La nature et la complexité du problème de transport dépendent du type de la fonction coût. Dans le cas d'une fonction linéaire, le problème de transport est un problème de flot de coût minimal dans un graphe biparti. Ce problème sera source de nombreux exemples illustratifs dans la section 1.2. Dans la section 2.1, nous étudions le cas particulier du problème de transport avec coût fixe. Nous verrons que même dans le cas le plus simple possible (un coût fixe constant identique sur tous les arcs) ce problème est déjà difficile (\mathcal{NP} -difficile).

1.2 Réseau bien équilibré et paradoxe de transport

On se place dans le cas le plus simple du problème de flot de coût min. C'est le problème du réseau de distribution $\mathcal{R}(a, C)$, avec C fonction coût linéaire. La modélisation sous forme de programme linéaire est rappelée ci-dessous. On note $F(a, C)$ la valeur de la fonction objectif. Pour faciliter la lecture, la fonction coût linéaire C correspond à une matrice des coûts C , dont on note c_{ij} les coefficients.

$$\text{Minimiser : } \sum_{ij \in A} c_{ij} x_{ij}, \quad (1.11)$$

s.c.

$$\sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji} = a_i, \quad \forall i \in N, \quad (1.12)$$

$$0 \leq x_{ij} (\leq u_{ij}), \quad \forall (ij) \in A. \quad (1.13)$$

Ce problème est polynomial. Le réseau étant fixe, on peut cependant se demander si le réseau est bien équilibré (*well-balanced network*) autrement dit si les "bonnes quantités sont au bon endroit". L'augmentation de certaines productions et de certaines demandes ne pourrait-elle pas permettre de réduire les coûts de transport? Ce phénomène appelé paradoxe de transport, ou *more-for-less paradox*, est l'objet de cette section. Le paradoxe de transport est défini formellement dans la section 1.2.1. Une caractérisation des réseaux bien

équilibrés est donnée dans la section 1.2.2. Finalement, la section 1.2.3 apporte l'éclairage de la dualité sur les réseaux bien équilibrés. Les exemples illustratifs sont pris dans la classe des problèmes de transport.

1.2.1 Définition d'un réseau bien équilibré

Définition. Soit $a = (a_i), i \in S \cup T \cup D$ tel que $\sum_{i \in S \cup T \cup D} a_i = 0$ et $C = (c_{ij})$ (matrice des coûts) une instance du problème de réseau de distribution avec coût linéaire noté $\mathcal{R}(a, C)$. $F(a, C)$ est le coût d'une solution optimale. Soit $a' = (a'_i)$ tel que $\forall i \in S \cup D, |a'_i| \geq |a_i|$ vérifiant $\sum_i a'_i = 0$ une nouvelle instance du problème de réseau de distribution noté $\mathcal{R}(a', C)$ sur le même réseau avec les mêmes coûts.

Si ce nouveau problème a une solution optimale de moindre coût que le problème initial, i.e $F(a, C) > F(a', C)$ alors on dit que le réseau de distribution $\mathcal{R}(a, C)$ n'est pas bien équilibré. Plus précisément on est en situation de *paradoxe* aussi appelé *more-for-less paradox* : on transporte plus et c'est moins cher !

Exemple. Nous présentons ici (figure 1.6) l'exemple de [Deineko *et al.*, 2003] pris dans le cadre du cas particulier du problème de transport avec coûts linéaires. Le réseau sous-jacent est un graphe biparti complet.

offre $a = (0, 1, 1)$

demande $b = (1, 1, 0)$

coûts de transport : $C = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 1 & 2 \\ 4 & 2 & 1 \end{pmatrix}$

solution optimale :

		1	1	0								
0	<table style="border-collapse: collapse;"> <tr><td style="padding: 2px 10px;"></td><td style="padding: 2px 10px;"></td><td style="padding: 2px 10px;"></td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;"></td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;"></td><td style="padding: 2px 10px;">1</td></tr> </table>				1	1		1		1	coût = 4.	
1		1										
1		1										
1	1	1										

paradoxe :

		1	1	1								
1	<table style="border-collapse: collapse;"> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;"></td><td style="padding: 2px 10px;"></td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;"></td></tr> <tr><td style="padding: 2px 10px;">1</td><td style="padding: 2px 10px;"></td><td style="padding: 2px 10px;">1</td></tr> </table>	1			1	1		1		1	coût = 3.	
1												
1		1										
1		1										
1	1	1										

$a' = (1, 1, 1)$,
 $b' = (1, 1, 1)$.

FIGURE 1.6 – Un exemple de matrice de coût, de demandes et d'offres qui produisent une situation de paradoxe de transport.

Dans l'exemple figure 1.6, l'augmentation des quantités transportées permet une réduction des coûts. On satisfait toutes les demandes et on transporte plus et c'est moins cher ! C'est le paradoxe de transport.

Remarque. Les nœuds de transshipment, peuvent être vus comme des nœuds d'offre (avec

offre nulle), ou comme des nœuds de demandes (avec demande nulle). Par soucis de clarté, les nœuds de transshipments sont des nœuds à part, sur lesquels il ne peut y avoir ni offre, ni demande, et on demande que $a'_i = 0, \forall i \in T$.

Il n'est pas facile de savoir à qui attribuer la découverte du paradoxe ni quand elle a été faite. La plupart des articles citent à [Charnes et Klingman, 1971] et [Szwarc, 1971] comme étant les premiers articles sur le sujet. Mais selon [Deineko *et al.*, 2003] le paradoxe de transport était déjà utilisé par Alison Doig dans les examens de la *London School of Economics* vers 1959 (sans publication sur le sujet). Ce paradoxe de transport est étudié sur le point théorique dans [Charnes et Klingman, 1971] et [Charnes *et al.*, 1987]. [Robb, 1990] donne une explication intuitive de l'apparition du paradoxe.

C'est la structure des matrices de coût qui engendre la situation de paradoxe. La question de savoir si le problème de transport $\mathcal{TP}(a,b,C)$ produit le paradoxe a été traitée dans la littérature aussi bien du point de vue théorique que du point de vue algorithmique [Charnes et Klingman, 1971, Finke, 1981, Finke, 1983]. Les expériences faites par [Finke, 1978] montrent que la situation de paradoxe est plutôt fréquente. Il génère des instances aléatoire du problème de transport de taille 100×100 , et montre qu'en moyenne la levée des situation de paradoxe permet une réduction des coûts de 18.6% en moyenne, avec 20.6% de surplus de marchandises transportées.

La question plus générale traitée dans [Deineko *et al.*, 2003] est de savoir si une matrice de coût C peut engendrer une situation de paradoxe. Plus précisément, on dit que la matrice C ne peut pas engendrer de situation de paradoxe (*immune against the paradox*), si, quelles que soient les offres a et demandes b , le problème $\mathcal{TP}(a,b,C)$ ne produit pas de paradoxe. De telles matrices de coûts sont caractérisées par un critère assez simple dans [Deineko *et al.*, 2003]. Ce critère est énoncé dans le théorème 1.2.1.

Théorème 1.2.1. La matrice C des coûts origine-destination ne peut jamais engendrer de paradoxe, si et seulement si, pour toutes origines q, s et pour toutes destinations r, t avec $q \neq s$ et $r \neq t$ l'inégalité suivante est vérifiée :

$$c_{qt} + c_{sr} \geq c_{qr}$$

Cette condition est plutôt restrictive, ce qui confirme le fait que la situation de paradoxe est fréquente [Finke, 1978].

1.2.2 Une caractérisation

Le but de cette section est de caractériser les réseaux bien équilibrés.

Définition. Deux problèmes d'optimisation sont équivalents s'il y a bijection entre leurs solutions admissibles qui préserve l'optimalité.

On considère le problème de réseau de distribution $\mathcal{R}(a, C)$ avec $F(a, C)$ pour valeur à l'optimum. Soit \mathcal{R}' le réseau construit à partir de \mathcal{R} en ajoutant des connexions retours destination vers source $(i, j), i \in D, j \in S$, de capacité infinie $u_{ij} = \infty$ et de coût $c_{ij} = 0$. On note x'_{ij} les variables de flux sur \mathcal{R}' . On définit alors les deux problèmes suivants :

- **problème** $\mathcal{P}(a', C)$: trouver le meilleur surplus a' qui minimise le coût $F(a', C)$ pour le problème de réseau de distribution $\mathcal{R}(a', C)$.
- **problème** $\mathcal{P}'(a, C)$: trouver le flot de coût minimal pour le problème du réseau de distribution $\mathcal{R}'(a, C)$, on note $F'(a, C)$ la valeur à l'optimum.

Théorème 1.2.2. Les problèmes $\mathcal{P}(a', C)$ et $\mathcal{P}'(a, C)$ sont équivalents et ont la même valeur à l'optimum $F(a', C) = F'(a, C)$.

Démonstration. Soit x' une solution admissible de $\mathcal{P}'(a, C)$ de valeur $F'(a, C)$. On construit un surplus a' de la façon suivante : pour toute source $i \in S$, $a'_i = a_i + \sum_{j \in \delta^-(i)} x'_{ji}$, pour

toute destination $j \in D$, $a'_j = a_j + \sum_{i \in \delta^+(j)} x'_{ji}$ et pour les nœuds de transbordement $i \in T$,

$a'_i = 0$. Ceci définit $\mathcal{P}(a', C)$ qui a pour solution optimale $x_{ij} = x'_{ij} + x'_{ji}$ de coût $F(a', C)$. Cette transformation définit une injection de l'ensemble des solutions de $\mathcal{P}'(a, C)$ dans l'ensemble des solutions de $\mathcal{P}(a', C)$ et par construction les valeurs à l'optimum sont les mêmes $F(a', C) = F'(a, C)$.

Réciproquement, soit a' solution de $\mathcal{P}(a', C)$. $|a'| \geq |a|$ composante par composante et l'inégalité est stricte pour au moins une composante. On souhaite construire une solution de $\mathcal{P}(a', C)$. On considère alors le supplément $a' - a$ et on se place dans le réseau restreint construit sur les nœuds vérifiant $|a' - a| > 0$. On échange alors origine et destination et on obtient un nouveau problème de transport. La solution est un arbre, on la note $x_{j,i}$.

Dans le réseau \mathcal{R}' , on fixe les variables de retour de la façon suivante, $x'_{ji} = x_{j,i}$ sur les nœuds du réseau restreint défini précédemment et $x'_{ji} = 0$ sinon. On complète cette solution par $x'_{ij} = x_{ij}$ solution du problème de transport initial sur \mathcal{R} . On obtient ainsi une solution optimale dans \mathcal{R}' . Cette transformation définit une injection de l'ensemble des solutions de $\mathcal{P}(a', C)$ dans l'ensemble des solutions de $\mathcal{P}'(a, C)$ et par construction les valeurs à l'optimum sont les mêmes $F(a', C) = F'(a, C)$. \square

Corollaire. En situation de paradoxe, une fois un paradoxe levé par augmentation des quantités transportées, on obtient, par construction, une solution dégénérée.

Théorème 1.2.3. Le réseau de distribution $\mathcal{R}(a, C)$ n'est pas bien équilibré, si et seulement si pour toute solution optimale dans le réseau étendu $\mathcal{R}'(a, C)$ il existe un flux retour non nul, i.e. qu'il existe $d \in D$ et $s \in S$ tels que $x'_{d,s} > 0$.

Démonstration. On raisonne par contraposition. Le réseau $\mathcal{R}(a, C)$ est bien équilibré si, et seulement si, $F(a', C) = F(a, C)$ ce qui équivaut à l'existence d'une solution dans $\mathcal{R}'(a)$ avec $x'_{d,s} = 0, \forall d \in D, \forall s \in S$ (on prend $x'_{ij} = x_{ij}$ dans \mathcal{R} et $x'_{d,s} = 0, \forall d \in D, \forall s \in S$). \square

Exemple. On considère le cas particulier du problème de transport $\mathcal{TP}(a, b, C)$, avec a, b , et C donnés ci-après. On donne la solution optimale de ce problème, ainsi que la solution optimale dans le réseau avec flux retour. Celle-ci met en évidence une situation de paradoxe.

Offres $a = (6, 6, 6)$

Demandes $b = (4, 5, 4, 5)$

Matrice des coûts : $C = \begin{pmatrix} 3 & 6 & 1 & 5 \\ 7 & 9 & 2 & 7 \\ 2 & 4 & 2 & 1 \end{pmatrix}$

Solution optimale :

		4	5	4	5	
6	4	2				coût = 59
6		2	4			
6		1			5	

Solution optimale dans réseau
avec flux retour :
(flot négatif si flux retour)

		4	5	4	5	
6	4	2				coût = 53
6			6			
6		3	-2		5	

Meilleure solution avec surplus :
 $a' = (6, 6, 8)$
 $b' = (4, 5, 6, 5)$

		4	5	6	5	
6	4	2	0	0		coût = 53
6	0	0	6	0		
8	0	3	0	5		

Une solution du problème de
transport avec surplus :
(solution dégénérée)

		4	5	6	5	
6	4	2	0			coût = 53
6			6			
8		3			5	

FIGURE 1.7 – Un exemple de situation de paradoxe de transport levé via flux retour.

1.2.3 Critère de dualité

Dans cette sous-section, on présente une condition nécessaire pour l'apparition du paradoxe, cette condition étant exprimée à l'aide des variables duales. L'étude se place au niveau du réseau, comme dans [Finke, 1981].

On note u_i les variables duales, le problème dual du problème $\mathcal{R}(a, C)$ est écrit figure 1.8. Il s'agit en réalité de la formulation duale qui utilise le théorème des écarts complémentaires et non du dual "direct".

$$\text{Maximiser : } \sum_{i \in S \cup D} u_i a_i, \quad (1.14)$$

s.c.

$$u_i - u_j \leq c_{ij}, \quad \text{pour les } x_{ij} = 0 \text{ (hors base), } i \in S, j \in D, \quad (1.15)$$

$$(u_i - u_j \geq c_{ij}, \quad \text{pour les } x_{ij} = u_{ij} \text{ (à capacité)),} \quad (1.16)$$

$$u_i - u_j = c_{ij}, \quad \text{pour les autres arcs } (ij) \text{ (dans la base),} \quad (1.17)$$

$$u_i \in \mathbb{R}. \quad (1.18)$$

FIGURE 1.8 – Problème dual du problème $\mathcal{R}(a, C)$.

La contrainte (1.16) est à prendre en compte si les arcs ont une capacité u_{ij} .

Théorème 1.2.4. Soit x_{ij} solution-arbre optimale dans $\mathcal{R}(a, C)$,

1. Si la différence de potentiel $u_d - u_s \leq 0$ pour tous les couples $(s, d) \in S \times D$, alors le réseau $\mathcal{R}(a, C)$ est bien équilibré.
2. Une condition nécessaire pour être en situation de paradoxe est que la différence de potentiel $u_d - u_s > 0$ pour un certain couple (s, d) .
3. Dans le cas particulier d'une solution-arbre non dégénérée la condition $u_d - u_s > 0$ pour un certain couple (s, d) est nécessaire et suffisante pour avoir une situation de paradoxe.

Démonstration. 1. Si $u_d - u_s \leq 0 = c_{ds}$ alors la solution dans $\mathcal{R}(a, C)$ reste optimale pour \mathcal{R}' .

2. Ce point est la contraposée du premier point.

3. Si $u_d - u_s > 0$ pour un certain couple (s, d) , on continue les itérations par l'algorithme du simplexe dans les réseaux avec (d, s) pour pivot. Sans dégénérescence l'amélioration est stricte, donc il y a paradoxe. □

Dans la suite nous présentons deux exemples illustratifs du point (2) du théorème 1.2.4. Dans les deux cas on a affaire à un problème dégénéré qui vérifie la condition nécessaire $u_d - u_s > 0$ pour un certain couple (s, d) . Dans le premier cas (figure 1.9), on a une situation de paradoxe bien que le problème soit dégénéré. Dans le second (figure 1.10) la dégénérescence est mal placée et empêche une réduction de coût stricte, il n'y a pas paradoxe. La condition nécessaire (2), n'est donc pas une condition suffisante pour être en situation de paradoxe.

Exemple. On peut avoir une solution dégénérée et avoir paradoxe.

Pour le problème de transport on note comme d'habitude les variables duales $u_j = -v_j$ pour $j \in D$. La condition (2) du théorème 1.2.4 s'écrit alors $-v_d - u_s > 0$, ce qui est équivalent à $v_d + u_s < 0$.

1.2 Réseau bien équilibré et paradoxe de transport

Offre $a = (10, 6, 15, 4)$

Demande $b = (5, 14, 10, 6)$

$$\text{Matrice des coûts } C = \begin{pmatrix} 5 & 8 & 7 & 6 \\ 6 & 10 & 5 & 5 \\ 7 & 15 & 3 & 16 \\ 15 & 21 & 8 & 18 \end{pmatrix}$$

$$\begin{array}{l} \text{solution optimale,} \\ \text{et variables duales :} \end{array} \quad \begin{array}{l} u_1 = 0 \\ u_2 = 2 \\ u_3 = 7 \\ u_4 = 9 \end{array} \quad \begin{array}{l} 10 \\ 6 \\ 15 \\ 4 \end{array} \quad \begin{array}{l} v_1 = 5 \\ v_2 = 8 \\ v_3 = -4 \\ v_4 = 3 \end{array} \quad \begin{array}{l} 5 \\ 14 \\ 10 \\ 6 \end{array} \quad \begin{array}{|c|c|c|c|} \hline & 10 & & \\ \hline & 0 & & 6 \\ \hline 5 & 4 & 6 & \\ \hline & & 4 & \\ \hline \end{array} \quad \text{coût} = 255$$

Meilleure solution avec sur-
plus :

$a' = (14, 6, 15, 4)$

$b' = (5, 14, 14, 6)$

$$\begin{array}{l} 14 \\ 6 \\ 15 \\ 4 \end{array} \quad \begin{array}{l} 5 \\ 14 \\ 14 \\ 6 \end{array} \quad \begin{array}{l} v_1 = 5 \\ v_2 = 8 \\ v_3 = -4 \\ v_4 = 3 \end{array} \quad \begin{array}{l} 5 \\ 14 \\ 10 \\ 6 \end{array} \quad \begin{array}{|c|c|c|c|} \hline & 14 & & \\ \hline & 0 & & 6 \\ \hline 5 & 0 & 10 & \\ \hline & & 4 & \\ \hline \end{array} \quad \text{coût} = 239$$

FIGURE 1.9 – Un exemple de problème de transport dégénéré avec situation de paradoxe de transport.

Exemple. On peut avoir $u_s + v_d < 0$ sans pour autant pouvoir avoir une réduction de coût stricte.

Offre $a = (7, 10)$

Demande $b = (7, 10)$

$$\text{Matrice des coûts } C = \begin{pmatrix} 5 & 15 \\ 16 & 6 \end{pmatrix}$$

$$\begin{array}{l} \text{Solution optimale} \\ u_1 = -5 \\ u_2 = 6 \end{array} \quad \begin{array}{l} 7 \\ 10 \end{array} \quad \begin{array}{l} v_1 = -10 \\ v_2 = 0 \end{array} \quad \begin{array}{l} 7 \\ 10 \end{array} \quad \begin{array}{|c|c|} \hline 7 & \\ \hline 0 & 10 \\ \hline \end{array} \quad \text{coût} = 95$$

$u_1 + v_2 = -5 < 0$

Solution optimale

dans réseaux avec flux retour

$u_1 + v_2 = -5 < 0$, mais pas de situation de paradoxe.

$v_1 = -10$ $v_2 = 0$

7 10

$$\begin{array}{l} u_1 = -5 \\ u_2 = 6 \end{array} \quad \begin{array}{l} 7 \\ 10 \end{array} \quad \begin{array}{l} v_1 = -10 \\ v_2 = 0 \end{array} \quad \begin{array}{l} 7 \\ -0 \\ 10 \end{array} \quad \begin{array}{|c|c|} \hline 7 & -0 \\ \hline 0 & 10 \\ \hline \end{array}$$

coût = 95

Pas de gain strict.

FIGURE 1.10 – Un exemple de problème de transport dégénéré qui vérifie la condition (2) mais sans paradoxe.

1.3 Réseau et changement d'outils

Dans cette dernière section, nous allons modéliser un problème de repositionnement de véhicules via un modèle de flot. Puis nous liions ce problème au problème de changement d'outils. Le problème étudié n'a pas de lien direct avec les applications industrielles, il s'agit d'une étude théorique annexe.

En fin d'année, les wagons de trains se retrouvent dispatchés sur la France (40000 gares), et des wagons sont disponibles dans différentes villes. Certains de ces wagons vont devoir être déplacés afin de satisfaire les besoins en wagons pour la première semaine de l'année suivante. Puis de nouveaux déplacements auront lieu pour satisfaire les demandes de la deuxième semaine...

La situation initiale est donc une distribution de wagons donnée sur un réseau à l'instant t . A l'instant $t + 1$ on doit satisfaire des nouvelles demandes en wagons. Pour cela, on doit déplacer certains wagons et on obtient une nouvelle distribution. De même à $t + 2$, on a de nouvelles demandes ... Dans un premier temps, on suppose connaître toutes les demandes futures.

Les coûts de repositionnement sont soit uniformes (les déplacements coûtent tous le même tarif), soit non uniformes. On note c_{ij} le coût de déplacement d'un wagon de la gare i à la gare j . Le but est de déplacer les wagons de sorte à minimiser la somme des coûts de déplacement sur l'ensemble des périodes.

Exemple :

	A	B	C	D	E
wagons disponibles	4	2	5	6	2
demande période t	0	4	8	1	1
demande période $t + 1$	1	2	3	2	0
demande période $t + 2$	2	2	3	4	2

Dans l'exemple ci-dessus, 4 wagons sont entreposés dans la ville A , 2 dans la ville B , 5 dans la ville C , ... ; ces wagons vont devoir être déplacés afin de satisfaire les besoins futurs. Pour la période suivante la ville A n'a besoin d'aucun wagon, la ville B a besoin de 4 wagons, la ville C de 8, ...

Dans la figure ci-dessous on présente un exemple des déplacements possibles (avec 14 déplacements) qui permet de satisfaire les demandes. Pour chaque sommet, à chaque période on note le nombre de wagons étant garés à ce sommet à coté du nom du sommet. Les flèches en trait plein indiquent les wagons qui sont déplacés. Les traits en pointillés, indiquent que les autres wagons ne bougent pas.

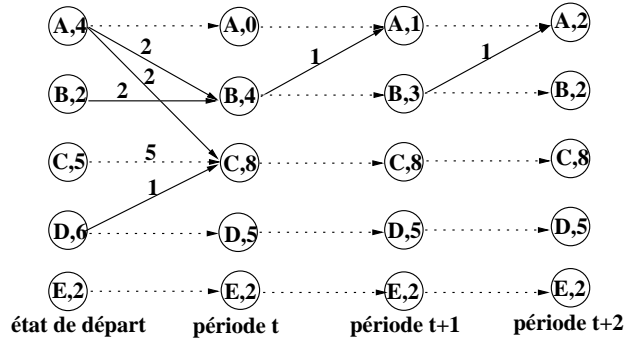


FIGURE 1.11 – Un exemple des déplacements possibles.

Étant donné des coûts de déplacement, uniformes ou non uniformes, on cherche à trouver le meilleur déplacement des wagons sur l'ensemble des périodes de sorte à minimiser le coût global des déplacements sur l'ensemble des périodes.

Remarque : Dans le cas particulier où l'inégalité triangulaire est vérifiée par les coûts, la solution qui minimise l'ensemble des coûts de déplacement minimise aussi le nombre de déplacements.

1.3.1 Modélisation flot borné

Le problème peut être modélisé sous forme de flot borné (capacité et borne inférieure sur les arcs), *directed flow with bounds*. Résoudre le problème revient à trouver le flot maximal de coût minimal dans un graphe sans cycles. Il s'agit donc d'un problème polynomial (*circulation problem*). La construction de ce graphe ainsi que les capacités et bornes inférieures sur le flot sur les différents arcs sont présentées sur l'exemple figure 1.12.

Exemple.

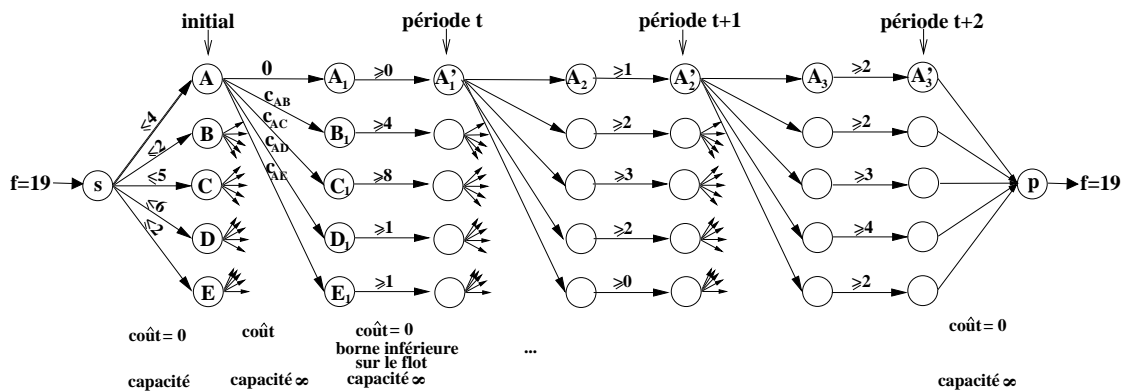


FIGURE 1.12 – Graphe *directed flow with bounds*.

De façon générale, le réseau construit est constitué d'une source s , d'un puit p , et de plusieurs niveaux de sommets. Pour le premier niveau, on a une copie des différentes gares A, B, \dots

Ce premier niveau correspond à la configuration initiale, i.e. aux wagons disponibles dans les différentes gares. Pour garantir la bonne configuration initiale, chacun des sommets A, B, \dots sont reliés à la source s par un arc de coût nul et de capacité égale à leur disponibilité. Ces arcs sont les seuls du réseau à être limités en capacité, dans un flot maximum ils vont donc être saturés ce qui donne le bon nombre de wagons disponibles initialement.

Les niveaux suivant correspondent aux différentes périodes. Pour chacun de ces niveaux on a deux copies des différentes gares, une qui correspond aux arrivées A_1, B_1, \dots et une A'_1, B'_1, \dots qui correspond aux départs. De sorte à satisfaire les besoins à chaque période on force un flot minimal, égal à la demande, entre ces deux copies de sommets.

Entre deux niveaux, on a un graphe biparti complet avec des arcs $(i'_\tau, j_{\tau+1})$ de capacité infinie et de coût c_{ij} . Ces arcs permettent le déplacement des wagons au coût correspondant, en particulier un wagon peut rester sur place à un coût nul.

Les sommets de départ du dernier niveau sont reliés au puit par des arcs de coût nul et de capacité infinie.

Remarques : Dans certains cas particuliers on retrouve le problème de transport.

- **Une seule période.** Dans le cas le plus simple, on a a_i wagons disponibles dans les villes i sources ($i \in S$), et des demandes de $b_j = -a_j$ wagons pour les destinations j ($j \in D$). Si de plus la demande ne se répète pas dans le temps (une seule période), on est dans le cas classique du problème de transport. Le problème se résout alors de façon polynomiale comme flot maximal de coût min.
- **Offre et demande équilibrées à chaque période.** Si l'offre et toutes les demandes sont équilibrées, i.e qu'à chaque instant la somme de toutes les demandes est égale à l'offre initiale, alors il n'y a pas d'interférence entre les différentes périodes. En effet, l'avenir n'intervient pas dans les choix de déplacement puisque à chaque période aucun wagon ne reste libre. Ce problème revient à résoudre des problèmes de transport les uns à la suite des autres, on peut formuler le problème sous forme de flot maximal à coût minimal dans un réseau avec capacité, ces capacités correspondent aux demandes.

1.3.2 Cas particulier du changement d'outils

Dans cette section, nous présentons le problème de changement d'outils, et proposons une interprétation du problème de repositionnement de wagons comme problème de changement d'outils dans certains cas particuliers.

Présentation générale du problème de changement d'outils. Le problème de changement d'outils apparaît dans des systèmes de production utilisant des machines à commandes numériques (*NC-machines*) pour la fabrication de différentes pièces [Privault et

Finke, 1993]. Ces *NC-machines* sont flexibles : elles sont munies d'un dispositif automatique de changement d'outils. Les m outils correspondant à la machine sont disponibles, d'une part dans un magasin d'outils de capacité k , et d'autre part dans une aire de stockage contenant les outils restant. On souhaite usiner une séquence de n pièces sur cette machine. Pour l'usinage d'une pièce, tous les outils nécessaires doivent se trouver dans le magasin d'outils. Certains outils vont devoir sortir du réservoir et être insérés dans le magasin, et si le magasin est plein, les outils non utilisés devront retourner dans le réservoir. Ces opérations de changement d'outils ont un coût, le but étant de minimiser l'ensemble des coûts de changement d'outils pour l'usinage de la séquence de pièces.

Ce problème peut également s'interpréter comme un problème d'allocation de mémoire, modélisé, en informatique, par le problème de k -serveur [Privault et Finke, 2000]. Ce problème apparaît sur des systèmes à 2 niveaux de mémoire : un ordinateur a des informations stockées sur m pages de mémoires, k de ces pages sont à accès rapide (*fast-access*), les autres sont à accès lent (*slow-access*). Exécuter un programme nécessite l'accès à une séquence de n pages. Si une page i demandée est dans la mémoire *slow-access*, il faut l'échanger avec une page j se trouvant dans la mémoire *fast-access* avant de pouvoir avoir accès à ses données. Un tel échange a un coût de c_{ij} , le but est de minimiser le coût total des échanges de pages nécessaires à l'exécution du programme. Ici les pages correspondent aux outils, la mémoire *fast-access* correspond au magasin d'outils, et les pièces à usiner correspondent aux n pages du programme. Une seule page est demandée à la fois (c'est le problème de *paging*), on peut l'étendre à plusieurs pages demandée à la fois (*k-server problem with bulk request*).

Interprétation de notre problème comme problème de changement d'outils.

On considère la cas particulier des offres et demandes booléennes (unitaires ou nulles). Dans ce cas le problème de repositionnement se ramène au problème de changement d'outils.

L'ensemble des outils disponibles correspond à l'ensemble des gares. Le magasin d'outils est composé par les villes qui ont un wagon disponible, déplacer un wagon d'une ville i vers une ville j correspond à faire entrer j dans le magasin d'outils et à ôter i , cette action a un coût c_{ij} . Chaque période correspond à l'usinage d'une pièce.

Exemple. On considère le problème de déplacement de wagons avec offres et demandes booléennes suivant :

	A	B	C	D	E
wagons disponibles	1	0	0	1	1
demande période t	0	1	1	0	0
demande période $t + 1$	0	0	0	0	1
demande période $t + 2$	0	1	0	1	0

En termes de changements d'outils le problème s'interprète de la façon suivante. On dispose en tout de 5 outils, A, B, \dots, E mais le magasin d'outils est de capacité 3. Au

départ ce magasin est rempli avec une certaine configuration initiale A, D, E . Les pièces à usiner et les outils nécessaires à leur usinage sont :

Pièces	outils nécessaires
1	B, C
2	E
3	B, D

On peut traduire ce problème dans le réseau construit précédemment, cette interprétation est donnée figure 1.13. Pour plus de lisibilité, les sommets correspondant aux outils qui doivent être dans la magasin sont grisés. La seule différence par rapport au réseau précédent est que le flot n'est pas seulement contraint inférieurement sur les arcs horizontaux, mais que sa valeur est fixée. Une alternative sur ces arcs horizontaux pour lesquels le flot est fixé à 1 est de limiter la capacité à 1, et de mettre un coût négatif de très grande valeur absolue sur ces arcs afin de "forcer" la saturation de ces arcs.

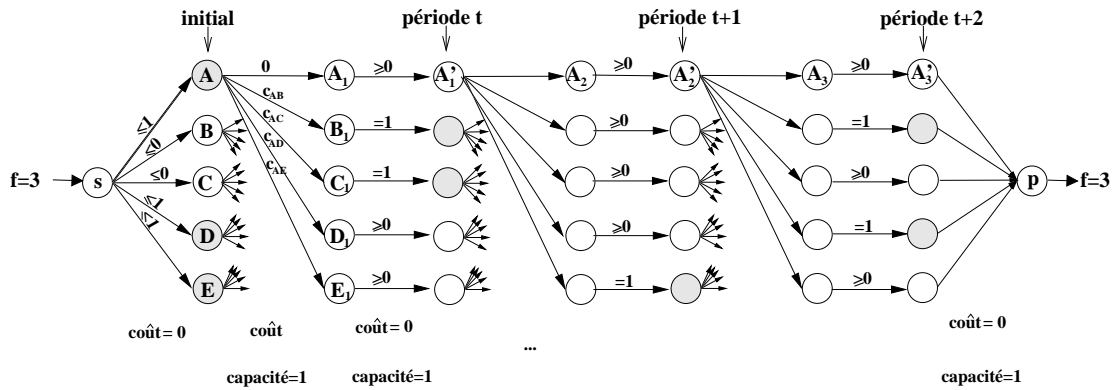


FIGURE 1.13 – Graphe *directed flow with bounds* dans le cas particulier du changement d'outils.

Notre formulation comme un problème de flot maximal à coût minimal est de taille $\mathcal{O}(MN)$, où N est le nombre de périodes (correspond au nombre de pièces) et M est le nombre de gares (correspondant au nombre total d'outils disponibles).

Une autre formulation du problème de changement d'outils comme un problème de flot maximal à coût minimal dans un réseau sans cycles est proposée dans [Privault et Finke, 1993]. Cette formulation est en $\mathcal{O}(N^2)$, où N est le nombre de pièces à usiner.

Le problème de changement d'outils est une partie d'un problème plus général de gestion d'outils. Celui-ci a deux aspects : c'est d'abord un problème d'ordonnancement, il faut trouver un ordre d'usinage des pièces ; puis il faut trouver la séquence des changements d'outils. Ces deux problèmes ne sont pas indépendants. Le problème général, comportant les deux aspects, est le problème dit *on-line*. Il est \mathcal{NP} -complet, même dans le cas le plus simple d'un magasin d'outils de capacité 2 [Crama *et al.*, 1994]. L'aspect ordonnancement

n'a pas de sens pour notre problème puisque les périodes ne sont pas interchangeables. Notre problème est *off-line*.

Coûts vérifiant l'inégalité triangulaire. On suppose de plus que les coûts de déplacements vérifient l'inégalité triangulaire. Les seuls déplacements possibles sont alors vers les villes de demande. Ainsi les seuls arcs du réseaux sont les arcs horizontaux (le wagon reste sur place), et les arcs d'origines A'_i et ayant pour destination les A_j de la période suivante qui demandent un wagon. Ceci est représenté dans la figure 1.14.

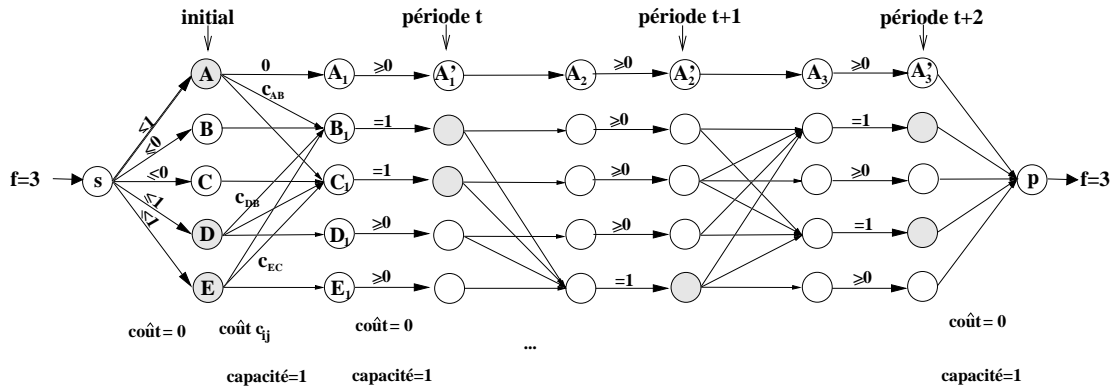


FIGURE 1.14 – Graphe *directed flow with bounds* simplifié dans le cas particulier de coûts vérifiant l'inégalité triangulaire.

Coûts uniformes. On suppose de plus que les coûts de déplacement sont uniformes. On a donc pour tout couple (i, j) de villes : $c_{ij} = 0$ si $i = j$ et $c_{ij} = 1$ si $i \neq j$. Le problème revient alors au problème de changement d'outils avec coût uniforme (figure 1.15). Dans ce cas minimiser les coûts de déplacement revient à minimiser le nombre de déplacements. Le problème se résout à l'optimum en utilisant la règle KTNS (*Keep Tool Needed Soonest*). Si cette règle s'énonce simplement, la preuve de son optimalité n'est pas aisée [Crama *et al.*, 1994], [Tang et Denardo, 1988]. On va montrer qu'avec le flot maximal de coût minimal dans notre réseau on retrouve la règle KTNS.

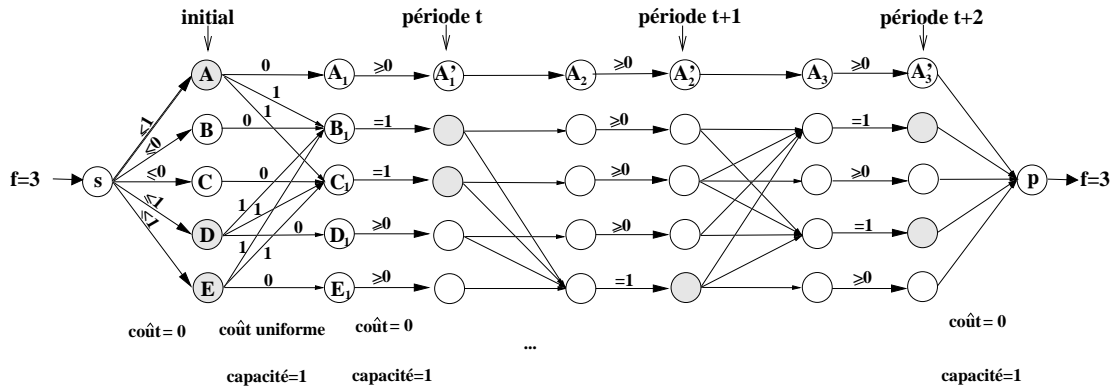


FIGURE 1.15 – Graphe *directed flow with bounds* dans le cas particulier du changement d'outils avec coûts uniformes.

Une solution de flot maximum, avec une valeur de flot égale à f , est un ensemble de f chemins sommets disjoints de s à p . La somme C des coûts de ces chemins est égale au nombre total de changements. Dans la figure 1.16, on représente une solution, les cases les plus foncées représentant les demandes, les cases légèrement grisées correspondant aux wagons qui restent sur place. A chaque période, exactement f cases sont grisées claires ou foncées. On a ainsi, à chaque instant, la distribution des wagons sur le réseau. Dans le cas de l'exemple $f = 3$, et la solution est de coût exactement 3, en effet elle comporte 3 changements (arcs non horizontaux).

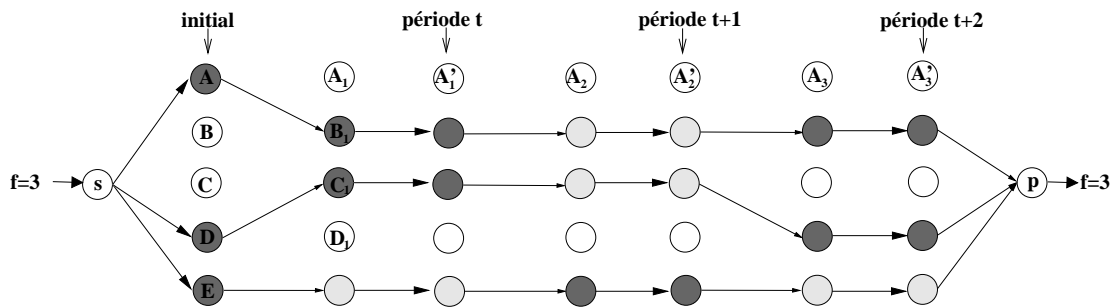


FIGURE 1.16 – Solution *directed flow with bounds* dans le cas particulier du changement d'outils avec coûts uniformes.

Ce premier chapitre nous a ainsi permis de nous familiariser avec les problèmes de flots et d'en étudier quelques extensions. Ces problèmes sont a priori indépendant de nos applications industrielles, mais ils concerne des problèmes de flot, notion au cœur de ces problèmes. En premier lieu, nous avons étudié des problèmes polynomiaux concernant le paradoxe de transport. Dans un second temps, nous avons montré comment un problème a priori complexe de repositionnement de véhicules pouvait se modéliser comme problème de changement d'outils.

Chapitre 2

Complexité de quelques problèmes combinatoires fondamentaux

Sommaire

2.1	Coût fixe et dégénérescence	29
2.1.1	Problèmes de transport avec coût fixe (FCTP)	30
2.1.2	Dégénérescence dans le FCTP	31
2.1.3	Plus grand degré de dégénérescence dans le FCTP	33
2.2	Complexité due aux contraintes spécifiques de gestion de véhicules	39
2.2.1	Rotation de véhicules : un problème de décomposition de graphe	40
2.2.2	De la complexité de l'équilibrage des charges entre véhicules	42

Dans ce chapitre on étudie la complexité de quelques problèmes combinatoires fondamentaux, sous problèmes très épurés de nos applications industrielles. On étudie la complexité du problème de transport avec coût fixe, et de ce même problème lorsqu'une partie de la combinatoire sous-jacente est connue. On reformule également les contraintes spécifiques du problème de transport de fret comme des problèmes de décomposition de graphes et on analyse leur complexité.

2.1 Coût fixe et dégénérescence

Nous étudions ici la complexité de problèmes de transport avec coût fixe. Les problèmes sont présentés dans la section 2.1.1. Dans la section 2.1.2, on se familiarise davantage avec la notion de dégénérescence. En particulier, on rappelle et prouve une caractérisation de l'existence de dégénérescence avec des sous-sommes à égalité. Ceci nous permet de démontrer que le problème d'existence d'une dégénérescence est \mathcal{NP} -complet. Le résultat nouveau est que le problème de trouver le plus grand degré de dégénérescence est \mathcal{NP} -difficile, même lorsque toutes les sous-sommes à égalité sont connues. Ce résultat est démontré dans la section 2.1.3. Cette preuve utilise notamment les problèmes de stable maximal et de set-packing. C'est une des contributions théoriques majeures de cette thèse. Cette section a

fait l'objet d'un article soumis fin 2009 dans le journal international *Mathematical and Computer Modelling*.

2.1.1 Problèmes de transport avec coût fixe (FCTP)

En général, dans un problème de transport, en plus d'un coût linéaire dépendant de la quantité transportée, il faut payer un coût fixe pour l'ouverture de chaque liaison. On a alors un problème de transport avec coût fixe *fixed charge transportation problem (FCTP)*. Pour obtenir la formulation de ce problème, il faut ajouter le coût fixe dans la modélisation du problème de transport linéaire. Comme pour le cas purement linéaire on note $a = \{a_1, a_2, \dots, a_m\}$ les offres et $b = \{b_1, b_2, \dots, b_n\}$ les demandes de transport. Les coûts de transport ont une part variable CV_{ij} et une partie fixe CF_{ij} . x_{ij} désigne la variable de flot et on introduit les variables binaires y_{ij} qui déterminent les liaisons ouvertes. La modélisation MIP est donnée dans la figure 2.1. L'objectif est de trouver le meilleur com-

$$\text{Minimiser : } \sum_{i=1}^m \sum_{j=1}^n CV_{ij}x_{ij} + \sum_{i=1}^m \sum_{j=1}^n CF_{ij} \cdot y_{ij}, \quad (2.1)$$

s.c.

$$\sum_{j=1}^n x_{ij} = a_i, \quad \forall i = 1..m, \quad (2.2)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad \forall j = 1..n, \quad (2.3)$$

$$x_{ij} \leq My_{ij}, \text{ i.e. } y_{ij} = \begin{cases} 0 & \text{si } x_{ij} = 0, \\ 1 & \text{si } x_{ij} > 0, \end{cases} \quad \forall i, j, \quad (2.4)$$

$$x_{ij} \geq 0, \quad \forall i, j, \quad (2.5)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j. \quad (2.6)$$

FIGURE 2.1 – Formulation MIP pour le FCTP.

promis entre coûts fixes d'ouverture de liaison et coûts variables de transport. La nouvelle contrainte (2.4) permet de garantir que les liaisons sur lesquelles du flot transite sont ouvertes. Pour écrire cette contrainte on utilise un nombre réel M suffisamment grand.

Le problème de transport avec coût fixe apparaît pour la première fois dans la littérature dans [Hirsch et Danzig, 1954]. [Balinski, 1961] proposent un algorithme dual de résolution et [Hirsch et Danzig, 1968] donnent une heuristique simple de résolution. Le problème de transport avec coût fixe est connu comme étant \mathcal{NP} -difficile. [Guisewite et Pardalos, 1990] étend ce résultat à tous les problèmes de flots de coût minimal avec une fonction coût concave. La plupart des méthodes de Recherche Opérationnelle ont été appliquées pour résoudre le FCTP. La littérature autour du FCTP est donc très riche [Sharma, 1977].

De plus, de nombreuses applications notamment en transport, distribution, ordonnancement et en localisation se modélisent via un FCTP. Pour plus de détails, et une revue de la littérature plus avancée, nous invitons le lecteur à consulter [Adlakha et Kowalski, 2003].

Définition. Le *problème de transport avec coût fixe constant* (*pure constant fixed charge transportation problem*) dit *pur FCTP* est le cas le plus simple d'un problème de transport avec coût fixe mais sans coût linéaire ; de plus le coût fixe est le même sur toutes les routes du réseau.

Un exemple pratique est le transport de matières très précieuses, pour lesquelles le coût de transport ne dépend pas de la quantité. La modélisation en MIP du problème de transport avec coût fixe constant est donnée figure 2.2. On note CF le coût fixe, identique sur toutes les arcs du réseau de transport.

$$\text{Minimiser : } \sum_{i=1}^m \sum_{j=1}^n CF \cdot y_{ij}, \quad (2.7)$$

s.t.

$$\sum_{j=1}^n x_{ij} = a_i, \quad \forall i = 1..m, \quad (2.8)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad \forall j = 1..n, \quad (2.9)$$

$$x_{ij} \leq My_{ij}, \quad \forall i, j, \quad (2.10)$$

$$x_{ij} \geq 0, \quad \forall i, j, \quad (2.11)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j. \quad (2.12)$$

FIGURE 2.2 – Formulation MIP du pur FCTP.

Dans ce cas, minimiser les coûts de transport revient à minimiser le nombre de liaisons ouvertes ce qui équivaut à avoir le nombre maximal d'arcs sur lesquels le flot de transport est nul. Il s'agit donc de trouver une solution de base de degré de dégénérescence maximal. La fonction coût étant concave, les solutions de base sont des arbres. Ainsi, résoudre le pur FCTP revient à trouver l'arbre de plus grand degré de dégénérescence. Commençons par traiter le cas d'une dégénérescence !

2.1.2 Dégénérescence dans le FCTP

Tout d'abord, dans la figure 2.3, nous redonnons l'exemple d'un problème de transport avec zéro, une ou deux dégénérescences.

Le but de cette section est de caractériser les dégénérescences, et d'étudier la complexité du problème d'existence d'une dégénérescence. Ces résultats sont déjà connus, mais n'ayant

pas trouvé de référence dans la littérature, nous les redémontrons ici.

Exemple. Offre $a = (4, 3, 5, 5)$

Demande $b = (4, 3, 6, 4)$

Coût fixe identique sur toutes les liaisons (coût fixe constant).

$b_j \rightarrow$	4	3	6	4	4	3	6	4	4	3	6	4
a_i	4	3		1	4	4		0	4	4		0
\downarrow	3			3	3			3	3	3		0
	5		5		5		5		5		5	
	5	1	3	1	5	3	1	1	5	1	4	4

FIGURE 2.3 – Problème de transport et solution avec zéro, une et deux dégénérescences.

Proposition 2.1.1. On considère le problème de transport avec coût fixe constant défini par $a = \{a_1, a_2, \dots, a_m\}$ et $b = \{b_1, b_2, \dots, b_n\}$.

L'existence d'une dégénérescence est équivalent à l'existence de deux sous-ensembles propres non-vides $S \subset \{a_1, a_2, \dots, a_m\}$ et $T \subset \{b_1, b_2, \dots, b_n\}$ tels que $\sum_{a_i \in S} a_i = \sum_{b_j \in T} b_j$. On dit

que l'on a une *sous-somme à égalité*.

Démonstration. – Supposons qu'il existe deux sous-ensembles propres non-vides $S \subset \{a_1, a_2, \dots, a_m\}$ et $T \subset \{b_1, b_2, \dots, b_n\}$ tels que $\sum_{a_i \in S} a_i = \sum_{b_j \in T} b_j$. Soit \bar{S} et \bar{T} les ensembles complémentaires respectivement de S et T . On a alors $\sum_{a_i \in \bar{S}} a_i = \sum_{b_j \in \bar{T}} b_j$.

On peut donc décomposer le problème de transport initial en deux sous-problèmes de transport. Une solution de base de chacun de ces problèmes est un arbre. On relie ces deux arbres par une dégénérescence (figure 2.4), et on obtient ainsi une solution dégénérée du problème de transport initial.

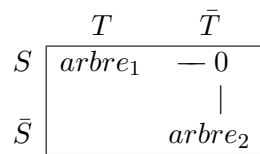


FIGURE 2.4 – Deux arbres reliés par une dégénérescence.

– Réciproquement, si il existe une dégénérescence, alors on peut décomposer le problème de transport initial en deux sous-problèmes de transport.

□

Proposition 2.1.2. Le problème de décision "Existe-t-il une dégénérescence?" dans le FCTP est \mathcal{NP} -complet.

Démonstration. On réduit le problème de bipartition connu comme étant \mathcal{NP} -complet vers le problème d'existence de sous-sommes à égalité qui est, d'après la proposition 2.1.1

équivalent au problème d'existence de dégénérescence. Les deux problèmes impliqués sont les suivants.

Problème DÉGÉNÉRESCENCE :

INSTANCE : Deux ensembles finis d'entiers positifs non nuls $\{a_1, a_2, \dots, a_m\}; \{b_1, b_2, \dots, b_n\}$ tels que $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$.

QUESTION : Existe-t-il deux sous-ensembles propres non-vides $S \subset \{a_1, a_2, \dots, a_m\}$ et $T \subset \{b_1, b_2, \dots, b_n\}$ tels que les sommes de leurs éléments sont les mêmes (i.e. $\sum_{a_i \in S} a_i = \sum_{b_j \in T} b_j$) ?

Problème PARTITION :

INSTANCE : Un ensemble fini d'entiers positifs non nuls $\{u_1, u_2, \dots, u_N\}$ dont la somme des éléments est paire $\sum_{i=1}^N u_i = 2D$.

QUESTION : Existe-t-il un sous-ensemble $S \subset \{u_1, u_2, \dots, u_N\}$ dont la somme des éléments est exactement égale à D ($\sum_{u_i \in S} u_i = D$) ?

Soit $\{u_1, u_2, \dots, u_N\}; u_i > 0; \sum_{i=1}^N u_i = 2D$; une instance du problème PARTITION. Nous construisons une instance de DÉGÉNÉRESCENCE de la façon suivante : $a_1 = u_1, \dots, a_N = u_N$ et $b_1 = D, b_2 = D$. Cette construction est polynomiale.

- Si DÉGÉNÉRESCENCE a une solution, alors il existe $S \subset \{a_1, a_2, \dots, a_N\}$ et $T \subset \{b_1, b_2\}$ tels que $\sum_{a_i \in S} a_i = \sum_{b_j \in T} b_j$. Comme T est un sous-ensemble propre non-vide, on a soit $T = \{b_1\}$ soit $T = \{b_2\}$. Et, dans les deux cas, $\sum_{b_j \in T} b_j = D$. On a donc $S \subset \{a_1, a_2, \dots, a_N\}$ tel que $\sum_{a_i \in S} a_i = D$. Donc, par définition S est une solution de PARTITION.
- Réciproquement, soit $S \subset \{u_1, u_2, \dots, u_N\}$ une solution de PARTITION. On a alors $\sum_{u_i \in S} u_i = D$, et les deux sous-ensembles S et $T = \{b_1\}$ forment une solution de DÉGÉNÉRESCENCE.

On a montré que PARTITION se réduisait polynomialement en DÉGÉNÉRESCENCE. Par conséquent le problème DÉGÉNÉRESCENCE est un problème \mathcal{NP} -complet. \square

On a démontré que le problème de transport à coût fixe est un problème \mathcal{NP} -difficile.

2.1.3 Plus grand degré de dégénérescence dans le FCTP

Définition. Le plus grand degré de dégénérescence plus un est égal à N si il existe une sous-partition de taille N $\{S_1, \dots, S_N\}$ de $\{a_1, a_2, \dots, a_m\}$ (S_k non vide) qui vérifie la propriété suivante : pour chaque S_k , il existe un ensemble $T_l \subset \{b_1, b_2, \dots, b_n\}$ tel que $\sum_{a_i \in S_k} a_i = \sum_{b_j \in T_l} b_j$.

On a vu que trouver une sous-somme à égalité (i.e. une dégénérescence) est déjà un problème \mathcal{NP} -difficile. Cette section a pour but d'identifier l'origine de cette complexité. **La difficulté est-elle seulement due à la difficulté de trouver les sous-sommes à égalité ?** Pour répondre à cette question, nous étudions la complexité du problème de trouver le plus grand degré de dégénérescence lorsque toutes les sous-sommes à égalité sont connues.

Définition. On considère deux sous-ensembles propres non vides $S \subset \{a_1, a_2, \dots, a_m\}$ et $T \subset \{b_1, b_2, \dots, b_n\}$. On dit que la sous-somme à égalité $\sum_{a_i \in S} a_i = \sum_{b_j \in T} b_j$ est une sous-somme *élémentaire* si il n'existe pas deux sous-ensembles S' et T' , sous-ensembles respectivement de S et de T , qui produisent une sous-somme à égalité. En d'autres termes, il n'existe pas de sous-somme à égalité plus petite sur les mêmes éléments.

Dans la suite de cette section, nous analysons la complexité du problème de trouver le plus grand degré de dégénérescence lorsque toutes les sous-sommes élémentaires à égalité sont connues.

Exemple. Le problème de transport support reste le même que dans l'exemple précédent. Les offres et demandes sont données par $a = \{4, 3, 5_3, 5_4\}$ ¹ et $b = \{4_1, 3, 6, 4_4\}$. Nous supposons que toutes les sous-sommes élémentaires à égalité sont connues, celles-ci sont données dans la figure 2.5.

	S_k		T_l	$V_i = S_k \cup T_l$
$S_1 = \{4\}$	4	T_1	4_1	$V_1 = \{4, 4_1\}$
		T_2	4_4	$V_2 = \{4, 4_4\}$
S_2	3	T_3	3	$V_3 = \{3, 3\}$
S_3	$3 + 5_3$	T_4	$4_1 + 4_4$	$V_4 = \{3, 5_3, 4_1, 4_4\}$
S_4	$3 + 5_4$			$V_5 = \{3, 5_4, 4_1, 4_4\}$
S_5	$4 + 5_3$	T_5	$3 + 6$	$V_6 = \{4, 5_3, 3, 6\}$
S_6	$4 + 5_4$			$V_7 = \{4, 5_4, 3, 6\}$
S_7	$5_3 + 5_4$	T_6	$6 + 4_1$	$V_8 = \{5_3, 5_4, 6, 4_1\}$
		T_7	$6 + 4_4$	$V_9 = \{5_3, 5_4, 6, 4_4\}$

FIGURE 2.5 – Liste de toutes les sous-sommes élémentaires à égalité.

Dans cet exemple, trouver l'arbre de plus grand degré de dégénérescence revient à trouver le plus grand nombre d'ensembles disjoints V_i qui forment une partition de $\{a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4\}$. C'est une instance du problème de Set-Packing, qui est en général un problème \mathcal{NP} -complet. Cependant, cette instance est très particulière et structurée, il faut donc étudier la complexité du problème de Set-Packing dans ce cas particulier.

Définition. On définit un graphe $G = (V, E)$. Chaque sommet correspond à un couple (S, T) définissant une sous-somme élémentaire à égalité. Une arête relie (S, T) et (S', T') si et seulement si $S \cap S' \neq \emptyset$ ou $T \cap T' \neq \emptyset$. On note $\{V_1, \dots, V_p\}$ la collection des ensembles

1. Afin de différencier les offres et les demandes qui ont la même valeur (même nombre entier), nous avons indexé ces nombres par leur position dans le tableau de transport.

V_i égal à l'union $S \cup T$ de la paire (S, T) correspondante. G est le graphe d'intersection de la collection $\{V_1, \dots, V_p\}$.

On peut aussi remarquer qu'une solution de Set-Packing est un stable maximal (*maximal stable set*) dans le graphe d'intersection de la famille des ensembles V_i . Dans la figure 2.6 on a représenté le graphe d'intersection de la famille des ensembles V_1, \dots, V_9 . Les trois ensembles $\{V_1, V_3, V_9\}$ définissent un stable maximal dans ce graphe, et sont aussi une solution de Set-Packing.

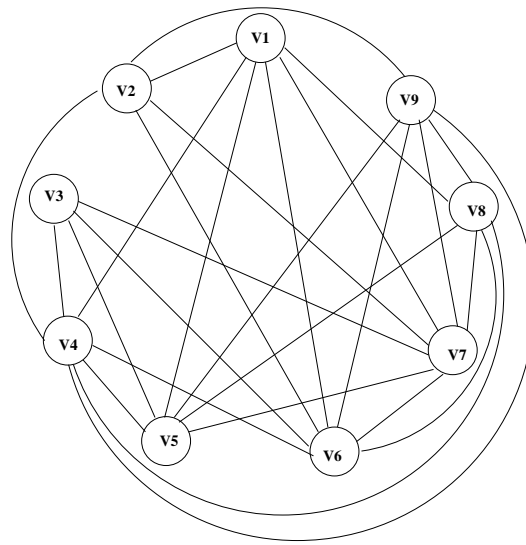


FIGURE 2.6 – Graphe lien dégénérescence et stable.

Nous allons maintenant établir de façon plus générale la relation entre les problèmes de degré maximal de dégénérescence, de set-packing et de stable max.

Proposition 2.1.3. On considère un problème de transport avec coût fixe constant défini par $a = \{a_1, a_2, \dots, a_m\}$ et $b = \{b_1, b_2, \dots, b_n\}$. On suppose que toutes les sous-sommes élémentaires à égalité sont connues, elles sont données via les ensembles V_1, \dots, V_p . Les trois problèmes suivants sont équivalents :

- Trouver un stable maximal dans le graphe G , graphe d'intersection de la collection $\{V_1, \dots, V_p\}$.
- Trouver une solution de base arbre du problème de transport avec coût fixe constant qui peut être décomposée en un maximum de sous-arbres disjoints. C'est à dire, trouver le nombre maximal d'ensembles V_i disjoints qui forment une partition de $\{a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n\}$.
- Trouver une solution de base arbre du problème de transport avec coût fixe constant de degré de dégénérescence maximal.

Démonstration. Chaque dégénérescence correspond à une partition en sous-arbres (proposition 2.1.1). Par conséquent, décomposer en le plus grand nombre de sous-arbres disjoints est équivalent à trouver le plus grand degré de dégénérescence. Il reste à montrer qu'un stable maximal dans le graphe G correspond à une décomposition de $\{a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n\}$ en le maximum de sous-arbres disjoints. Pour cela il suffit de montrer qu'un stable maximal dans le graphe G couvre tous les a_i et b_j .

Soit M un stable maximal, chaque a_i et b_j est couvert au plus une fois par M . Supposons qu'il reste des a_i et b_j non couverts. Alors les complémentaires \bar{A} et \bar{B} des ensembles $A = \{a_i \text{ couverts}\}$ et $B = \{b_j \text{ couverts}\}$ sont non vides et $\sum_A a_i = \sum_B b_j$ et $\sum_{\bar{A}} a_i = \sum_{\bar{B}} b_j$. Donc on a au moins un arbre de plus et M peut être augmenté par au moins un sommet, ce qui contredit sa maximalité. Par conséquent, M couvre nécessairement tous les a_i et b_j . On a ainsi montré que les éléments d'un stable maximal forment une partition de $\{a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n\}$. \square

Le problème de trouver un stable maximal est \mathcal{NP} -difficile en général. Ce problème reste-t-il difficile dans le graphe G qui a une structure très particulière ?

Théorème 2.1.1. On considère un problème de transport avec coût fixe constant défini par $a = \{a_1, a_2, \dots, a_m\}$ et $b = \{b_1, b_2, \dots, b_n\}$. On suppose que toutes les sous-sommes élémentaires à égalité sont données. Alors, trouver la solution optimale du problème de transport avec coût fixe constant (i.e. trouver l'arbre de degré de dégénérescence maximal) est un problème \mathcal{NP} -difficile.

Démonstration. La réduction est comme suit : on réduit d'abord MAX-STABLE à SET-PACKING. Puis on étend cette réduction et on réduit MAX-STABLE à SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ. Comme les problèmes SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ et DEGRÉ DE DÉGÉNÉRESCENCE sont équivalents, cette dernière réduction prouvera que le problème DEGRÉ DE DÉGÉNÉRESCENCE est \mathcal{NP} -difficile.

Les quatre problèmes impliqués dans cette preuve sont les suivants.

Problème DEGRÉ DE DÉGÉNÉRESCENCE :

INSTANCE : Deux ensembles $a = \{a_1, a_2, \dots, a_m\}$ et $b = \{b_1, b_2, \dots, b_n\}$ d'entiers positifs tels que $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ et un entier positif N .

QUESTION : Existe-t-il deux partitions, $\{S_1, \dots, S_N\}$ partition de $\{a_1, a_2, \dots, a_m\}$ et $\{T_1, \dots, T_N\}$ partition $\{b_1, b_2, \dots, b_n\}$, toutes deux de taille N , telles que pour tout $k = 1..N$, $\sum_{a_i \in S_k} a_i = \sum_{b_j \in T_k} b_j$?

Problème MAX-STABLE :

INSTANCE : $G = (V, E)$ un graphe et N un entier positif.

QUESTION : Existe-t-il un stable dans le graphe G de taille au moins N ?

Problème SET-PACKING :

INSTANCE : Une collection \mathcal{C} d'ensembles finis et un entier K , $K < |\mathcal{C}|$.

QUESTION : Existe-t-il K ensembles deux à deux disjoints dans la collection \mathcal{C} ?

Problème SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ :
 INSTANCE : $\mathcal{V} = \{V_1, \dots, V_p\}$ la collection des ensembles définissant toutes les sous-sommes élémentaires à égalité d'un problème de transport donné. Un entier positif K , $K < |\mathcal{V}|$.

QUESTION : La collection \mathcal{V} contient-elle K ensembles deux à deux disjoints ?

Reduction de MAX-STABLE à SET-PACKING.

Soit $G = (V, E)$ un graphe et N un entier, une instance de MAX-STABLE. Nous voulons construire une instance de SET-PACKING. Sans perte de généralité on peut supposer que G est connexe. On construit $|V| = n$ différents ensembles V_1, \dots, V_n de la façon suivante : on numérote les sommets de V de 1 à n . On considère toutes les arêtes de G dans un ordre quelconque. La l^{ieme} arête relie deux sommets, disons i et j , on ajoute alors l'entier 2^{l-1} à chacun des ensembles V_i et V_j .

Exemple. Soit G un graphe. Nous allons construire une collection d'ensembles dont le graphe d'intersection est G .

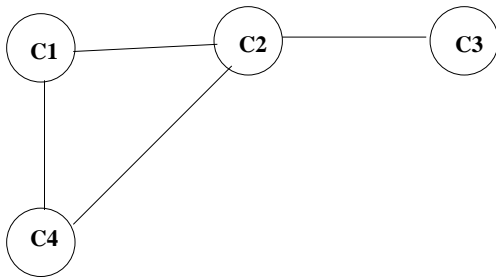


FIGURE 2.7 – Graphe d'intersection de la collection.

V_i	éléments
V_1	$2^0, 2^1$
V_2	$2^0, 2^2, 2^3$
V_3	2^2
V_4	$2^1, 2^3$

FIGURE 2.8 – Une collection possible.

La construction commence par balayer toutes les arêtes du graphe G , et construit autant d'ensembles qu'il y a de sommets dans G .

Une arête relie C_1 et C_2 , on ajoute l'élément 2^0 dans V_1 et V_2 .

Une arête relie C_1 et C_4 , on ajoute l'élément 2^1 dans V_1 et V_4 .

Une arête relie C_2 et C_3 , on ajoute l'élément 2^2 dans V_2 et V_3 .

Une arête relie C_2 et C_4 , on ajoute l'élément 2^3 dans V_2 et V_4 .

Le graphe G est identique au graphe d'intersection de la collection $\{V_1, \dots, V_k\}$. Un stable dans ce graphe correspond exactement à une collection de sous-ensembles disjoints. Donc, G contient un stable de taille N si et seulement si il existe N sous-ensembles disjoints dans la collection $\{V_1, \dots, V_k\}$.

Nous étendons maintenant cette réduction au problème de SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ.

Réduction de MAX-STABLE à SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ. Soit $G = (V, E)$ une instance de MAX-STABLE. Sans perte de généralité, on peut supposer que G ne contient pas de sommet isolé. Nous allons construire une instance de SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ.

Dans un premier temps, nous expliquons le mécanisme de construction sur un exemple. Ensuite nous donnerons la construction générale.

Exemple. Nous allons construire un problème de transport à partir du graphe G de la figure 2.7. La construction commence par balayer toutes les arêtes du graphe G .

Une arête relie C_1 et C_2 , on ajoute l'élément 2^0 dans les ensembles S_1 et S_2 .

Une arête relie C_1 et C_4 , on ajoute l'élément 2^1 dans les ensembles S_1 et S_4 .

Une arête relie C_2 et C_3 , on ajoute l'élément 2^2 dans les ensembles S_2 et S_3 .

Une arête relie C_2 et C_4 , on ajoute l'élément 2^3 dans les ensembles S_2 et S_4 .

On définit $a_1 = 2^0, a_2 = 2^1, a_3 = 2^2, a_4 = 2^3$. Pour $l = 1..4$ on définit chaque ensemble T_l comme le singleton composé du seul élément $b_l = \sum_{a_i \in S_l} a_i$. Ce qui donne $b_1 = 3, b_2 = 13, b_3 = 4, b_4 = 10$.

Ensuite, afin d'obtenir un problème de transport équilibré, il faut définir un élément supplémentaire $a_5 = 2^4 - 1 = 15$. Pour finir la construction, et décrire toutes les sous-sommes élémentaires à égalité, on ajoute tous les ensembles S_i contenant a_5 et les T_i correspondants à la collection. Comme par hypothèse toutes les sous-sommes à égalité sont connues, ces nouveaux sous-ensembles le sont également. Les ensembles V_i sont alors définis par $V_i = S_i \cup T_i$.

V_i	S_i	T_i		V_5	$2^0, a_5$	$3, 13$
V_1	$2^0, 2^1$	$2^0 + 2^1 = 3$		V_6	$2^1, a_5$	$4, 13$
V_2	$2^0, 2^2, 2^3$	$2^0 + 2^2 + 2^3 = 13$		V_7	$2^3, a_5$	$10, 13$
V_3	2^2	4				
V_4	$2^1, 2^3$	10				

Dans le cas général, le mécanisme de la construction est identique. On numérote les sommets du graphe G de 1 à k . On considère toutes les arêtes de G dans un ordre arbitraire. La l^{eme} arête relie deux sommets, disons i et j , on ajoute alors l'élément $a_l = 2^{l-1}$ dans les deux ensembles V_i et V_j .

On définit $a_1 = 2^0, a_2 = 2^1, \dots, a_k = 2^{k-1}$ et $a_{k+1} = 2^k - 1$; et $b_j = \sum_{S_j} a_i$ pour $j = 1 \dots k$. On ajoute alors tous les ensembles V_i contenant a_{k+1} ainsi que les S_i et T_i correspondants, par hypothèse ceux-ci sont connus.

Lemme 1. Pour $i = 1..k$, chaque a_i apparaît exactement deux fois dans les ensembles S_1, \dots, S_k , et $\{a_1, \dots, a_{k+1}\}, \{b_1, \dots, b_k\}$ définissent un problème de transport équilibré.

Démonstration. Ce lemme est une conséquence directe de la construction et du fait qu'au-

cun des ensembles S_1, \dots, S_k n'est isolé. □

Lemme 2. Les paires $(S_1, T_1), \dots, (S_k, T_k)$ sont les seules qui définissent des sous-sommes à égalité ne contenant pas a_{k+1} .

Démonstration. Par unicité de la décomposition en base 2, chaque b_j ($j = 1..k$) se décompose de façon unique avec les a_i , ($i = 1..k$). □

Nous allons démontrer que le problème de transport défini précédemment a un degré de dégénérescence égal à K si et seulement si le graphe G contient un stable de taille K .

- Supposons que le problème de transport a un degré de dégénérescence égal à K . Alors il existe $K + 1$ arbres disjoints, i.e $K + 1$ sous-sommes élémentaires à égalité indépendantes. Au plus une de ces sous-sommes contient a_{k+1} . Le cas échéant, on enlève cette sous-somme, il reste alors au moins K sous-sommes élémentaires à égalité indépendantes qui correspondent à un choix de K ensembles dans la collection $\{V_1, \dots, V_k\}$. Sinon, on a directement un choix de $K + 1$ ensembles dans la collection $\{V_1, \dots, V_k\}$. Dans les deux cas, ces K ou $K + 1$ ensembles de la collection $\{V_1, \dots, V_k\}$ correspondent à un stable de taille au moins K dans le graphe de départ.
- Réciproquement, supposons qu'il existe un stable de taille au moins K dans le graphe G . Alors les K ensembles V_i correspondants sont disjoints. Il reste à trouver un $K + 1^{eme}$ ensemble.

$$a_{k+1} + \sum_{a_i \text{ non couverts}, i=1..k} a_i = \sum_{b_j \text{ non couverts}, j=1..k} b_j$$

Si cette somme à égalité est élémentaire, elle nous donne un $K + 1^{eme}$ ensemble. Sinon, une de ses sous-sommes élémentaire nous donne ce $K + 1^{eme}$ ensemble.

On a donc bien une réduction de MAX-STABLE vers SET-PACKING CORRESPONDANT À UN PROBLÈME DE TRANSPORT ÉQUILIBRÉ. On a donc démontré, que même si l'on connaît toutes les sous-sommes élémentaires à égalité, le problème DEGRÉ DE DÉGÉNÉRESCENCE reste \mathcal{NP} -difficile. □

2.2 Complexité due aux contraintes spécifiques de gestion de véhicules

Dans cette section nous étudions la complexité due aux contraintes spécifiques de gestion de véhicules du problème de conception de réseau de service pour le transport de fret. Ces contraintes sont d'une part la rotation des véhicules sur un nombre limité de périodes, et d'autre part l'équilibrage des charges entre véhicules. Ces deux problèmes s'interprètent comme des problèmes combinatoires de décomposition de graphe. Dans la suite, nous établissons cette interprétation et analysons la complexité de ces problèmes.

2.2.1 Rotation de véhicules : un problème de décomposition de graphe

Une caractéristique de nombreux problèmes de transport est le "cyclage" des véhicules par rapport à un terminal d'origine. Ce cyclage est souhaité par les transporteurs pour des raisons de maintenance des véhicules. Ainsi, le transporteur veut que chaque véhicule repasse régulièrement par son terminal d'origine. Plus précisément, le temps entre deux passages d'un véhicule à son terminal d'origine doit être limité par un nombre de périodes, noté N . Dans le cas de planification de transport régulier, le transporteur fait sa planification sur une seule période. Cette planification est alors répétée sur une succession de périodes à l'identique, avec éventuellement des échanges de véhicules mais en aucun cas des changements de type de véhicule. Dans ces conditions, peut-on garantir que la taille des cycles finaux des véhicules est bornée par N ?

Plutôt que de regarder le graphe des morceaux de cycle dans le réseau espace-temps on regarde ces morceaux de cycles dans le réseau physique sous-jacent (cf figure 2.9(a) et figure 2.9(b))

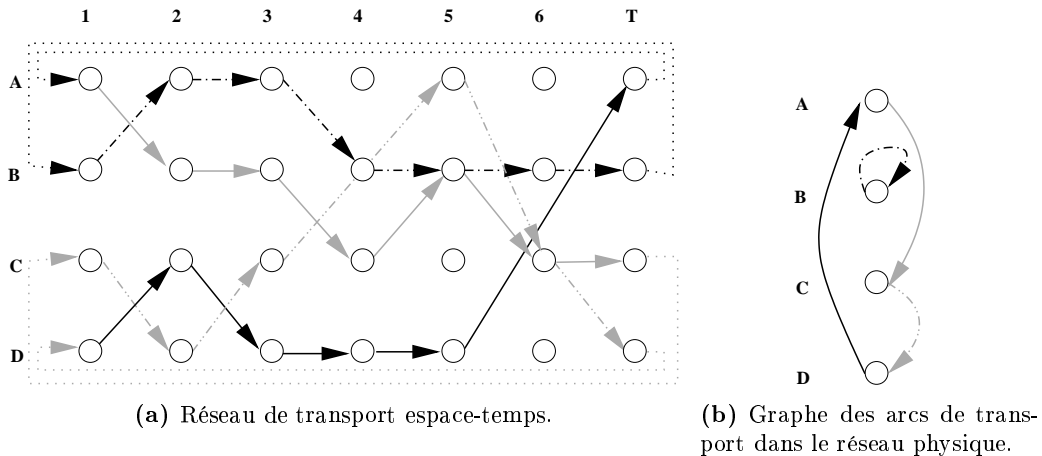


FIGURE 2.9 – Cycle dans le réseau espace-temps et dans le réseau physique.

On se place dans le graphe réduit (graphe du réseau physique, figure 2.9(b)). On suppose que chaque composante fortement connexe de ce graphe forme un graphe eulérien. Chacune de ces composantes comporte un parcours. Le graphe est donc bien décomposable en parcours (circuits). Dans ce graphe, un circuit d'au plus N arcs correspond au cyclage d'un véhicule sur un nombre de période inférieur à N . On veut que tous les véhicules cyclent en au plus N périodes. Cela revient à demander, une décomposition du graphe en circuits, de sorte que chaque circuit soit de longueur inférieure à N . On veut donc trouver une partition $A = A_1 \cup A_2 \cup \dots \cup A_p$ des arcs A du graphe tel que $\max_{i=1 \dots p} |A_i| \leq N$.

Pour s'en assurer, il suffit que

$$\min_{\substack{A=A_1 \cup A_2 \cup \dots \cup A_p \\ \text{partition en circuits}}} \max_{i=1 \dots p} |A_i| \leq N$$

Mais la détermination de ce nombre est-elle un problème facile ? Est-il possible de garantir cette inégalité avec des contraintes linéaires ?

De façon plus formelle, nous voulons connaître la complexité du problème suivant :

Problème DÉCOMPOSITION EN CIRCUITS :

INSTANCE : G graphe orienté eulérien, N un entier.

QUESTION : Existe-t-il une partition des arcs de G en circuits de longueur au plus N ?

Exemple : Dans le cas particulier où $N = 4$, on souhaite savoir si le graphe G est arc-décomposable en les graphes représentés figure 2.10.

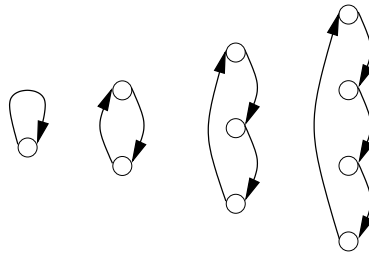


FIGURE 2.10 – Les cycles simples d'au plus 4 arcs

Dans la littérature, certains problèmes de décomposition de graphe ont déjà été étudiés dans le cas de graphes non orientés. Le problème très général de décomposition d'un graphe G en sous-graphes isomorphes à un graphe H donné est \mathcal{NP} -complet, dès lors que H a plus de 3 arêtes [Dor et Tarsi, 1997]. On note ce problème, H-DÉCOMPOSITION. Ce résultat a été conjecturé par Holyer dans [Holyer, 1981], qui l'a d'abord démontré dans le cas particulier de la décomposition en sous-graphes isomorphes à K_n , avec n entier $n \geq 3$.

Théorème 2.2.1 (d'Holyer). Etant donné un entier $n \geq 3$, le problème de décomposition (partition des arêtes) d'un graphe G en sous-graphes isomorphes à K_n est \mathcal{NP} -complet.

Corollaire. Le problème de décomposition (partition des arêtes) d'un graphe en cycles C_n de longueur n est un problème \mathcal{NP} -complet.

Le problème de décomposition en cycles reste \mathcal{NP} -complet si le graphe est eulérien. Il faut voir si ce problème reste \mathcal{NP} -complet dans le cas orienté.

Nous affirmons que la preuve du théorème d'Holyer [Holyer, 1981] peut être étendue au cas orienté. En quelques mots, la preuve du théorème d'Holyer [Holyer, 1981] est une réduction à partir de 3-SAT. Le premier graphe construit correspond à un pavage du plan par des

triangles, et on peut orienter le premier triangle dans le sens direct et orienter toutes les copies de ce triangle de la même façon. Dans un deuxième temps différentes copies de ces graphes sont recollées entre-elles. Ces recollages respectent l'orientation donnée. On peut donc étendre trivialement la preuve du théorème d'Holyer au cas orienté.

Ainsi le problème de partition des arcs d'un graphe en circuits de longueur donnée est \mathcal{NP} -complet. La réduction montre alors également que le problème DÉCOMPOSITION EN CIRCUITS est \mathcal{NP} -complet.

Dans le cas $n = 3$, la \mathcal{NP} -complétude du problème de décomposition des arêtes (arcs) d'un (di)graphe en cycles (circuit) de longueur 3 (triangles) est une conséquence du résultat plus ancien de la \mathcal{NP} -complétude du problème de *exact-cover with 3-sets* [Karp, 1972].

2.2.2 De la complexité de l'équilibrage des charges entre véhicules

Supposons que toutes les liaisons de transport à opérer sont déjà choisies. Elles sont les arcs d'un graphe. Il faut alors assigner des véhicules à ces liaisons, chaque véhicules opérant un certain nombre de liaisons, pour former un chemin dans le graphe. Equilibrer la charge entre les véhicules peut se reformuler sous la forme d'un problème de graphe consistant à partitionner l'ensemble des arcs en un ensemble de chemins arcs disjoints vérifiant des contraintes de longueur (minimum et maximum).

Plus formellement, cela revient à décomposer un graphe en chemins de longueurs bornées entre a et b . C'est le problème suivant.

Problème (a, b) -DÉCOMPOSITION :

INSTANCE : $G = (V, E)$ graphe orienté pondéré (chaque $e \in E$ a un poids c_e), $a, b \in \mathbb{N}$.

QUESTION : Existe-t-il une décomposition des arêtes de G en chemins C_i tels que $\forall i, a \leq \sum_{e \in C_i} c_e \leq b$?

Dans le cas général, la complexité de ce problème reste encore un problème ouvert, aussi bien dans le cas orienté que dans le cas non orienté. Certains cas particuliers ont été démontrés polynomiaux ou \mathcal{NP} -complet dans [Teypez et Rapine, 2008]. Ceux-ci sont précisés dans la proposition 2.2.1. La preuve et plus de références sur le sujet sont données dans le rapport de recherche [Teypez et Rapine, 2008] et la thèse [Teypez, 2008].

Proposition 2.2.1. Le problème (a, b) -Décomposition est un problème polynomial (de la classe \mathcal{P}) dans les cas suivants [Teypez et Rapine, 2008] :

- $a \leq 2$ et $b \geq a$;
- $a \geq 3$ et le graphe est un arbre ou est eulérien.

Ce problème est \mathcal{NP} -complet dans les cas suivants :

- $a = b = 3$ même dans le cas d'un graphe biparti ;
- $a \geq 3$ et $b > a$ (conjecture).

Le dernier point de ce théorème n'est pas encore démontré. La complexité du problème (a, b) -Décomposition lorsque $a \geq 3$ et $b > a$, est encore un problème ouvert. [Teypez et Rapine, 2008] ont conjecturé que ce problème était \mathcal{NP} -complet.

Synthèse

Dans cette première partie, nous avons étudié quelques problèmes combinatoires sous-jacents aux deux problèmes industriels.

Nous avons d'abord analysé des modèles de flot dans les réseaux de distribution. Ces modèles permettent notamment de caractériser le paradoxe de transport, et de modéliser un problème pratique de repositionnement de véhicules.

Dans un deuxième temps, nous nous sommes intéressés à la complexité de problèmes combinatoires. Le premier problème est autour du coût fixe. Même dans le cas d'un coût fixe identique sur tous les arcs d'un réseau, le problème de transport est déjà un problème difficile. L'une des contributions théoriques majeures de cette thèse, a été de montrer que ce problème reste difficile, même lorsque une partie de la combinatoire sous-jacente est connue.

Les deux autres problèmes soulevés sont des problèmes de partition des arcs d'un graphe. Là encore, nous avons démontré leur \mathcal{NP} -complétude dans la plupart des cas.

Deuxième partie

Conception de réseau de service pour le transport de fret

Dans cette partie nous présentons le travail de recherche réalisé sur un problème de conception d'un réseau de service avec gestion de flotte. Ce problème modélise un problème industriel de transport de fret.

Les caractéristiques de ce problème sont détaillées dans le chapitre 3. Dans le chapitre 4, nous présentons différentes formulations MIP du problème et les comparons analytiquement et expérimentalement sur de petites instances. Finalement, dans le chapitre 5, nous ébauchons un algorithme à base de génération de colonnes capable de résoudre le problème pour de plus grandes instances. Nous évaluons la qualité de cette algorithme selon différentes approximations. Pour faciliter la lecture, l'ensemble des notations et paramètres du problèmes sont récapitulés dans l'annexe A.

Le chapitre 4 de cette troisième partie, qui présente une étude comparative de différentes formulations a fait l'objet d'une publication en tant que chapitre d'ouvrage en collaboration avec Teodor Gabriel Crainic [Schrenk *et al.*, 2010].

Chapitre 3

Conception de réseau de service pour le transport de fret

Sommaire

3.1	Description du problème	48
3.1.1	Données	48
3.1.2	Contraintes	51
3.1.3	Fonction d'optimisation	52
3.1.4	Solution	52
3.2	Complexité du problème	54
3.2.1	Un problème \mathcal{NP} ?	55
3.2.2	Un problème au moins \mathcal{NP} -difficile	55
3.2.3	Complexité théorique et pratique due aux contraintes spécifiques	58
3.3	Etat de l'art	59
3.3.1	Systèmes de transport de marchandises	59
3.3.2	Flot multicommodités	62
3.3.3	Tournées de véhicules	63
3.3.4	Conception de réseau	64
3.3.5	Conception de réseau de service	67
3.3.6	Positionnement de notre problème de transport de fret	69

Ce problème est issu d'une collaboration entre le laboratoire G-SCOP et un industriel du domaine du transport de fret. Pour des raisons de confidentialité, nous ne mentionnerons ni le nom de cet industriel, ni le secteur de transport concerné. Une première réponse a été apportée à l'industriel dans [Teypez, 2008]. Il s'agissait alors de déterminer une meilleure planification d'opérations régulières de transport, sans limitation a priori des ressources (nombre de véhicules), en fonction d'un marché régulier constitué d'un ensemble de demandes de transport répétitives. Ces demandes pouvant, selon leur rentabilité, être soit rejetées, soit partiellement ou totalement satisfaites. La réponse apportée fournit non seulement la flotte de véhicule idéale pour répondre au marché, mais aussi le plan de transport

pour offrir un service régulier. Ce plan de transport définit à la fois les routes empruntées par les véhicules et leurs horaires et les opérations de chargement et de déchargement du fret transporté. Ce problème est modélisé via un modèle linéaire en nombres entiers, et résolu par une heuristique. Cette modélisation ainsi que l'heuristique de résolution et les résultats sont publiés dans [Teypez *et al.*, 2010]. Ma part de travail dans cet article se situe principalement au niveau des tests de performance de l'heuristique proposée et implémentée par Nicolas Teypez.

Le but de notre étude se trouve dans la continuité de ce problème : on suppose désormais que le transporteur a une flotte prédéfinie. Il faut alors faire au mieux avec cette flotte. Notre problème se situe à l'interface entre la planification tactique et la planification opérationnelle. C'est un problème de conception de réseaux de service avec gestion de flotte, *Service Network Design with Asset Management*. Un *service* est défini comme étant un trajet spécifique (d'un terminal de départ vers un terminal d'arrivée) à des horaires spécifiques. Les décisions à prendre pour concevoir le réseau de service concernent à la fois les véhicules et les marchandises. Elles ont pour but de répondre aux questions suivantes :

- Quels sont les itinéraires et les horaires des services ?
- Quels véhicules opèrent ces services ?
- Quelles demandes de transport accepter totalement, partiellement ou refuser ?
- Comment les marchandises transitent-elles dans ce réseau de services ?

Le but est d'obtenir un plan de transport qui maximise le profit du transporteur (revenu moins coûts de transport), tout en s'assurant qu'un certain nombre de contraintes spécifiques à l'application industrielle soient vérifiées (quotas d'utilisation des véhicules, fenêtres de temps, cyclage des véhicules, ...).

3.1 Description du problème

Cette partie a pour objet la description des principales caractéristiques du problème. Le but est d'apporter une aide à la décision pour un transporteur afin de déterminer un ensemble de plannings des véhicules de sa flotte en vue de répondre le mieux possible à des demandes de transport.

Un exemple illustratif complet est donné au fil de cette description. Un exemple de solution est proposé dans la section 3.1.4.

3.1.1 Données

- *Réseau* : Le problème de transport est défini sur un réseau physique (figure 3.1) composé de terminaux (ports, gares ferroviaires, aéroports ou entrepôts) que l'on modélise par un graphe complet orienté $G' = (N', A')$. Les connexions possibles (voies maritimes, chemins de fer, voies aériennes ou routes) sont les arcs de ce réseau, un arc $(i', j') \in A'$ représentant un service potentiel entre les terminaux i' et j' . Une fonction de distance euclidienne $d' : A' \rightarrow \mathbb{R}_+$ (qui respecte l'inégalité triangulaire) est donnée sur ce réseau, $d'(i', j')$ représentant la distance entre les deux terminaux

i' et j' . Pour plus de lisibilité on peut représenter les distances dans une matrices des distances (figure 3.2).

Exemple. Un réseau physique de 4 terminaux.

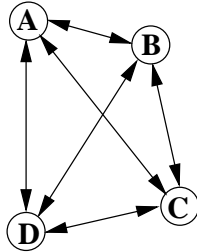


FIGURE 3.1 – Réseau physique de transport.

	A	B	C	D
A	0	200	300	450
B	200	0	150	300
C	300	150	0	200
D	450	300	200	0

FIGURE 3.2 – Matrice des distances.

La dimension temporelle est définie par un *horizon de planification* aussi appelé *période*. Prendre en compte la gestion des véhicules et la construction de leurs tournées nécessitent d'introduire le temps dans ce réseau en définissant un réseau espace-temps. Pour ce faire, la période de planification est discrétisée en intervalles $\{1, \dots, T\}$. Le réseau espace-temps est modélisé par le graphe $G = (N, A)$, dans lequel les nœuds correspondent à des copies des nœuds du réseau physique, une copie par intervalle de temps. A un nœud i du réseau espace-temps correspond donc un terminal N_i et une période T_i . Les nœuds sont reliés par deux types d'arcs : d'une part les services potentiels entre différents terminaux avec implicitement leurs horaires, d'autre part on ajoute des arcs d'attente sur place ou arcs de stationnement (*holding arcs*). Un certain nombre de contraintes temporelles sont directement intégrées dans ce réseau (contraintes de couvre-feu, temps de trajet entre deux terminaux, ...), ce qui limite le nombre d'arcs du réseau. Un exemple de réseau espace-temps, construit en ajoutant une dimension temporelle au réseau physique de la figure 3.1, est représenté dans la figure 3.3, les arcs horizontaux correspondant à des arcs d'attente.

Exemple. Réseaux espace-temps avec services potentiels.

- *Demandes de transport* : Le transporteur peut satisfaire globalement ou partiellement des demandes de transport (marchandises) origine-destination. Les demandes sont périodiques, i.e. répétées à l'identique de période en période (planification d'un service régulier pour le transport de marchandises). La demande de transport périodique k est définie par son terminal d'origine o^k , son terminal de destination d^k , sa quantité (tonnage) w^k et le revenu engendré P^k par le transport d'une unité de cette demande. On note K l'ensemble des demandes.

Exemple. Demandes de transport.

- *Flotte de véhicules* : Pour transporter le fret, le transporteur dispose d'une flotte prédéfinie, limitée et hétérogène V de véhicules. Ces véhicules sont classés selon leur

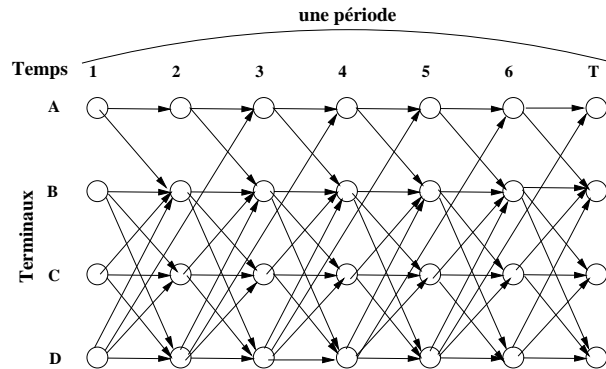


FIGURE 3.3 – Réseau espace-temps.

demande k	origine o^k	destination d^k	quantité w^k	revenu P^k
1	A	B	80	40
2	B	C	40	30
3	C	D	130	40
4	B	D	120	10
5	D	C	150	20
6	D	A	70	20
7	D	B	60	10
8	B	A	60	60

TABLE 3.1 – Demandes de transport.

type. On désigne par V_{type} l'ensemble des véhicules de type $type$, ainsi $V = \cup_{type} V_{type}$. Nous supposons que les véhicules sont répartis sur le réseau comme souhaité au début de l'horizon de planification. La capacité de chaque véhicule $v \in V$ dépend de la distance à parcourir, elle est donnée par la fonction $u_v : A' \rightarrow \mathbb{R}_+$. Pour plus de lisibilité, on note u_{ijv} la capacité du véhicule v entre N_i et N_j . On connaît la vitesse moyenne des véhicules, identique pour tous les véhicules du même type $vitesse(type)$ et on note $t_v(i', j')$ le temps de trajet du véhicule v pour aller du terminal i' au terminal j' , pour faciliter la lecture on note t_{ijv} le temps de trajet entre les terminaux N_i et N_j . On note τ le temps de réengagement, commun à tous les véhicules. Ce temps (environ 1 heure) est nécessaire pour réaliser les opérations de chargement et de déchargement du fret ainsi que le réapprovisionnement éventuel en carburant.

Exemple. Dans notre exemple nous considérons une flotte homogène (un seul type) de deux véhicules.

VM#1 VehiculeMoyen 100t 100km/h

VM#2 VehiculeMoyen 100t 100km/h

Ces véhicules sont de type VehiculeMoyen, leur capacité est de 100 unités, et leur vitesse moyenne de 100km/h

3.1.2 Contraintes

- *Satisfaction des demandes* : Si le transporteur accepte partiellement ou totalement une demande, il s'engage à transporter la même quantité de cette demande à chaque période. De plus, chaque livraison doit avoir lieu dans la même période que la collecte correspondante. Afin d'améliorer ses bénéfices, il a la possibilité de consolider les flux de marchandises via des opérations de transbordement, à condition de ne pas dépasser la capacité des véhicules. La fraction de demande satisfaite peut aussi être divisée pour être transportée par différents véhicules.
- *Planification périodique* : Le plan de transport calculé sur une période doit pouvoir être répété de période en période sur un ensemble de périodes aussi appelé *saison*. Pour se faire, il faut qu'au début de chaque période les véhicules soient répartis de façon analogue ; c'est-à-dire qu'on retrouve le même nombre de véhicules du même type dans chaque terminal au début et à la fin de la période de planification (= début période suivante). Les véhicules peuvent ensuite se remplacer sur leurs routes. Cette rotation des véhicules est illustrée figure 3.4. On a représenté les routes de deux véhicules sur une période. Sur la période suivante ces deux véhicules s'échangeront leurs routes et ainsi de suite. Dans le cadre de notre application, le temps avant lequel un véhicule retrouve sa route initiale est borné par un nombre N de périodes. En pratique les véhicules cyclent rarement sur plus de 3 à 4 périodes.

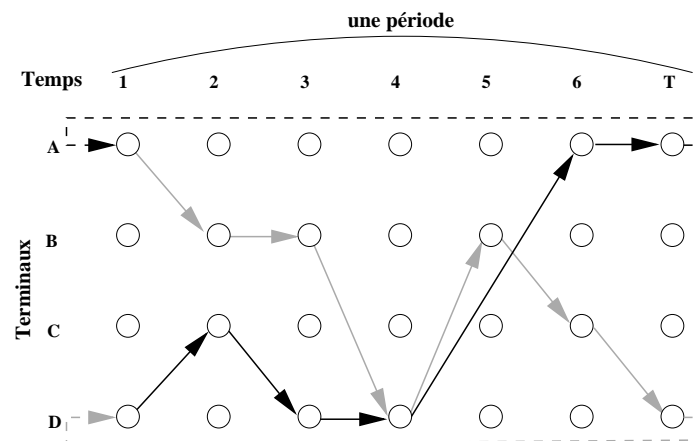


FIGURE 3.4 – Réseau espace-temps et rotation de deux véhicules.

- *Quota d'utilisation des véhicules* : La deuxième contrainte spécifique sur les véhicules est que ceux-ci doivent être utilisés de façon équitable : sur une saison aucun des véhicules utilisés ne doit être ni sous-exploité ni sur-exploité. Dans notre cas, cette contrainte se traduit par le fait que si un véhicule est utilisé, alors son temps d'utilisation global sur une saison doit être compris entre deux bornes. Autrement dit les véhicules doivent respecter des quotas d'utilisation minimum et maximum

d'heures d'utilisation saisonnière. Ces quotas ont été ramenés à des quotas par période de planification. Ainsi, la longueur, en temps de parcours, du chemin opéré par le véhicule v pendant une période est bornée inférieurement par $Quota_{v,min}$ et supérieurement par $Quota_{v,max}$. Ceci est une condition suffisante pour que, au final, les quotas d'utilisation annuels soient respectés, cette condition est trop forte par rapport aux contraintes industrielles, mais est quand même satisfaisante du point de vue de la qualité des solutions obtenues.

3.1.3 Fonction d'optimisation

La fonction d'optimisation intègre des paramètres liés aux coûts et gains induits par le plan de transport. Le but est de maximiser le profit du transporteur, défini comme la différence entre d'une part ses revenus, et d'autre part les coûts fixes et variables de transport.

Les coûts de transport sont de deux types : un coût fixe pour l'ouverture d'un service de transport (coût du personnel, coût de manutention et coût en carburant) et un coût variable linéaire selon le tonnage de marchandises à bord. Une recette est engrangée par le transporteur pour chaque unité de fret transportée jusqu'à sa destination. La valeur de cette recette dépend de la marchandise. Ainsi, le transporteur doit faire un arbitrage entre coûts de transport et recettes pour maximiser son profit.

- *Coûts fixes* : Le coût fixe d'ouverture d'un service entre les terminaux i' et j' avec le véhicule v est $CF_v(i', j')$.
- *Coûts variables* : On suppose que le coût variable de transport pour un véhicule v sur l'arc $(i'j')$ est linéaire par rapport à la quantité transportée w . On note $CV_v(i', j')$ le coût variable unitaire de transport entre i' et j' avec le véhicule v .

Etant donné que ces coûts sont les mêmes pour tous les véhicules de même type et indépendants des heures de départ, on étend ces fonctions coût au réseau espace-temps, le coût entre deux nœuds i et j étant égal à celui entre les nœuds physiques correspondant. On note alors, pour plus de lisibilité CF_{ijv} et CV_{ijv} le coût fixe et le coût variable unitaire entre les terminaux N_i et N_j avec le véhicule v .

- *Revenus* : La satisfaction d'une unité de demande k engendre un revenu P^k .

Le but de cette étude est d'apporter une aide à la décision pour le transporteur. L'idée est de lui proposer une planification sur une période, planification qu'il pourra ensuite répéter sur toutes les périodes d'une saison. Sur une période, il faut à la fois déterminer un ensemble de plannings (itinéraires et horaires) des véhicules de sa flotte, mais aussi décrire les demandes acceptées totalement ou partiellement et leur trajet à bord des véhicules. Cette planification devra garantir un maximum de profit pour le transporteur.

3.1.4 Solution

Une solution du problème est composée d'un ensemble de plannings pour les véhicules de la flotte du transporteur et des itinéraires décrivant le transport des marchandises choisies. La figure 3.5 représente les plannings des deux véhicules de la flotte, on peut lire

3.1 Description du problème

les routes et horaires des véhicules, qui sont écrit formellement à gauche de la figure 3.5. Les trajets et horaires de transport des marchandises sont représentés dans la figure 3.6. On retrouve les deux routes, et sur chacune des flèches correspondant à un service on a détaillé les marchandises à bord (quantité et numéro de la demande en exposant). À gauche de la figure 3.6, on retrouve le détail des itinéraires des marchandises avec notamment les demandes de transport non satisfaites, partiellement satisfaites, ou satisfaites en plusieurs morceaux. Les deux dernières lignes concernent la marchandise n°8. Elles signifient que 100 unités de la demande 8 sont transportées en empruntant le service 2 puis le service 8, et que 10 unités de la demande 8 sont réjettées.

Exemple. Solution : services opérés par les véhicules.

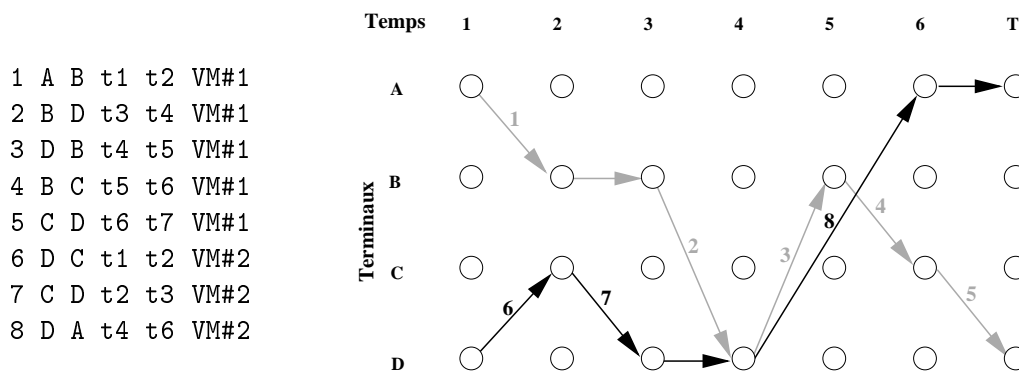


FIGURE 3.5 – Plannings solutions pour les deux véhicules.

Cette solution respecte l'ensemble des contraintes du problème, elle satisfiera le transporteur si elle engendre le profit le plus élevé possible, c'est-à-dire si elle réalise le meilleur compromis entre le profit engendré par la satisfaction de demandes et les coûts fixes et variables de transport des demandes satisfaites. Selon les calculs du tableau 3.2, ce profit s'élève à +11 680.

Coûts fixes	
coûts fixe d'ouverture des 8 services : $8 * 300$	-2 400
Coûts variables quantité*distance*prix	
$80 * 200 * 0.020 + 100 * 300 * 0.020 + \dots + 100 * 450 * 0.020$	-5 120
Revenu quantité*revenu unitaire	
satisfaction des demandes $80 * 30 + \dots + 100 * 60$	+19 200
Profit	+11 680

TABLE 3.2 – Profit de la solution pour le transporteur.

Exemple. Solution : itinéraires des marchandises.

- 1 80 1
- 2 40 4
- 3 90 5
- 3 40 7
- 3 10 4 5
- 4 110 rejeté
- 5 50 3 4
- 5 100 6
- 6 70 rejeté
- 7 50 3
- 7 10 rejeté
- 8 100 2 8
- 8 10 rejeté

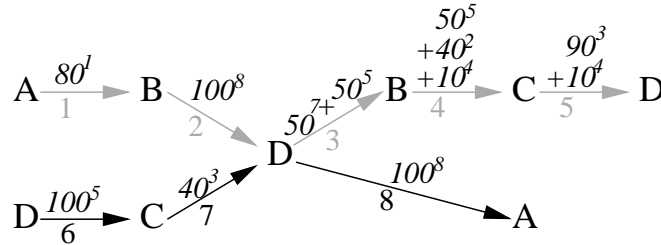


FIGURE 3.6 – Marchandises transportées via les deux routes.

3.2 Complexité du problème

Le but de cette section est d'analyser la complexité du problème de conception de réseau de service avec gestion de flotte pour le transport de fret. Nous démontrons que notre problème de transport de fret est très difficile. Cette section étant technique, nous conseillons au lecteur qui n'est pas familier avec la notion de \mathcal{NP} -complétude de passer directement à la lecture de la section suivante. Le problème de décision dont nous analysons la complexité est écrit ci-après de façon condensée avec un rappel des notations essentielles.

Problème CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE :

INSTANCE :

- Un réseau physique comportant n terminaux, représenté par un graphe complet $G' = (N', A')$. Une fonction de distance euclidienne $d' : A' \rightarrow \mathbb{R}_+$.
- La durée $DureeP$ de l'horizon de planification i.e de la période, et un nombre total de périodes $NombreP$ (égal au nombre de périodes constituant une saison).
- Un ensemble de demandes origine-destination K , répétées sur chaque période, et pour chaque demande $k \in K$: son origine $o^k \in N'$, sa destination $d^k \in N'$, la quantité w^k à transporter et la recette P^k engendrée par le transport d'une unité de marchandise.
- Une flotte de véhicules V de différents types V_{type} et pour tout type $type$, un nombre $|V_{type}|$ de véhicules de ce type ; la capacité de ces véhicules, $u_v : A' \rightarrow \mathbb{R}_+$; leur vitesse moyenne $vitesse(type)$ et un temps maximal de cyclage de N périodes. Le coût de transport (comportant la partie fixe et partie variable) $Cout_v(i, j, w)$ qui représente le coût de transport de w unités de marchandises du terminal i au terminal j à bord du véhicule v .
- Un nombre réel P .

3.2 Complexité du problème

BUT : Trouver un plan de transport (véhicules + marchandises), qui maximise le profit du transporteur, i.e. le revenu engendré par les demandes satisfaites moins les coûts de transport.

QUESTION : Existe-t-il un plan de transport de profit au moins égal à P ?

3.2.1 Un problème \mathcal{NP} ?

Avant tout, pour étudier la complexité du problème de CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE, il faut déterminer si ce problème appartient à la classe \mathcal{NP} , classe des problèmes non déterministes polynomiaux.

Savoir si le problème est ou non dans \mathcal{NP} , n'est pas si simple. Naïvement, le raisonnement est le suivant. Une solution du problème de conception de réseau de service contient un ensemble de plannings pour les véhicules. Pour s'assurer que ces plannings vérifient bien les contraintes du problème, il est nécessaire de les parcourir. Il faut donc parcourir tous les services. Cette vérification, peut se faire en temps polynomial par rapport à la taille de l'instance, uniquement si le nombre total de services est polynomialement borné. Un exemple de borne sur le nombre de service opérés par un véhicule v est $DureeP * vitesse(v) / \min\{d'(i', j') | (i', j') \in A'\}$. La conclusion serait alors que, dans le cas général, le problème étudié n'est pas dans la classe \mathcal{NP} .

Cependant, il existe peut être une représentation compacte de la solution, qui permette de vérifier en temps polynomial la faisabilité d'une solution. Un exemple de problème d'ordonnancement ayant une telle représentation compacte est donné dans [Brauner *et al.*, 2007].

On ne sait donc pas si le problème étudié est ou non \mathcal{NP} .

3.2.2 Un problème au moins \mathcal{NP} -difficile

Dans la suite nous allons démontrer que ce problème est plus difficile que tout problème de la classe \mathcal{NP} . Pour ce faire nous allons faire une réduction du problème de PLUS COURT CHEMIN HAMILTONIEN ENTRE s ET t vers notre problème.

Problème PLUS COURT CHEMIN HAMILTONIEN ENTRE s ET t :

INSTANCE : $G = (V, E)$ graphe, $s, t \in V$ deux sommets, $d : E \rightarrow \mathbb{R}_+$ une fonction distance, un nombre réel L .

BUT : Trouver le plus court chemin hamiltonien de s à t .

QUESTION : Existe-t-il un chemin hamiltonien de s à t de longueur inférieure ou égale à L ?

Ce problème est équivalent au problème du voyageur de commerce, et est, de ce fait \mathcal{NP} -difficile. Ce problème reste \mathcal{NP} -difficile lorsque le graphe G est complet, et lorsque la fonction d vérifie l'inégalité triangulaire.

Théorème 3.2.1. Le problème de décision correspondant à CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE est au moins \mathcal{NP} -complet.

Démonstration. Réduction à partir du problème PLUS COURT CHEMIN HAMILTONIEN

ENTRE s ET t

Soit $G = (V, E)$ un graphe complet, $V = \{v_1, v_2, \dots, v_n\}$, et $d : E \rightarrow \mathbb{R}_+$ une fonction distance vérifiant l'inégalité triangulaire une instance du problème PLUS COURT CHEMIN HAMILTONIEN ENTRE s ET t .

Nous construisons une instance du problème CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE de la façon suivante.

- Le graphe $G' = (N', A')$ avec $N' = V \cup \{v_0\}$, et $A' = E \cup \{(v_0, v_i) | v_i \in V\}$
- la fonction distance $d' : A' \rightarrow \mathbb{R}_+$, identique à d sur E et $d'(v_0, v_i) = \sum_{e \in E} d(e) = D$ pour $i = 1, \dots, n$.
- La durée d'une période $DureeP = 1$, et on considère $NombreP = 2$ périodes.
- n demandes $K = \{k_1, \dots, k_n\}$ répétées à chacune des deux périodes et pour tout $j = 1, \dots, n$,
 - $o^{k_j} = v_0$,
 - $d^{k_j} = v_j$,
 - $w^{k_j} = 1$,
 - $P^{k_j} = M$ très grand ($M > n * \sum_{e' \in E'} Cout_1(e')$).
- Un seul type de véhicule $type = 1$, et un seul véhicule de ce type $|V_1| = 1$.
 - la capacité du véhicule $u_1 : E \rightarrow \mathbb{R}_+$ fonction constante égale à n ,
 - le coût de transport de q unités entre i et $j : Cout_1(i, j, q) = d((i, j))$ pour tout $(i, j) \in E$,
 - la vitesse du véhicule $vitesse(1) = 3 * D$,
 - un temps de cyclage de $N = 1$ période.
- Un nombre réel $P = nM - (L + 2D)$.

L'idée générale de cette instance est d'avoir un véhicule assez rapide pour être capable de partir de v_0 , de visiter au moins une fois tous les sommets et de revenir à v_0 , sans pour autant être retourné à v_0 entre-temps, et tout cela en moins d'une unité de temps (=durée d'une période). On note que dans le cadre de cette preuve le temps n'est pas discrétisé, cette discrétisation ne faisant pas partie intégrante du problème dont on analyse la complexité. Le temps de cyclage $N = 1$ période assure que le véhicule retourne à v_0 à la fin de sa tournée. La preuve formelle suit.

Les figures 3.7 et 3.8 permettent de visualiser les graphes des différentes instances.

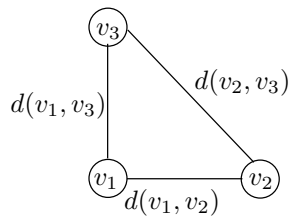


FIGURE 3.7 – Graphe G , instance de plus court chemin hamiltonien.

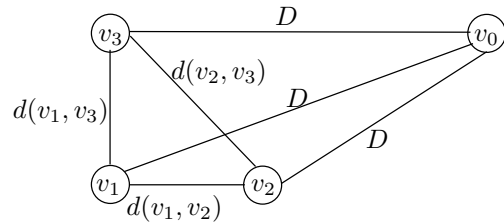


FIGURE 3.8 – Graphe G' , instance de conception de réseau de transport avec gestion de flotte.

3.2 Complexité du problème

On remarque que toutes les demandes peuvent être satisfaites dans la période par un véhicule qui part du dépôt et qui retourne au dépôt (longueur plan de transport $\leq 3 * D$, et vitesse véhicule = $3 * D$). On remarque également, que dans toute solution admissible, le véhicule passe au plus une fois par v_0 .

Le prix de toute commande étant supérieur à la longueur de n'importe quel cycle de G' passant par v_0 , toutes les demandes sont satisfaites dans une solution optimale qui maximise le profit. Donc toute solution optimale passe par tous les sommets. Le profit engendré par une telle solution est $nM - (d(\Pi) + 2D)$ où Π est le plan de transport restreint à $G = G' - \{v_0\}$.

Montrons qu'il existe une solution de PLUS COURT CHEMIN HAMILTONIEN de longueur inférieure ou égale à L si et seulement si il existe une solution de CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE de profit supérieur ou égal à $P = nM - (L + 2D)$.

Supposons qu'il existe une solution de PLUS COURT CHEMIN HAMILTONIEN Π de longueur inférieure ou égale à L . Alors le cycle $v_0\Pi v_0$ est de longueur inférieure ou égale à $L + 2D$, son coût est donc également inférieur ou égal à $L + 2D$. Par conséquent, il définit un planning qui engendre un profit supérieur ou égal à $P = nM - (L + 2D)$.

Réciproquement, supposons qu'il existe une solution (un cycle) Γ de CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE, de profit supérieur à $C = nM - (L + 2D)$. Disons que Γ passe exactement une fois par v_0 (obligé) et par un nombre t d'autres sommets. On a donc $tM - d(\Gamma) \geq nM - (L + 2D)$, donc $d(\Gamma) \leq L + 2D$. On peut supposer que Γ passe par v_1, v_2, \dots, v_t et qu'il existe un chemin Π tel que $\Gamma = (v_0\Pi v_t v_0)$.

Le cycle $\Gamma' = (v_0\Pi v_t v_{t+1} \dots v_{n-1} v_n v_0)$ a un profit supérieur à Γ . Donc $nM - d(\Gamma') \geq tM - d(\Gamma) \geq nM - (L + 2D)$, d'où $d(\Gamma') \leq d(\Gamma) \leq L + 2D$.

Il reste à transformer Γ' en un cycle hamiltonien sans augmenter sa longueur. Supposons que Γ' passe plusieurs fois par le même sommet v , i.e. qu'il existe des chemins Π_1, Π_2, Π_3 et des sommets $w_1, \dots, w_4 \in V$ tels que $\Gamma' = (v_0\Pi_1 w_1 v w_2 \Pi_2 w_3 v w_4 \Pi_3 v_0)$.

Si $w_1 = w_2$, alors on remplace $w_1 v w_2$ par w_1 dans Γ' . Si $w_1 \neq w_2$, alors on remplace $w_1 v w_2$ par $w_1 w_2$ dans Γ' , cette transformation est représentée dans la figure 3.9. Dans les deux cas, grâce à l'inégalité triangulaire, on n'augmente pas la longueur du cycle.

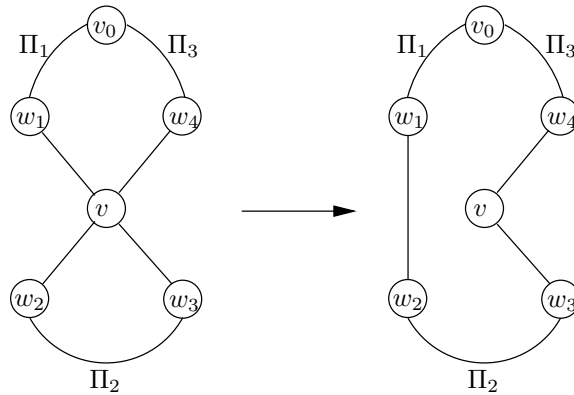


FIGURE 3.9 – Réduction du nombre d’occurrence d’un sommet $v \neq v_0$ dans la solution du problème de CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE.

Ainsi, on peut transformer Γ' en un cycle hamiltonien de longueur inférieure ou égale à $L + 2D$. En ôtant v_0 de ce cycle cela définit donc un chemin hamiltonien de G de longueur inférieure ou égale à L .

Ceci prouve que problème de CONCEPTION DE RÉSEAU DE TRANSPORT AVEC GESTION DE FLOTTE est au moins \mathcal{NP} -complet. \square

3.2.3 Complexité théorique et pratique due aux contraintes spécifiques

Dans la première partie de cette thèse, nous avons démontré que 3 différents problèmes, sous-problèmes du problème industriel de conception de réseau de service avec gestion de flotte, étaient difficile.

Le premier problème concerne le problème de transport avec coût fixe. Dès lors que le problème comporte une part de *design* avec coût fixe associé, le problème devient difficile.

Le second problème concerne la rotation des véhicules. Nous avons vu que ce problème pouvait se ramener à un problème de décomposition d’un graphe en circuits de longueur bornée et qu’il s’agissait d’un problème difficile. Cependant, **en pratique** la contrainte ne devrait pas poser trop de problèmes. En effet, le nombre de terminaux dans lesquels il y a initialement des véhicules majore la longueur des cycles. Ainsi, si les véhicules sont initialement positionnés dans deux terminaux, les cycles seront au plus de longueur deux. Il se peut aussi qu’on ne génère que des cycles de longueur 1, auquel cas la contrainte sur la longueur des cycles est automatiquement vérifiée. A priori pour des instances avec une période longue (une semaine ou plus) cette restriction ne devrait pas trop dégrader les solutions. Par contre, dans le cas du transport express où la période est de seulement 1 jour, n’autoriser que des cycles sur un jour interdit certaines destinations. En effet, certains trajets aller-retour ne peuvent pas être réalisés en une journée.

Le troisième problème s'attache à l'équilibrage des charges entre véhicules. Là encore, le problème se modélise comme un problème de décomposition de graphe, cette fois-ci il s'agit d'une décomposition en chaînes de longueur bornée. Bien qu'il s'agisse de problème en théorie difficile, l'équilibrage des charges entre véhicules s'intègre directement dans les modèles. Dans une formulation chemin, on ne générera que les chemins vérifiant des contraintes de longueur (minimum et maximum). Ceci limite par ailleurs le nombre de chemin, c'est donc une contrainte appréciée pour ce type de formulations.

3.3 Etat de l'art

Nous aborderons deux points dans cet état de l'art. Dans un premier temps, nous décrivons un ensemble de problèmes issus des systèmes de transport de marchandises, et organisons ces problèmes selon les différents niveaux de planification. Les activités et systèmes de transport sont riches et variés, et nous illustrons cette richesse du point de vue de la Recherche Opérationnelle. Dans un second temps, nous faisons un état de l'art plus approfondi et spécifique des problèmes de conception de réseau et de conception de réseau de service.

3.3.1 Systèmes de transport de marchandises

[Crainic et Semet, 2005] définit un *système de transport de marchandises* comme étant un système qui permet l'acheminement de biens (ou de services) entre les lieux de production et de consommation. Des exemples classiques de tels systèmes sont : le transport international et multimodal de conteneurs, le transport de fret par rail ou par camion, les systèmes de courrier express et régulier, la collecte d'ordure ménagère... Parmi ces systèmes, on distingue le transport de marchandises longue distance (comme le fret long courrier), et les activités de collecte et distribution s'effectuant généralement à l'échelle locale (comme la *city logistics*).

Comme pour le transport de passagers, les services sont de deux types.

- Le transport "porte-à-porte" (*Truckload* s'il s'agit de transport par camions), se rapproche du taxi avec lequel on a planifié à l'avance son trajet. Un véhicule (convoi, service) est consacré à la demande d'un seul client. On retrouve par exemple le camionnage en charge pleine, le transport express de marchandises.
- Le transport avec consolidation, qui se rapproche du transport en commun. Les charges de plusieurs clients sont regroupées, consolidées, et transitent ensemble à bord des mêmes véhicules (même convoi). On retrouve le transport par rail, les lignes régulières de bateau, le *Less-Than-Truckload motor carriers*.

L'objectif du paragraphe suivant est de décrire le transport avec consolidation.

Transport avec consolidation. Un véhicule prend en charge des demandes de différents clients. En général, le transporteur s'organise et propose un service régulier, il doit décider quelles routes, quelles fréquences de services, et quels plannings mettre en place.

Les transporteurs de marchandises consolidées sont organisés sous la forme de réseaux en étoile dit *hub-and-spoke*, illustrés figure 3.10.

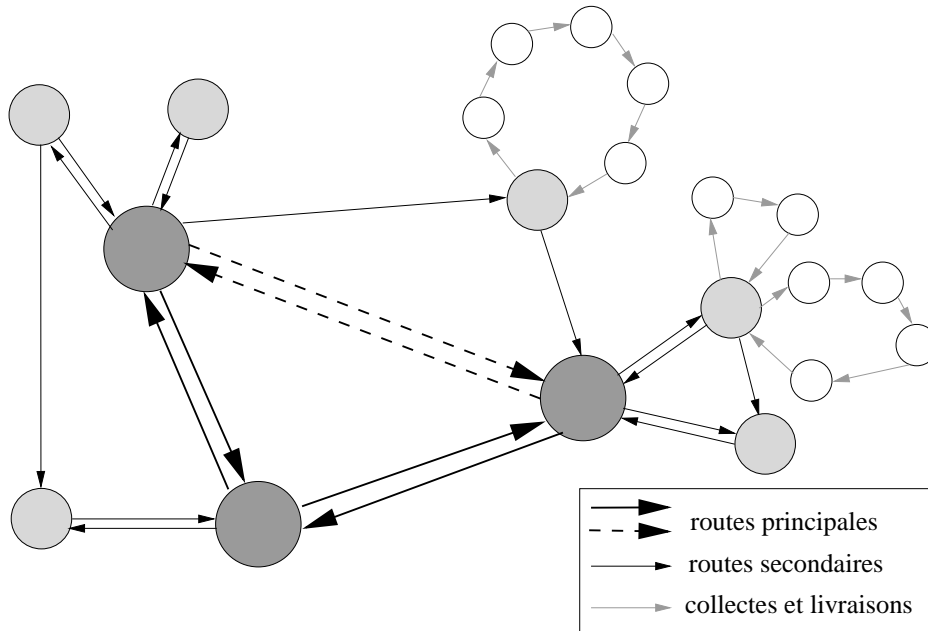


FIGURE 3.10 – Réseau *hub-and-spoke*.

L'avantage principal est que ce type de réseau favorise les économies d'échelles. Des flux plus importants sont triés et groupés dans les terminaux (hubs), et sont acheminés vers d'autres terminaux via des services à grande fréquence et grande capacité. Par contre, les itinéraires sont souvent plus longs et le coût et temps des opérations dans les terminaux peuvent aussi être pénalisant. De plus, les risques de pertes et de dégradations des colis sont plus importants.

Les terminaux jouent ainsi un rôle central et essentiel dans les systèmes avec consolidation : ils permettent de trier les marchandises, de consolider le fret via des opérations de transbordement, de regrouper les véhicules en convois, de réarranger et modifier ces convois. Ceci nécessite donc une multitude d'équipements spécifiques, et engendre un grand nombre de problèmes de gestion.

Les systèmes avec consolidation permettent, à condition d'une bonne planification, d'une bonne mise en oeuvre de cette planification et d'une gestion efficace des opérations, notamment dans les terminaux, de réduire les coûts pour le client et pour le transporteur. Ils offrent également des solutions moins polluantes, le prix à payer étant une réduction de la flexibilité pour le client, qui ne sera satisfait que si la qualité de service promise (délais, fiabilité, ...) est respectée. Du côté du transporteur, des méthodes pour planifier et diriger les opérations sont essentielles, leur efficacité conditionnant le profit du transporteur.

Niveaux de planification. On retrouve les trois niveaux de planification selon la classification de [Crainic et Laporte, 1997].

Niveau stratégique. Associé à la planification à long terme, il s'agit en général de construire l'infrastructure physique du système. On y compte notamment les problèmes suivants :

- localisation de terminaux (plateforme multimodale, gare de fret, plateforme de crossdocking, ...);
- équipement des plateformes (type et quantité de grues);
- construction de lignes (routes, lignes de chemin de fer) supplémentaires;
- destruction, désaffectation de certaines infrastructures.

Niveau tactique. Associé à la planification à moyen terme, il s'agit en général de la gestion et du contrôle optimal des ressources, routes et planning. Un des points clé à ce niveau est de garantir la satisfaction de la compagnie en termes de qualité de service offerte à ses clients et en termes économiques. Le principal problème de la planification tactique est la conception de réseau de service. A ce niveau, on apporte des réponses à des questions telles que :

- Sur quelles routes doit-on offrir un service ?
- Quel type de service (mode) utiliser ?
- A quelle fréquence doit-on offrir le service et selon quels horaires ?
- Comment faire transiter les marchandises via les réseaux physiques et de service ?
- Comment répartir la charge sur les différents terminaux du système ?

La planification tactique produit un plan de transport qui garantit la qualité de service requise pour les clients et une utilisation efficace des ressources afin d'assurer les meilleurs profits pour le transporteur. Ainsi, les décisions prises au niveau tactique concernent principalement : la sélection des services à offrir et leur fréquence (ou horaire), la distribution du transport des demandes sur ces services, les politiques au niveau des terminaux (formation de convois, politique de consolidation, ...) et les politiques de repositionnement à vide.

Niveau opérationnel. Associé aux opérations, les problèmes opérationnels sont sous une forte influence du temps. Il s'agit en général de la gestion et du contrôle dynamique des ressources, des routes, des plannings. On retrouve les problèmes suivants.

- Problèmes dans le transport de fret maritime par conteneurs : ordonnancement des grues d'un port, affectation et ordonnancement des arrimages, gestion de l'espace de stockage.
- Problèmes de tournées de véhicules, ou *vehicle routing problem* décrit dans la section 3.3.3.

Notre problème de conception de réseaux de service pour le transport de fret se situe principalement au niveau tactique. Il intègre également quelques composantes plus opérationnelles, notamment la construction d'horaires pour les routes des véhicules. Les problèmes fondamentaux du domaine du transport apparaissent comme sous-problèmes :

- les multi-flots *multicommodity flow problem*,
- les problèmes de tournées de véhicules *vehicle routing problem*,
- la conception de réseau.

Dans la suite, nous proposons une revue de la littérature de ces problèmes et du problème de conception de réseau de service. Puis, nous replaçons notre problème par rapport à ceux de la littérature.

3.3.2 Flot multicommodités

Dans un réseau déjà conçu, avec des capacités sur les arcs, le problème résiduel est de faire transiter les demandes sur ce réseau. C'est un problème de flot multicommodités ou *multicommodity flow*. Dans le cas d'une seule marchandise, avec éventuellement plusieurs origines et destinations, le problème est le problème de flot. Ce problème est polynomial, même si l'on demande l'intégralité des variables de flots. Dès lors qu'il y a au moins deux marchandises, demander l'intégralité des variables de flot rend le problème \mathcal{NP} -complet, cf. [Even *et al.*, 1976]. [Assad, 1978] et [Ahuja *et al.*, 1993] proposent deux états de l'art incluant les modèles classiques et des méthodes de résolution pour les problèmes de multiflots.

Nous donnons ci-après une formulation classique du problème de multiflot sur un réseau avec capacité, avec les variables de flot sur les arcs.

On note $G = (N, A)$ le réseau, l'arc (ij) ayant la capacité u_{ij} . Chaque demande $k \in K$ doit être transportées de son origine o^k à sa destination d^k en quantité w^k . Les coûts de transport sont linéaires, on note CV_{ij}^k le coût de transport d'une unité de la demande k sur l'arc (ij) . On introduit les variables de flot x_{ij}^k qui représentent la quantité de marchandise k transportée sur l'arc (ij) .

Formulation arc du problème de multiflot avec capacité.

$$\text{Minimiser : } \sum_{k \in K} \sum_{ij \in A} CV_{ij}^k x_{ij}^k, \quad (3.1)$$

sous contraintes

flot de marchandises :

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{ji}^k = \begin{cases} w^k & \text{si } i = o^k, \\ -w^k & \text{si } i = d^k, \\ 0 & \text{sinon,} \end{cases} \quad \forall i \in N, \forall k \in K, \quad (3.2)$$

contraintes de capacité :

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} \quad \forall (ij) \in A, \quad (3.3)$$

espace des variables :

$$x_{ij}^k \geq 0, \quad \forall (ij) \in A, \forall k \in K. \quad (3.4)$$

L'objectif 3.1 est la minimisation des coûts de transport. Les contraintes 3.2 sont les contraintes de conservation du flot et de satisfaction des demandes. Le respect des capacités des arcs est garanti par la contrainte 3.3.

Les travaux sur la résolution efficace du problème de flot multicommodités ou sur des variantes adaptées pour des applications ainsi que des travaux plus théoriques à ce sujet sont encore très nombreux ces dernières années. On compte déjà plus de 1000 articles référencés par Google Scholar publiés entre 2007 et 2010.

3.3.3 Tournées de véhicules

Le problème de tournée de véhicules, *vehicle routing problem*, est à l'interface entre les niveaux de planification tactique et opérationnel. Ce problème classique d'optimisation combinatoire est une généralisation du problème de voyageur de commerce. Il est très largement étudié, notamment du fait de son grand nombre d'applications. Ce problème est défini selon [Ahuja *et al.*, 1993] comme suit. Etant donnés :

- un réseau défini par le graphe orienté $G = (N, A)$;
- une flotte de véhicules V domiciliés à un dépôt commun $d \in N$;
- un ensemble de sites clients j avec chacun une demande (besoin d'être visités) ;
- le coût fixe CF_{ij} pour aller du site i au site j .

Il s'agit de déterminer un ensemble de tournées pour livrer tous les clients avec un coût minimal. On s'intéresse ainsi principalement aux trajets (tournées) des véhicules, les trajets des marchandises sont implicites.

Les variantes de ce problème sont nombreuses. En voici quelques unes : tournées de véhicules avec capacités, avec contraintes de longueur, avec fenêtres de temps, avec flotte hétérogène, avec livraisons fractionnées, avec collectes et livraisons. Un état de l'art complet sur les types de modèles et les méthodes exactes et approchées de résolution est donné dans [Toth et Vigo, 2002] et [Laporte, 1992]. Les problèmes de tournées de véhicules forment une famille très riche de problèmes, pour la plupart \mathcal{NP} -difficile.

Les formulations sous forme de programmes linéaires en nombres entiers sont nombreuses. Nous donnons une formulation arc du problème classique de tournées de véhicules sans capacité. Pour cela, on introduit les variables binaires y_{ijv} , $y_{ijv} = 1$ si la tournée du véhicule v va de i à j en empruntant l'arc (ij) .

Formulation arc du problème de tournée de véhicules sans capacité.

$$\text{Minimiser : } \sum_{v \in V} \sum_{ij \in A} CF_{ij} y_{ijv}, \quad (3.5)$$

sous contraintes

contraintes de tournées :

$$\sum_{j \in \delta^+(d)} y_{djv} = 1, \quad \forall v \in V, \quad (3.6)$$

$$\sum_{i \in \delta^-(d)} y_{idv} = 1, \quad \forall v \in V, \quad (3.7)$$

$$\sum_{i \in \delta^-(j)} \sum_{v \in V} y_{ijv} = 1, \quad \forall v \in V, \forall j \in N, \quad (3.8)$$

contraintes d'élimination des sous-tours :

$$\sum_{i \in S} \sum_{j \in S} y_{ijv} \leq |S| - 1, \quad \forall v \in V, \forall S \subset N, 2 \leq |S| \leq |N| - 2, \quad (3.9)$$

espace des variables :

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A. \quad (3.10)$$

Les contraintes (3.6) et (3.7) assurent que chaque véhicule fasse une tournée qui commence et termine au dépôt. La contrainte (3.8) assure la visite de chaque client par exactement une unique tournée. La contrainte (3.9) garantit la connexité des tournées en empêchant la formation de sous-tours.

3.3.4 Conception de réseau

Les problèmes de conception de réseau (*network design*) sont très riches et sont omniprésents dans les applications aussi bien en planification de réseaux de télécommunications que de transports. En général, il s'agit de faire transiter des flux (marchandises, personnes, informations, signaux) en utilisant des équipements déjà installés et en installant éventuellement de nouveaux équipements. Il faut alors payer non seulement pour le transit du flot, mais aussi pour utiliser ou installer un équipement. L'objectif est de minimiser le coût total (coût de conception + coût de routage) tout en offrant une solution ayant la qualité de service requise.

En d'autres termes, le problème revient à sélectionner un sous-ensemble de connexions qui permettent de minimiser la somme des coûts linéaires de routage et des coûts fixes de conception. De façon générique, les problèmes de conception de réseaux sont des problèmes de graphes [Feremans *et al.*, 2003] : il faut identifier un sous-graphe optimal d'un certain graphe, en respectant des contraintes. De nombreux problèmes de graphes sont des problèmes particuliers de *network design*. Dans [Feremans *et al.*, 2003] on retrouve notamment les problèmes de graphe tels que le couplage parfait de coût min, la clique couvrante minimale, l'arbre couvrant de poids minimal ou l'arbre de Steiner, tous modélisés sous forme de problèmes de conception de réseau. Les problèmes de conception de réseaux sont \mathcal{NP} -difficiles [Magnanti et Wong, 1984], comme en particulier l'arbre couvrant de poids minimal. Dans la figure 3.11, on a représenté un réseau orienté et un choix de design sur ce réseau.

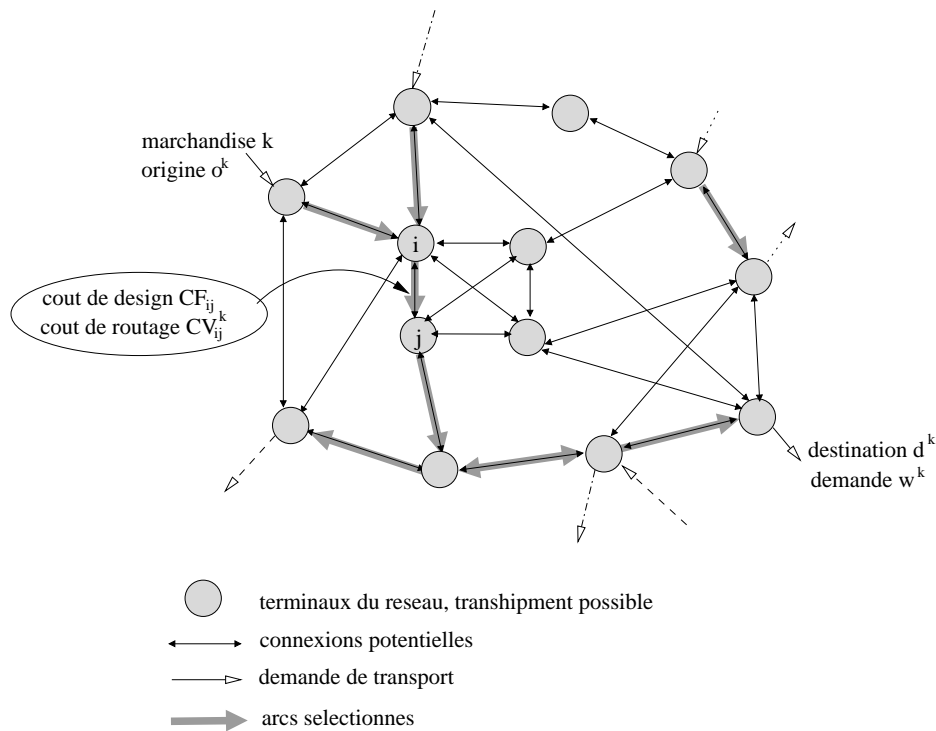


FIGURE 3.11 – Conception de réseau.

Des revues de la littérature ainsi que des exemples variés d'applications sont présentés dans les articles [Magnanti et Wong, 1984], [Minoux, 1989] ainsi que dans le chapitre 13 de l'ouvrage [Ahuja *et al.*, 1993]. Une synthèse de littérature annotée est proposée dans [Balakrishnan *et al.*, 1997]. Comme pour les problèmes de tournées de véhicules, on compte une large variété de problèmes classiques de conception de réseau. On distingue principalement deux grandes classes de problèmes de conception de réseau : la conception de réseau sans capacité sur les arcs (*uncapacitated*) et la conception de réseau avec capacité (*capacitated*).

Des algorithmes de résolution très efficaces ont été développés pour des problèmes de conception de réseau sans capacité sur les arcs *uncapacitated network design*. En particulier, [Balakrishnan *et al.*, 1989] propose une approche de *dual ascent* permettant d'obtenir rapidement des bornes inférieures à 1-4% de l'optimum. Cette approche est à la base d'heuristiques performantes pour la résolution d'instances de taille réelles de problèmes de camionage *Less-than-Truckload*.

Les problèmes de conception de réseau avec capacité sur les arcs *capacitated network design* sont nettement plus difficiles à résoudre. La difficulté majeure réside dans le calcul de bonnes bornes. Un état de l'art sur ces problèmes se trouve dans [Gendron *et al.*, 1998], qui propose également une approche par relaxation lagrangienne. Un cas particulier du problème de conception de réseau avec capacité est le problème de *network loading*. Le but est d'installer sur chacun des arcs un certain nombre d'infrastructures (par exemple

des câbles) qui ont une certaine capacité. Ces capacités sont en général particulières, multiples les unes des autres. Ces problèmes particuliers sont, là encore, bien traités dans la littérature, aussi bien par des approches polyédrales [Magnanti *et al.*, 1995] que par une heuristique de type tabou [Berger *et al.*, 2000].

Formellement le problème de conception de réseau orienté avec capacité est défini comme suit. Le réseau est modélisé par un graphe orienté $G = (N, A)$. Un ensemble K de demandes doit être satisfait, chaque demande $k \in K$ doit être transportée en quantité w^k de son origine o^k à destination d^k . Les coûts de transports sont linéaires, on note $CV_{ij}^k > 0$ le coût unitaire de transport de la marchandise k sur l'arc (ij) . On note CF_{ij} le coût fixe d'ouverture de l'arc (ij) , ce coût est à payer dès lors que l'arc (ij) est utilisé. On note u_{ij} la capacité de l'arc (ij) . L'objectif est de minimiser la somme des coûts variables de routage du flot et des coûts fixes de design.

Nous présentons une modélisation MIP du problème de conception de réseau avec capacité et coût fixe. Pour cette formulation, nous utilisons des variables de flot $x_{ij}^k \geq 0$ qui décrivent le transport de la marchandise k sur l'arc (ij) . Nous introduisons également des variables binaires de design y_{ij} qui déterminent les connexions ouvertes.

Formulation arc-arc du problème de conception de réseau avec capacité.

$$\text{Minimiser : } \sum_{ij \in A} CF_{ij} y_{ij} + \sum_{k \in K} \sum_{ij \in A} CV_{ij}^k x_{ij}^k \quad (3.11)$$

sous contraintes

flot de marchandises :

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{ji}^k = \begin{cases} w^k & \text{si } i = o^k, \\ -w^k & \text{si } i = d^k, \\ 0 & \text{sinon,} \end{cases} \quad \forall i \in N, \forall k \in K, \quad (3.12)$$

contraintes liantes :

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij} \quad \forall (ij) \in A, \quad (3.13)$$

espace des variables :

$$x_{ij}^k \geq 0, \quad \forall (ij) \in A, \forall k \in K, \quad (3.14)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (ij) \in A. \quad (3.15)$$

On retrouve les contraintes de flot (3.12). Le rôle de la contrainte (3.13) est double. C'est non seulement la contrainte de capacité mais surtout une contrainte forçante et liante : aucune marchandise ne peut transiter via un arc qui n'est pas ouvert. La fonction objectif (3.11) vise à trouver le meilleur compromis entre coût fixe de design du réseau et coût variable linéaire de transport.

3.3.5 Conception de réseau de service

La spécificité supplémentaire dans la conception de réseau de service *service network design* est que des ressources (véhicules) doivent être affectées lors de la conception du réseau. Pour qu'un arc soit ouvert, il faut opérer cet arc avec un véhicule. Ceci implique de nouvelles contraintes. En particulier, dans la conception de réseau de service le choix des arcs opérés doit permettre de définir des routes pour les véhicules. Pour cela, on demande un équilibre entre le nombre de véhicules qui arrivent à un nœud et le nombre qui repartent de ce nœud, c'est la contrainte dite de *design balance*.

La figure 3.12 reprend le même réseau orienté que pour la conception de réseau, mais cette fois ci on a représenté un design de réseau de service. Comparé au réseau conçu figure 3.11, le réseau de service figure 3.12 comporte plus d'arcs. Cette solution est plus chère mais elle intègre déjà la gestion des véhicules, en intégrant des repositionnement qui s'avèreraient encore plus coûteux dans une gestion a posteriori. On remarque notamment que le réseau conçu est modifié : l'arc $(o^k i)$ choisi dans la conception de réseau ne l'est plus dans le réseau de service.

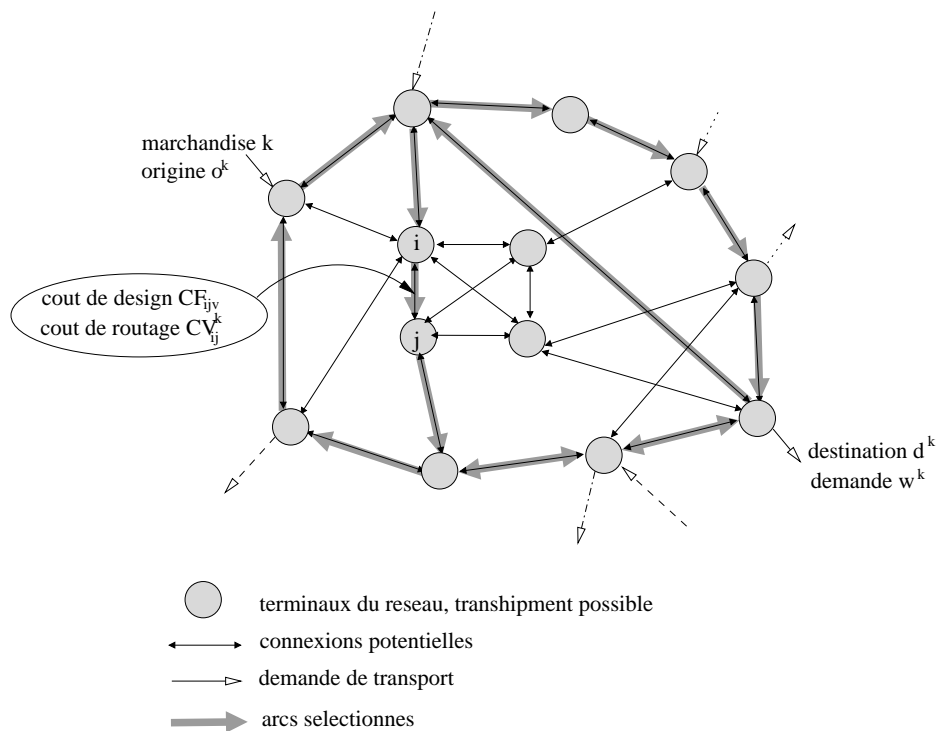


FIGURE 3.12 – Conception de réseau de service.

Définition. Un *service* est défini comme étant un trajet spécifique (d'un terminal de départ vers un terminal d'arrivée) à des horaires spécifiques (où avec une certaine fréquence) avec une certaine qualité de service (classe de confort par exemple).

La conception de réseau de service vise ainsi à apporter une aide à la décision en

répondant notamment aux questions suivantes :

- Quels sont les services (trajets et horaires/fréquences) ouverts ?
- Quels véhicules (avec quelle capacité) doivent être utilisés pour opérer ces services ?
- Comment faire transiter les demandes à travers le réseau de service en assurant le niveau de qualité de service requis (délais, fiabilité, ...) ?
- Quelles sont les meilleures stratégies de repositionnement des véhicules pour satisfaire les futures demandes ?

Une vue globale de la conception de réseau de service ainsi qu'un état de l'art sont donnés dans [Crainic, 2000] qui place clairement ce problème comme étant le problème majeur de la planification tactique. La conception de réseau de service est classiquement utilisée pour résoudre des problèmes de transport de fret avec consolidation : par train [Barnhart *et al.*, 2000], par avion [Barnhart *et al.*, 2002, Jaillet *et al.*, 1996] et aussi par camion [Favolden et Powell, 1994]. Un des problèmes classiques de conception de réseau de service est le transport express de colis ou lettres comme par exemple UPS [Kim, 1997], ou Deutsche Post AG [Irnich, 2002]. Un état de l'art récent est proposé par [Wieberneit, 2008].

Nous présentons une modélisation MIP du problème de conception de réseau de service avec capacité et coût fixe. On reprend les notations du problème de conception de réseau. On introduit un ensemble de ressources (véhicules) V , chaque ressource $v \in V$ a une capacité u_v . Les variables de flot restent $x_{ij}^k \geq 0$. Un indice supplémentaire, indiquant la ressource affectée au service, est ajouté aux variables binaires de design. Ainsi la variable binaire y_{ijv} indique si le service (ij) est opéré par la ressource v .

Formulation arc-arc du problème de conception de réseau de service.

$$\text{Minimiser : } \sum_{ij \in A} CF_{ijv} y_{ijv} + \sum_{k \in K} \sum_{ij \in A} CV_{ij}^k x_{ij}^k, \quad (3.16)$$

sous contraintes

flot de marchandises :

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{ji}^k = \begin{cases} w^k & si & i = o^k, \\ -w^k & si & i = d^k, \\ 0 & sinon, \end{cases} \quad \forall i \in N, \forall k \in K, \quad (3.17)$$

contraintes liantes :

$$\sum_{k \in K} x_{ij}^k \leq \sum_{v \in V} u_v y_{ijv}, \quad \forall (ij) \in A, \quad (3.18)$$

contraintes de *design balance* :

$$\sum_{v \in V} \sum_{j \in N} y_{ijv} - \sum_{v \in V} \sum_{j \in N} y_{jiv} = 0, \quad \forall i \in N, \quad (3.19)$$

espace des variables :

$$x_{ij}^k \geq 0, \quad \forall (ij) \in A, \forall k \in K, \quad (3.20)$$

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A, \forall v \in V. \quad (3.21)$$

La fonction objectif (3.16) et les contraintes (3.17) et (3.18) sont inchangées par rapport au modèle de conception de réseau. La nouvelle contrainte (3.19) est la contrainte de design balance, contrainte qui différencie la conception de réseau de service de la conception de réseau. Cette contrainte oblige un nombre égal de véhicules qui entrent et sortent de chaque terminal dans le réseau. Ces contraintes ont été utilisées dans les modèles pour une conception d'un réseau de service pour un transport express de fret par voie aérienne dans [Barnhart et Schneur, 1996] et [Barnhart *et al.*, 2002], par la route [Farvolden et Powell, 1994] et pour une application de service de transport de passagers par ferry dans [Lai et Lo, 2004, Wang et Lo, 2008].

3.3.6 Positionnement de notre problème de transport de fret

Notre problème de conception de réseau de service pour le transport de fret est clairement un problème de conception de réseau de service. Cependant, il comporte quelques spécificités supplémentaires. La première spécificité concerne la gestion des véhicules. Pour permettre une planification régulière, le plan de transport calculé sur une période doit pouvoir être répété de période en période. Pour ce faire, il faut qu'au début de chaque période les véhicules soient répartis de façon analogue. C'est-à-dire qu'on retrouve le même nombre de véhicules du même type dans chaque terminal au début et à la fin de la période de planification, i.e. le début de la période suivante. Ce n'est pas nécessairement le même véhicule qui doit être au même endroit, les véhicules pouvant alterner d'une période à la suivante. Dans le cadre de notre application, le temps avant lequel un véhicule retrouve sa route initiale est borné par un nombre N de périodes. En pratique les véhicules cyclent rarement sur plus de 3 à 4 périodes. [Chou *et al.*, 2003] donnent un exemple de rotation des véhicules pour le transport maritime, et [Yan *et al.*, 2005] pour le transport aérien. Cela se produit rarement pour le transport routier. En particulier, le problème comporte explicitement une dimension temporelle. Cette gestion supplémentaire des véhicules définit le problème de conception de réseau avec gestion de ressources *service network design with asset management* [Andersen *et al.*, 2009b, Pedersen *et al.*, 2007, Andersen *et al.*, 2009a]. Notre problème s'inscrit dans ce cadre.

La seconde contrainte spécifique sur les véhicules est que ceux-ci doivent être utilisés de façon équitable : à la fin d'une saison aucun véhicule ne doit être ni sous- ni sur-exploité. La longueur du chemin opéré par le véhicule v pendant une période est bornée inférieurement par $QuotaMin_v$ et supérieurement par $QuotaMax_v$. Dans la littérature sur les problèmes de tournées de véhicules, cette contrainte est appelée la contrainte de temps ou de distance [Laporte, 1992].

La structure de coût est classique, un coût variable linéaire avec le flux de produits et un coût fixe d'ouverture de service. La spécificité réside dans le fait que l'on cherche à maximiser un profit, et non à seulement minimiser le coût. Une recette est engrangée par le transporteur pour chaque unité de fret transportée jusqu'à sa destination. Ainsi, le transporteur doit faire un arbitrage entre coûts de transport et recettes pour maximiser son profit, il doit donc satisfaire les demandes les plus "rentables" pour lui, et rejeter partiellement ou totalement certaines demandes. Il y a donc là une décision supplémentaire à prendre. Cette caractéristique, de satisfaire ou non une demande, est connue comme

price collecting ou *orienteeing tour problem*, elle a été intégrée par [Feillet *et al.*, 2005] et [Archetti *et al.*, 2008] dans différents problèmes de tournées de véhicules.

Chapitre 4

Etude comparée de différentes formulations

Sommaire

4.1	Quatre formulations du problème	72
4.1.1	Formulation <i>arc-arc</i>	72
4.1.2	Formulation <i>cycle-arc</i>	74
4.1.3	Formulation <i>arc-chemin</i>	76
4.1.4	Formulation <i>cycle-chemin</i>	78
4.2	Comparaison théorique des formulations	79
4.2.1	Taille des formulations	79
4.2.2	Comparaison théorique des formulations	80
4.3	Comparaison expérimentale des modèles	82
4.3.1	Instances	82
4.3.2	Résultats sur nos instances	83

Le but de ce chapitre est de présenter et de comparer quatre formulations en programme linéaire en nombre entiers mixte (MIP) du problème de conception de réseau de service avec gestion de flotte. Cette étude comparative est similaire à celle de [Andersen *et al.*, 2009b]. Nous nous focalisons sur quatre formulations, qui combinent les différentes représentations des variables de conception du réseau et des variables de flot. Les variables de décision concernant la conception du réseau portent soit sur les arcs de ce réseau soit sur des chemins potentiels pour les véhicules. Dans [Andersen *et al.*, 2009b] ces chemins sont des cycles sur une période, pour nous ce sont des chemins de longueur une période, morceaux d'un cycle sur plusieurs périodes. Les variables de flot sont soit des variables arcs, soit des variables chemins. Nous comparons expérimentalement ces modèles, ce qui nous permet de valider mais aussi de nuancer les résultats de [Andersen *et al.*, 2009b] quant à la qualité de la formulation à base de morceaux de cycles. De plus, nous proposons une preuve mathématique qui valide le fait que la relaxation linéaire des formulations fondées sur les cycles donnent une meilleure borne que la relaxation linéaire des formulations correspondantes fondées sur des variables arcs.

Ce chapitre constitue une de nos contributions majeures sur les problèmes de conception de réseau de service avec gestion de flotte. Les résultats concernant la comparaison des modèles arc-arc et cycle-arc sont intégralement publiés [Schrenk *et al.*, 2010] dans le chapitre d'un n° spécial d'articles invités en l'honneur de Catherine Roucairol.

4.1 Quatre formulations du problème

Dans cette section on présente quatre formulations différentes de notre problème de transport. Ce sont les formulations de conception de réseau de service avec gestion de flotte proposées par [Andersen *et al.*, 2009b] et adaptées à notre problème. Pour ces formulations, les variables de conception du réseau (flot de véhicules) sont représentées soit sur les arcs du réseau espace-temps, soit sur des chemins (ou plus exactement sur des morceaux de cycle) de ce réseau. Les variables de flot de marchandises sont définies soit sur les arcs du réseau, ce qui permet notamment d'intégrer facilement le transbordement, soit sur des chemins de ce réseau. Les quatre formulations ainsi obtenues sont *arc-arc*, *cycle-arc*, *arc-chemin*, *cycle-chemin*.

Les successeurs et prédécesseurs d'un nœud dépendent du type de véhicule considéré. Dans le réseau espace-temps $G = (N, A)$ correspondant au véhicule v , étant donné un nœud $i \in V$, on note $\delta_v^+(i) = \{j \in N : (ij) \in A\}$ et $\delta_v^-(i) = \{j \in N : (ji) \in A\}$ respectivement, l'ensemble de ses successeurs et de ses prédécesseurs. On note également $b_{ijv}^k = \min\{u_{ijv}, w^k\}$, le minimum entre la capacité du véhicule v pour l'arc de service (ij) et la quantité totale de la demande k .

4.1.1 Formulation *arc-arc*

La variable binaire y_{ijv} indique si l'arc de service (ij) est opéré par le véhicule v . Ces variables définissent le réseau de service, elles font le choix des routes pour les véhicules. La variable binaire δ_v indique si oui ou non le véhicule v est utilisé. Le flot de marchandises est représenté par les variables $x : x_{ijv}^k$ correspond à la quantité de marchandise de la demande k chargée à bord du véhicule v entre les nœuds i et j du réseau espace-temps. La variable z^k représente la quantité totale de marchandise de type k transportée sur une période de planification, i.e. la quantité de demande k qui est satisfaite. La formulation arc-arc s'écrit alors comme suit.

$$\begin{aligned} \text{Maximiser : } & + \sum_{k \in K} P^k \cdot z^k - \sum_{v \in V} \sum_{(ij) \in A} \sum_{k \in K} CV_{ijv} \cdot x_{ijv}^k \\ & - \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} \cdot y_{ijv}, \end{aligned} \tag{4.1}$$

4.1 Quatre formulations du problème

sous contraintes

flot de marchandises :

$$\sum_{v \in V} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = 0, \quad \forall k \in K, \forall i \in N, N_i \neq o^k \text{ et } N_i \neq d^k, \quad (4.2)$$

$$\sum_{v \in V} \sum_{i \in N, N_i = o^k} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{i \in N, N_i = o^k} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = z^k, \quad \forall k \in K, \quad (4.3)$$

$$\sum_{v \in V} \sum_{i \in N, N_i = d^k} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{i \in N, N_i = d^k} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = -z^k, \quad \forall k \in K, \quad (4.4)$$

flot de véhicules :

$$\sum_{j \in \delta_v^+(i)} y_{ijv} - \sum_{j \in \delta_v^-(i)} y_{jiv} = 0, \quad \forall i \in N, T_i \neq 1, \forall v \in V, \quad (4.5)$$

$$\sum_{v \in V_{type}} \sum_{j \in \delta_v^+(i)} y_{ijv} - \sum_{v \in V_{type}} \sum_{j \in \delta_v^-(i)} y_{jiv} = 0, \quad \forall i \in N, T_i = 1, \forall type, \quad (4.6)$$

$$\sum_{(i,j) \in A: T_i \leq t < T_j} y_{ijv} - \delta_v \leq 0, \quad \forall t \in T, \forall v \in V, \quad (4.7)$$

contraintes liantes de capacité :

$$\sum_{k \in K} x_{ijv}^k - u_{ijv} y_{ijv} \leq 0, \quad \forall (ij) \in A, \forall v \in V, \quad (4.8)$$

$$x_{ijv}^k - b_{ijv}^k y_{ijv} \leq 0, \quad \forall (ij) \in A, \forall v \in V, \forall k \in K, \quad (4.9)$$

contraintes spécifiques :

$$QuotaMin_v \cdot \delta_v \leq \sum_{(ij) \in A: N_i \neq N_j} y_{ijv} t_{ijv} \leq QuotaMax_v \cdot \delta_v, \quad \forall v \in V, \quad (4.10)$$

variables :

$$x_{ijv}^k \geq 0, \quad \forall (ij) \in A, \forall v \in V, \forall k \in K, \quad (4.11)$$

$$z^k \in [0..w^k], \quad \forall k \in K, \quad (4.12)$$

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A, \forall v \in V, \quad (4.13)$$

$$\delta_v \in \{0, 1\}, \quad \forall v \in V. \quad (4.14)$$

La fonction objectif (4.1) maximise le profit i.e. le revenu engendré par le transport auquel on soustrait les coûts de transport variables (flots) et fixes (coûts d'ouverture d'un service).

Les contraintes (4.2)-(4.4) sont les contraintes de flot sur les marchandises. Elles permettent le transbordement : une marchandise peut changer de véhicule. Cependant, elles n'autorisent pas de stockage temporaire d'une marchandise dans un terminal en dehors d'un véhicule.

Les contraintes (4.5) et (4.6) sont des contraintes de *design balance*. Toutes deux visent à équilibrer les départs et arrivées dans les terminaux. La première permet de définir les routes des véhicules à l'intérieur de la période de planification. Elle force l'équilibre entre

les arrivées et les départs des véhicules. Alors que la deuxième se focalise sur les routes des véhicules aux extrémités de la période. Ensemble elles permettent de générer implicitement des cycles pour les véhicules sur une ou plusieurs périodes. La contrainte (4.7) garantit qu'un véhicule utilisé opère au plus un service à la fois.

La contrainte (4.8) est la contrainte forçante : du flot ne peut circuler sur une liaison que si un véhicule l'opère. Cette contrainte, qui lie les deux flots, assure également le respect des capacités des véhicules sur les arcs opérés. La contrainte (4.9) est redondante. On l'introduit néanmoins pour obtenir une formulation dont la relaxation linéaire est plus forte. En effet, cette contrainte permet d'améliorer nettement la qualité de la relaxation linéaire comme le montrent [Crainic *et al.*, 2001] ainsi que nos résultats expérimentaux en section 4.2. Finalement, la contrainte (4.10) assure que les quotas d'utilisations (temps de trajet minimal et maximal) soient respectés pour les véhicules utilisés. Les contraintes de type des variables sont données de (4.11) à (4.13). En particulier, la variable binaire (4.14) permet de décider si un véhicule est utilisé ou non.

4.1.2 Formulation *cycle-arc*

Dans cette deuxième formulation, on définit les variables de conception non plus sur les arcs du réseau, mais sur des chemins potentiels pour les véhicules. Ces chemins sont des morceaux de cycles. On introduit les variables binaires g_v^p où g_v^p vaut 1 si le morceau de cycle p est opéré par le véhicule v , 0 sinon. Les morceaux de cycles potentiels dépendent du type de véhicule, pour plus de lisibilité, on note \mathcal{P}_v l'ensemble des morceaux de cycles possibles pour le véhicule v . Dans la figure 4.1, on a représenté trois chemins potentiels pour un véhicule. Il y en a beaucoup d'autres. On note r_{ij}^p le paramètre qui est égal à 1 si l'arc (ij) est dans le cycle p , 0 sinon.

Pour plus de lisibilité, on étend la fonction coût fixe aux chemins : CF_v^p est le coût fixe d'ouverture du chemin p par le véhicule v , c'est la somme des coûts fixes des arcs composant ce chemins.

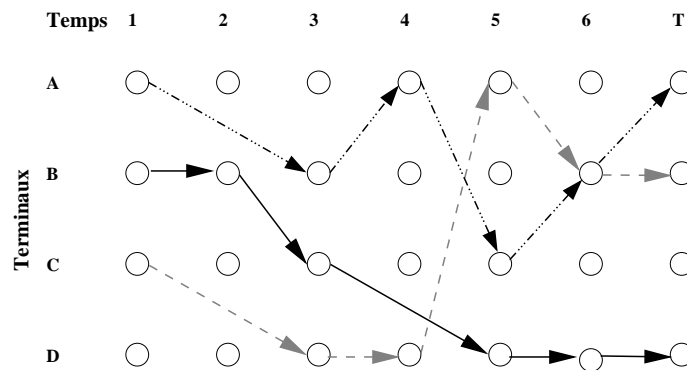


FIGURE 4.1 – Morceaux de cycles possibles pour un véhicule dans le réseau espace-temps correspondant.

4.1 Quatre formulations du problème

La formulation cycle-arc du problème s'écrit alors comme suit.

$$\begin{aligned} \text{Maximiser : } & + \sum_{k \in K} P^k \cdot z^k - \sum_{v \in V} \sum_{(ij) \in A} \sum_{k \in K} CV_{ijv} \cdot x_{ijv}^k \\ & - \sum_{v \in V} \sum_{p \in \mathcal{P}_v} CF_v^p \cdot g_v^p, \end{aligned} \quad (4.15)$$

sous contraintes

flot de marchandises, identique à (4.2), (4.3) et (4.4) :

$$\sum_{v \in V} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = 0, \quad \forall k \in K, \forall i \in N, N_i \neq o^k \text{ et } N_i \neq d^k, \quad (4.16)$$

$$\sum_{v \in V} \sum_{i \in N, N_i = o^k} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{i \in N, N_i = o^k} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = z^k, \quad \forall k \in K, \quad (4.17)$$

$$\sum_{v \in V} \sum_{i \in N, N_i = d^k} \sum_{j \in \delta_v^+(i)} x_{ijv}^k - \sum_{v \in V} \sum_{i \in N, N_i = d^k} \sum_{j \in \delta_v^-(i)} x_{jiv}^k = -z^k, \quad \forall k \in K, \quad (4.18)$$

flot de véhicules :

$$\sum_{p \in \mathcal{P}_v} g_v^p \leq 1, \quad \forall v \in V, \quad (4.19)$$

$$\begin{aligned} \sum_{v \in V_{type}} \sum_{p \in \mathcal{P}_v} \sum_{j \in \delta_v^+(i)} g_v^p r_{ij}^p - \sum_{v \in V_{type}} \sum_{p \in \mathcal{P}_v} \sum_{j \in \delta_v^-(i')} g_v^p r_{i'j}^p = 0, \quad \forall i \in N, T_i = 1, \\ i' = (N_i, T), \forall type, \end{aligned} \quad (4.20)$$

contraintes liantes de capacité :

$$\sum_{k \in K} x_{ijv}^k - \sum_{p \in \mathcal{P}_v} u_{ijv} r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, v \in V, \quad (4.21)$$

$$x_{ijv}^k - \sum_{p \in \mathcal{P}_v} b_{ijv}^k r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, \forall k \in K, \forall v \in V, \quad (4.22)$$

contraintes spécifiques :

$$QuotaMin_v \leq \sum_{p \in \mathcal{P}_v} \sum_{(ij) \in A; N_i \neq N_j} g_v^p r_{ij}^p t_{ijv} \leq QuotaMax_v, \quad \forall v \in V, \quad (4.23)$$

variables :

$$x_{ijv}^k \geq 0, \quad \forall (ij) \in A, \forall v \in V, \forall k \in K, \quad (4.24)$$

$$z^k \in [0..w^k], \quad \forall k \in K, \quad (4.25)$$

$$g_v^p \in \{0, 1\}, \quad \forall v \in V, \forall p \in \mathcal{P}_v. \quad (4.26)$$

Dans (4.15) on retrouve donc l'écriture de la fonction objectif qui maximise le profit. Les contraintes décrivant le flot de marchandises (4.16)-(4.18) sont inchangées par rapport à la formulation arc-arc (4.2)-(4.4). Le flot de véhicules est défini de façon implicite via les variables chemin g et la contrainte (4.19) assure qu'un véhicule n'opère pas plus d'un

chemin sur une période. La rotation des véhicules correspond à la contrainte (4.20). Comme pour la formulation arc-arc, on retrouve les contraintes liantes et les contraintes de capacité faibles (4.21) et fortes (4.22). Finalement, on retrouve la contrainte de quotas d'utilisation (4.23). En pratique, cette contrainte ne concernant que les cycles, elle est directement incluse dans la génération des morceaux de cycle pour les véhicules. Ceci permet de réduire le nombre de contraintes, mais surtout cela limite le nombre de chemins potentiels à envisager et réduit d'autant le nombre de variables binaires.

4.1.3 Formulation *arc-chemin*

Les écritures ci-dessus sont des écritures "arcs" pour le flot de marchandises, on peut les transformer en écritures "chemins". On note alors h_v^{kl} le flot de marchandise k empruntant le chemin l opéré par le véhicule v . \mathcal{L}_v^k est l'ensemble des chemins pour le produit k avec le véhicule v .

Dans cette formulation le transbordement n'est pas pris en compte. Un chemin est opéré par un seul véhicule, il est nécessaire de mettre v en indice des chemins pour pouvoir gérer les véhicules et calculer la fonction objectif. Les formulations avec variables chemins ne sont donc pas équivalentes à leurs homologues avec variables arcs, puisque le transbordement n'est pas permis. Nous avons quand même fait le choix de présenter ces modèles et d'analyser leur qualité, ils peuvent notamment permettre de gérer des fenêtres de temps, contraintes souvent au cœur des problèmes pratiques.

Les variables de flot sont indicées par des chemins. Il faut aussi un indice correspondant au véhicule. On note h_v^{kl} la quantité de la marchandise k qui emprunte le chemin l avec le véhicule v . On note \mathcal{L}_v^k , l'ensemble des chemins qui permettent de transporter la marchandise de son origine ($o^k, t = 0$) à sa destination ($d^k, t = T$) dans le réseau espace-temps du véhicule v . De plus, comme précédemment, on introduit la variable z^k qui représente la fraction de la demande k satisfaite.

Dans la figure 4.2 on donne un exemple de différents chemins possibles pour une marchandise. Ces chemins vont de l'origine de la marchandise à sa destination et transitent éventuellement par d'autres terminaux.

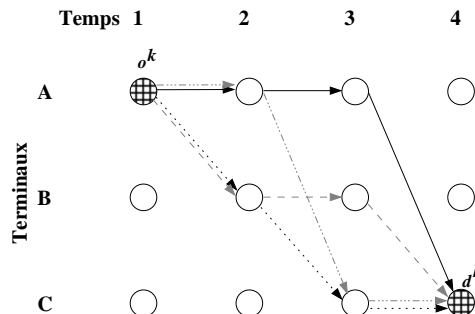


FIGURE 4.2 – 4 chemins parmi les 6 possibles pour la marchandise $k : A \rightarrow C$.

4.1 Quatre formulations du problème

Le modèle arc-chemin s'écrit comme suit.

$$\begin{aligned} \text{Maximiser : } & + \sum_{k \in K} P^k \cdot z^k - \sum_{v \in V} \sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} CV^l \cdot h_v^{kl} \\ & - \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} \cdot y_{ijv}, \end{aligned} \quad (4.27)$$

sous contraintes

flot de marchandises :

$$\sum_{v \in V} \sum_{l \in \mathcal{L}_v^k} h_v^{kl} = z^k, \quad \forall k \in K, \quad (4.28)$$

contraintes liantes de capacité :

$$\sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - u_{ijv} y_{ijv} \leq 0, \quad \forall (ij) \in A, \forall v \in V, \quad (4.29)$$

$$\sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - b_{ijv}^k y_{ijv} \leq 0, \quad \forall (ij) \in A, \forall k \in K, \forall v \in V, \quad (4.30)$$

flot de véhicules, identique à (4.5), (4.6) et (4.7) :

$$\sum_{j \in \delta_v^+(i)} y_{ijv} - \sum_{j \in \delta_v^-(i)} y_{jiv} = 0, \quad \forall i \in N, T_i \neq 1, \forall v \in V, \quad (4.31)$$

$$\sum_{v \in V_{type}} \sum_{j \in \delta_v^+(i)} y_{ijv} - \sum_{v \in V_{type}} \sum_{j \in \delta_v^-(i)} y_{jiv} = 0, \quad \forall i \in N, T_i = 1, \forall type, \quad (4.32)$$

$$\sum_{(i,j) \in A: T_i \leq t < T_j} y_{ijv} - \delta_v \leq 0, \quad \forall t \in T, \forall v \in V, \quad (4.33)$$

contraintes spécifiques, identique à (4.10) :

$$QuotaMin_v \cdot \delta_v \leq \sum_{(ij) \in A: N_i \neq N_j} y_{ijv} t_{ijv} \leq QuotaMax_v \cdot \delta_v, \quad \forall v \in V, \quad (4.34)$$

variables :

$$h_v^{kl} \geq 0 \quad \forall k \in K, \forall l \in \mathcal{L}_v^k, \forall v \in V, \quad (4.35)$$

$$z^k \in [0..w^k], \quad \forall k \in K, \quad (4.36)$$

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A, \forall v \in V, \quad (4.37)$$

$$\delta_v \in \{0, 1\}, \quad \forall v \in V. \quad (4.38)$$

Dans (4.27) on retrouve l'écriture de la fonction objectif qui maximise le profit. La contrainte (4.28) permet de définir la quantité totale de chaque demande qui est satisfaite, elle remplace les contraintes de flot qui sont implicites dans la formulation chemin. Les contraintes concernant les véhicules (4.31)-(4.34) sont identiques à (4.5)-(4.10). On retrouve les contraintes liantes et les contraintes de capacité faibles (4.29) et fortes (4.30).

4.1.4 Formulation *cycle-chemin*

Comme précédemment, la variable binaire g_v^p vaut 1 si le cycle p est assuré par le véhicule v , 0 sinon. On note r_{ij}^p le paramètre égal à 1 si l'arc (ij) est dans le cycle p , 0 sinon.

h_v^{kl} est la quantité de marchandise k transportée à bord du véhicule v sur le chemin l . La formulation cycle-chemin s'écrit comme suit.

$$\begin{aligned} \text{Maximiser : } & + \sum_{k \in K} P^k \cdot z^k - \sum_{v \in V} \sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} CV^l \cdot h_v^{kl} \\ & - \sum_{v \in V} \sum_{p \in \mathcal{P}_v} CF_v^p \cdot g_v^p, \end{aligned} \quad (4.39)$$

sous contraintes

flot de marchandises, identique à (4.28) :

$$\sum_{v \in V} \sum_{l \in \mathcal{L}_v^k} h_v^{kl} = z^k, \quad \forall k \in K, \quad (4.40)$$

contraintes liantes de capacité :

$$\sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - \sum_{p \in \mathcal{P}_v} u_{ijv} r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, \forall v \in V, \quad (4.41)$$

$$\sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - \sum_{p \in \mathcal{P}_v} b_{ijv}^k r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, \forall k \in K, \forall v \in V, \quad (4.42)$$

flot de véhicules, identique à (4.19) et (4.20) :

$$\sum_{p \in \mathcal{P}_v} g_v^p \leq 1, \quad \forall type, \quad (4.43)$$

$$\sum_{v \in V_{type}} \sum_{p \in \mathcal{P}_v} \sum_{j \in \delta_v^+(i)} g_v^p r_{ij}^p - \sum_{v \in V_{type}} \sum_{p \in \mathcal{P}_v} \sum_{j \in \delta_v^-(i')} g_v^p r_{ij}^p = 0, \quad \forall i \in N, T_i = 1, \quad (4.44)$$

$i' = (N_i, T), \forall type,$

contraintes spécifiques, identique à (4.23) :

$$QuotaMin_v \leq \sum_{(ij) \in A; N_i \neq N_j} g_v^p r_{ij}^p t_{ijv} \leq QuotaMax_v, \quad \forall v \in V, \quad (4.45)$$

variables :

$$h_v^{kl} \geq 0 \quad \forall k \in K, \forall l \in \mathcal{L}_v^k, \forall v \in V, \quad (4.46)$$

$$z^k \in [0..w^k], \quad \forall k \in K, \quad (4.47)$$

$$g_v^p \in \{0, 1\}, \quad \forall p \in \mathcal{P}_v. \quad (4.48)$$

Le modèle cycle-chemin est une combinaison des modèles cycle-arc et arc-chemin. La fonction objectif (4.39) maximise le profit. La satisfaction éventuellement partielle des demandes est donnée dans (4.40). Les contraintes (4.41) et (4.42) sont les contraintes de capacité faibles et fortes. La taille de la flotte est respectée grâce à (4.43). La contrainte (4.44) assure le cyclage des véhicules.

4.2 Comparaison théorique des formulations

4.2.1 Taille des formulations

Le premier aspect pour comparer les formulations est leur taille détaillée dans le tableau 4.1. Cela permet au moins d'identifier les tailles d'instances limites pour que les modèles puissent être chargés en mémoire. En dehors des contraintes de capacité fortes, le nombre de contraintes est assez faible. Quant aux variables binaires, elles sont en nombre exponentiel dans les formulations à base de cycles.

	arc-arc	cycle-arc	arc-chemin	cycle-chemin
nb variables flot marchandises	$ A V K $	$ A V K $	$ K V \mathcal{L}_v^k $	$ K V \mathcal{L}_v^k $
nb variables satisfaction demande	$ K $	$ K $	$ K $	$ K $
nb variables binaires flot véhicules	$ A V $	$ type \mathcal{P}_{type} $	$ A V $	$ type \mathcal{P}_{type} $
nb contraintes taille flotte	$ V T $	$ V $	$ V T $	$ V $
nb contraintes flot véhicule	$ N $	-	$ N $	-
nb contraintes rotation véhicules	$n = N' $	$n = N' $	$n = N' $	$n = N' $
nb contraintes quotas	$ V $	-	$ V $	-
nb contraintes flot marchandises	$ N K $	$ N K $	$ K $	$ K $
nb contraintes capacité faibles	$ A V $	$ A V $	$ A V $	$ A V $
nb contraintes capacité fortes	$ A V K $	$ A V K $	$ A V K $	$ A V K $

TABLE 4.1 – Comparaison de la taille des modèles arc-arc, cycle-arc, arc-chemin et cycle-chemin.

Exemple du problème n5k15v3t5 :

Les instances de type n5k15v3t5 comportent $n = 5$ terminaux, $|K| = 15$ demandes, $|V| = 3$ véhicules du même type, sur une période T discrétisée en $|T| = 5$ intervalles de temps. Le réseau espace-temps contient donc $|N| = n * |T| = 5 * 5 = 25$ nœuds. Si l'on suppose qu'un véhicule peut se rendre d'un terminal à n'importe quel autre en un intervalle de temps, alors le nombre d'arcs dans le réseau espace-temps est $|A| = n * n * (|T| - 1) = 5 * 5 * 4 = 80$. Le nombre maximal de cycles pour les véhicules, en ignorant les contraintes spécifiques de quota et les temps de réengagement, est $n^{|T|} = 3125$. Les résultats expérimentaux montrent qu'en moyenne, en intégrant ces contraintes, le nombre de morceaux de cycles à considérer pour ces instances n'est plus que $|\mathcal{P}_{type}| \approx 500$.

Le nombre maximal de chemin pour une marchandise k à bord du véhicule v est de l'ordre de $n^{|T|-2} = 125$.

Dans le tableau 4.2, un ordre de grandeur des modèles pour ce type d'instance est donné. Ce tableau permet de mettre en évidence le grand nombre de contraintes de capacité fortes ainsi que le plus grand nombre de variables binaires pour le modèle à base de cycles.

	arc-arc	cycle-arc	arc-chemin	cycle-chemin
nb variables continues	3 215	3 215	5 640	5 640
nb variables binaires (design)	160	500	160	500
nb contraintes	482	467	303	263
nb contraintes capacité fortes	2 400	2 400	2 400	2 400

TABLE 4.2 – Comparaison de la taille des formulations arc-arc, cycle-arc, arc-chemin et cycle-chemin pour une instance de type n5k15v3t4.

4.2.2 Comparaison théorique des formulations

Théorème 4.2.1. Les modèles avec variables de design à base de cycles ont des relaxations linéaires plus fortes que leurs équivalents avec variables de design à base d'arc. Plus précisément :

- La relaxation linéaire du modèle cycle-arc offre une meilleure borne supérieure (problème de maximisation) que la relaxation linéaire du modèle arc-arc.
- La relaxation linéaire du modèle cycle-chemin offre une meilleure borne supérieure (problème de maximisation) que la relaxation linéaire du modèle arc-chemin.

Preuve : Nous démontrons le premier point du théorème. La preuve du second point est identique.

- On note \mathcal{P}_{aaLP} (respectivement \mathcal{P}_{caLP}) le polyèdre défini par les contraintes du modèle arc-arc (respectivement cycle-arc), avec les variables relaxées linéairement.

Il faut montrer que l'on peut transformer toute solution $(x, z, g) \in \mathcal{P}_{caLP}$ en une solution $(x, z, y, \delta) \in \mathcal{P}_{aaLP}$ et que les fonctions objectifs correspondantes vérifient l'inégalité *objectif caLP* \leq *objectif aaLP*. Pour cela, on construit une application affine \mathcal{T} , telle que $\mathcal{T}(\mathcal{P}_{caLP}) \subseteq \mathcal{P}_{aaLP}$.

Soit $(x, z, g) \in \mathcal{P}_{caLP}$, on définit \mathcal{T} par $\mathcal{T}(x, z, g) = (x, z, y, \delta)$ où x et z sont inchangées et

$$y_{ijv} = \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p$$

$$\delta_v = \sum_{p \in \mathcal{P}_v} g_v^p$$

Dans la suite nous montrons que \mathcal{T} , qui est une application affine par définition, est à images dans \mathcal{P}_{aaLP} . Pour cela, il faut démontrer que l'élément (x, z, y, δ) image par \mathcal{T} de (x, z, g) vérifie toutes les contraintes de la relaxation linéaire du modèle arc-arc. Au cours de la démonstration, on utilise bien entendu le fait que (x, z, g) vérifie toutes les contraintes de la relaxation linéaire du modèle cycle-arc.

- Espace des variables.
Comme $0 \leq \sum_{p \in \mathcal{P}_v} g_v^p \leq 1$, on a bien $0 \leq y_{ijv} \leq 1$ et $0 \leq \delta_v \leq 1$.
- Contraintes flot de marchandises.
Ces contraintes sont automatiquement vérifiées puisque dans les deux solutions les variables concernant les marchandises, x et z , sont identiques.

– Contraintes cycles de véhicules.

$$\begin{aligned}
 \sum_{j \in \delta_v^+(i)} y_{ijv} - \sum_{j \in \delta_v^-(i)} y_{jiv} &= \sum_{j \in \delta_v^+(i)} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p - \sum_{j \in \delta_v^-(i)} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p \\
 &= \sum_{p \in \mathcal{P}_v} g_v^p \left(\underbrace{\sum_{j \in \delta_v^+(i)} r_{ij}^p - \sum_{j \in \delta_v^-(i)} r_{ij}^p}_{=0} \right) \\
 &= 0.
 \end{aligned}$$

$$\begin{aligned}
 \sum_{(i,j) \in A : T_i \leq t < T_j} y_{ijv} &= \sum_{(i,j) \in A : T_i \leq t < T_j} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p \\
 &= \sum_{p \in \mathcal{P}_v} g_v^p \underbrace{\sum_{(i,j) \in A : T_i \leq t < T_j} r_{ij}^p}_{\leq 1} \\
 &\leq \sum_{p \in \mathcal{P}_v} g_v^p = \delta_v.
 \end{aligned}$$

– Contraintes de capacité faibles et fortes.

$$\begin{aligned}
 \sum_{k \in K} x_{ijv}^k - u_{ijv} y_{ijv} &= \sum_{k \in K} x_{ijv}^k - u_{ijv} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p \\
 &= \sum_{k \in K} x_{ijv}^k - \sum_{p \in \mathcal{P}_v} u_{ijv} g_v^p r_{ij}^p \\
 &\leq 0.
 \end{aligned}$$

On démontre de même que si (x, z, g) vérifie les contraintes de capacité fortes alors son image les vérifie également.

– Contraintes de quotas pour un véhicule utilisé.

$$\begin{aligned}
 \sum_{(ij) \in A; N_i \neq N_j} y_{ijv} t_{ijv} &= \sum_{(ij) \in A; N_i \neq N_j} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p t_{ijv} \\
 &= \sum_{p \in \mathcal{P}_v} \sum_{(ij) \in A; N_i \neq N_j} t_{ijv} r_{ij}^p g_v^p \\
 &\in [QuotaMin_v; QuotaMax_v].
 \end{aligned}$$

Il ne reste plus qu'à comparer la valeur des fonctions objectifs. Les coûts et profits relatifs aux marchandises sont identiques puisque dans les deux solutions les variables concernant les marchandises x et z sont identiques. On va donc comparer les coûts fixes

d'ouverture de services.

$$\begin{aligned}
 \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} y_{ijv} &= \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} \sum_{p \in \mathcal{P}_v} g_v^p r_{ij}^p \\
 &= \sum_{v \in V} \sum_{p \in \mathcal{P}_v} \underbrace{\left(\sum_{(ij) \in A} CF_{ijv} r_{ij}^p \right)}_{=CF_v^p} g_v^p \\
 &= \sum_{v \in V} \sum_{p \in \mathcal{P}_v} CF_v^p g_v^p.
 \end{aligned}$$

Les deux fonctions objectifs ont la même valeur.

On a démontré que \mathcal{T} définit bien une application affine $\mathcal{T} : \mathcal{P}_{caLP} \rightarrow \mathcal{P}_{aaLP}$, qui conserve la valeur objectif des solutions. D'une façon imagée, si les espaces de définition étaient les mêmes, le polyèdre \mathcal{P}_{caLP} serait inclus dans le polyèdre \mathcal{P}_{aaLP} . \square

Les modèles exacts arc-arc et cycle-arc d'une part, arc-chemin et cycle-chemin d'autre part, ayant la même valeur à l'optimum, ce théorème prouve uniquement que le gap entre une solution de la relaxation linéaire et une solution exacte est plus faible pour les modèles à base de cycle par rapport à leur équivalent à base d'arcs. Cependant, ce théorème ne donne aucune mesure de cette "domination". Les résultats expérimentaux confirment le fait que les formulations à base de morceaux de cycle sont meilleures, de plus ils permettent de mesurer les gaps respectifs des différents modèles. Par ailleurs, pour mettre en valeur l'intérêt d'introduire les contraintes de capacité fortes dans nos formulations, nous détaillons les résultats expérimentaux obtenus pour la version faible et la version forte de chacun des quatres modèles.

4.3 Comparaison expérimentale des modèles

Le but de notre étude expérimentale est de comparer la qualité des quatre formulations, à la fois en terme de qualité des relaxations linéaires et en temps de résolution. Nous donnons les résultats pour les versions faibles et fortes des formulations.

4.3.1 Instances

Nous avons développé un générateur d'instances en commun avec Nicolas Teypaz. Nous avons généré différents jeux d'instances sur 5 terminaux, avec une flotte fixe et homogène composée de 2, 3 ou 4 véhicules et une période de temps divisée en 4 intervalles. Les matrices de coûts fixes et variables sont données. Les matrices de demandes sont de quatre physionomies distinctes.

1. 10 demandes origine-destination aléatoirement réparties dans la matrice des demandes.
2. 10 demandes origine-destination regroupées autour d'un pôle d'attractivité (*hub*).
3. 15 demandes origine-destination aléatoirement réparties dans la matrice des demandes.

4. 20 demandes origine-destination qui remplissent entièrement la matrice des demandes.

La taille des demandes suit quatre répartitions possibles : demandes grandes, moyennes, petites et minuscules. Pour les demandes minuscules, toutes les demandes sont de taille inférieure à la capacité des véhicules alors que pour les petites demandes, quelques demandes peuvent dépasser cette capacité. Pour les moyennes (resp. grandes) demandes la majorité des demandes est de taille proche (resp. largement supérieure) de la capacité des véhicules. Si l'on considère une flotte homogène avec des véhicules de capacité 100, la taille des demandes suit les répartitions données dans le tableau 4.3.

w^k	1-30	30-90	90-150	150-250	250-800
Demandes Minuscules	50%	50%	0%	0%	0%
Demandes faibles	40%	40%	10%	5%	5%
Demandes moyennes	5%	20%	50%	20%	5%
Demandes fortes	5%	5%	10%	40%	40%

TABLE 4.3 – Répartition des demandes.

On obtient ainsi 48 profils d'instances (4 physiologies de demandes, 4 types de demandes, et 3 flottes de véhicules). Pour chacun de ces profils, nous avons généré 3 instances.

4.3.2 Résultats sur nos instances

Les résultats présentés sont des moyennes des résultats obtenus sur 3 instances par profil.

Les modèles ont été implémentés en Visual C++ et résolus avec ILOG Cplex 10.2 sur un ordinateur de bureau doté d'un processeur Intel Pentium 4 à 2.6GHz avec 1.5Go de RAM. Pour les modèles fondés sur des variables cycles pour les véhicules, tous les morceaux de cycles sont générés par un algorithme de parcours en profondeur, puis seuls les morceaux admissibles (respectant les quotas et le temps de réengagement) sont conservés et utilisés dans la construction du modèle. Cette génération est très rapide sur ces petites instances. De même, tous les chemins possible origine-destination pour le transport des marchandises sont générés par un algorithme de parcours en profondeur.

Le tableau 4.4 donne le temps de résolution moyen des versions faibles et fortes des quatre modèles. C'est le temps de résolution requis par Cplex pour obtenir une solution optimale entière. Le temps de calcul est limité à 1h (3 600 secondes). Le tableau 4.5 donne les valeurs du gap entre la relaxation linéaire et la solution exacte, ce gap est défini par la formule suivante :

$$gap = \frac{(\text{valeur RL} - \text{valeur exacte})}{\text{valeur exacte}}.$$

Ces gaps permettent de mesurer la qualité des relaxations linéaires des quatre modèles, et d'évaluer l'impact des contraintes de capacité fortes. Pour plus de lisibilité nous ne détaillons pas les résultats obtenus sur les 48 profils d'instances, mais présentons les moyennes des résultats obtenus en faisant des moyennes sur différentes caractéristiques : taille des demandes, nombre de véhicules et nombre de demandes.

	arc-arc faible temps	arc-arc fort temps	cycle-arc faible temps	cycle-arc fort temps	arc-che faible temps	arc-che fort temps	cycle-che faible temps	cycle-che fort temps
Moyenne	315	311	269	155	464	539	335	229
Gd	59	52	11	13	79	76	13	16
Mo	320	273	131	138	392	574	132	161
Pt	546	580	622	390	773	801	678	609
Min	336	337	309	78	611	706	516	128
v2	7	6	10	12	9	12	13	17
v3	96	110	152	111	203	288	216	126
v4	843	816	643	341	1180	1317	775	542
k10	273	268	213	110	354	419	64	31
k15	283	292	171	155	382	466	181	136
k20	524	556	477	256	766	850	570	457

TABLE 4.4 – Temps de résolution (en secondes).

D'une manière générale les résultats reportés dans ces tableaux mettent en avant la qualité des formulations à base de morceaux de cycles qui garantissent à la fois un meilleur gap (gain d'environ 4% par rapport aux modèles sur les arcs), et un temps de calcul plus court. Ceci valide donc expérimentalement la dominance des modèles à base de cycle sur les modèles à base d'arcs donnée dans le théorème 4.2.1 On voit également que le "gap" entre ces deux types de formulations est important.

Les résultats confirment également la mauvaise qualité de la borne obtenue par la relaxation linéaire du modèle arc-arc sans les contraintes de capacité forte, avec un gap de plus de 10% en moyenne. Ceci contribue à un temps de résolution plus long pour obtenir une solution entière à cause d'un grand nombre de nœuds à explorer dans l'arbre de Branch&Bound.

Les contraintes de capacité fortes améliorent elles aussi de près de 4% les relaxations linéaires pour les quatre modèles. Elles permettent d'accélérer nettement le temps de résolution pour le modèle cycle-arc (temps divisé par 2). Le gain pour le modèle cycle-chemin est encore conséquent (temps divisé par 3). Par contre elles n'améliorent pas le temps de résolution pour le modèle arc-arc : le nombre de nœuds explorés est plus petit, mais le temps passé à chaque nœud est plus important à cause du plus grand nombre de contraintes. Le gain lié aux contraintes de capacité fortes est d'autant plus important que les demandes sont de plus petite taille. Pire, le temps de résolution est augmenté pour le modèle arc-chemin.

Ces contraintes ont un net intérêt sur ces petites instances, cependant, leur nombre augmentant très rapidement avec la taille des instances, elles alourdissent trop les modèles sur de plus grandes instances et saturent vite la mémoire ralentissant l'exploration des nœuds de l'arbre de branchement. Une alternative est alors d'ajouter dynamiquement les contraintes de capacité fortes violées, à condition qu'il soit facile de les identifier.

Notons que le temps de résolution augmente considérablement lorsque le nombre de

4.3 Comparaison expérimentale des modèles

	arc-arc	arc-arc	cycle-arc	cycle-arc	arc-che	arc-che	cycle-che	cycle-che
Pb	faible	fort	faible	fort	faible	fort	faible	fort
	gap	gap	gap	gap	gap	gap	gap	gap
Moyenne	10,1	6,3	6,2	2,8	10,4	6,5	7,0	3,0
Gd	7,5	6,6	3,5	2,7	7,5	6,7	3,5	2,7
Mo	11,3	9,0	6,4	4,4	11,6	9,2	6,4	4,4
Pt	11,2	6,0	10,0	2,8	11,5	6,3	10,0	2,8
Min	10,5	3,3	8,3	2,3	11,0	3,8	8,3	2,3
v2	11,5	6,4	8,0	3,3	11,7	6,6	8,1	3,5
v3	10,2	6,6	6,2	3,1	10,4	6,8	6,3	3,3
v4	8,7	5,8	4,4	2,1	9,1	6,1	6,7	2,3
k10	8,8	5,4	5,6	2,6	9,2	5,5	7,0	2,7
k15	12,8	8,8	6,6	3,4	13	8,9	6,6	3,5
k20	9,6	5,3	7,0	2,7	10,2	5,9	7,5	3,2

TABLE 4.5 – Gap (en %) des différentes formulations.

véhicules augmente. Ceci est dû, d'une part à l'augmentation du nombre de variables binaires, et d'autre part à l'augmentation du nombre de symétries dans le problème. Ce temps augmente également, lorsque le nombre de demandes augmente. La formulation cycle-arc a plus de variables binaires, mais sa résolution est presque toujours plus rapide du fait du meilleur gap. Cette tendance s'inverse notamment pour les instances avec des demandes grandes et moyennes pour lesquelles le modèle cycle-arc n'apporte pas une grande amélioration du gap.

Parmi les instances testées, il apparaît dans le tableau 4.4 que ce sont les instances avec des petites demandes qui nécessitent le plus de temps de calcul. Cela est lié à la difficulté des problèmes qui contiennent un mélange d'objets de petite et de grande taille, comme pour les problèmes de *packing* ou de *cutting*, voir [Cung *et al.*, 2000].

Nos tests nous ont également permis d'obtenir les résultats suivants. Le temps de résolution de la relaxation linéaire, de quelques secondes pour le modèle faible, est nettement augmenté pour le modèle fort (multiplié en moyenne par 6) du fait du très grand nombre de contraintes de capacité fortes. Ceci n'a pas d'impact sur ces petites instances, mais peut devenir gênant pour des instances de plus grande taille. De plus, la taille en mémoire des modèles augmente considérablement en ajoutant les contraintes fortes, ce qui limite d'autant plus la taille des instances pour lesquelles on peut obtenir la valeur de la relaxation linéaire.

La comparaison de la moyenne des résultats sur 6 profils avec hub et 6 profils sans hub, indique un gain en terme de temps de résolution (10% plus rapide) pour des demandes structurées autour de hub, aussi bien pour la relaxation linéaire que pour la résolution exacte.

Chapitre 5

Résolution par génération de colonnes

Sommaire

5.1	La génération de colonnes	88
5.1.1	Principe pour résoudre un PL	88
5.1.2	Principe pour résoudre un PLNE	89
5.2	Algorithme de génération de colonnes	90
5.2.1	Problème maître restreint	91
5.2.2	Sous-problème de génération de cycles pour les véhicules	92
5.2.3	Sous-problème de génération de chemins pour les marchandises	93
5.3	Résultats numériques	93
5.4	Résolution d'instances industrielles	96
5.4.1	Instances de type transport express	97
5.4.2	Instances de type transport lent	97

Ce chapitre a pour objectif d'établir une première base algorithmique pour la résolution efficace du problème de transport de fret avec flotte fixée sur de grandes instances. Vraisemblablement, les techniques à favoriser pour résoudre de grandes instances sont à base de génération de colonnes. Nous allons donc esquisser un algorithme simple de génération de colonnes, et proposer une étude comparative entre différentes stratégies. Nous verrons sur quels paramètres on peut jouer pour garder un bon compromis entre temps de calcul, qualité de la solution et taille d'instance.

Dans la section 5.1, nous rappelons brièvement ce qu'est la génération de colonnes et son fonctionnement. Dans la section 5.2, nous explicitons la génération de colonnes adaptée à notre problème (écriture des problèmes maître et esclave). Finalement, dans la section 5.3 nous présentons les premiers résultats numériques.

5.1 La génération de colonnes

La génération de colonnes est une technique sophistiquée de résolution exacte de problèmes linéaires de grande taille [Lubbecke et Desrosiers, 2005], comportant un nombre exponentiel de variables. Ces formulations sont en général obtenues via des décompositions de Dantzig et Wolfe [Dantzig et Wolfe, 1960]. Traditionnellement, on rappelle que la première publication sur la génération de colonnes [Gilmore et Gomory, 1963] est sur le problème de cutting stock.

L'idée centrale est que les programmes linéaires de grande taille ont trop de variables (ou colonnes) pour qu'on puisse les représenter toutes de manière explicite. A l'optimum, la plupart des variables sont hors base et, très souvent, la plupart d'entre elles sont nulles (les autres à leur borne sup), c'est-à-dire que seul un (petit) sous-ensemble de variables doit être pris en compte pour résoudre le problème. Reste donc à déterminer ces bonnes colonnes.

La génération de colonnes permet ainsi de résoudre des programmes linéaires (PL) (section 5.1.1). On peut également l'adapter et l'insérer dans un algorithme de branchement pour résoudre des programmes linéaires en nombres entiers (PLNE) (section 5.1.2).

5.1.1 Principe pour résoudre un PL

On souhaite résoudre un programme linéaire qui comporte un grand nombre de variables/colonnes : c'est le problème dit problème maître (MP). En général, on considère une formulation avec des variables complexes : pattern, chemins, ... selon les problèmes traités, obtenue à partir d'une décomposition de Dantzig-Wolfe. On suppose que ce problème consiste à maximiser une certaine fonction objectif.

On commence par choisir un sous-ensemble restreint de variables pour construire un problème maître restreint (RMP) de préférence facile à résoudre et qui admet une solution réalisable. La résolution de ce problème maître restreint permet d'obtenir les premières valeurs des variables duales associées aux contraintes de ce problème maître. On résout alors un sous-problème, appelé problème esclave (SP), qui vise à chercher des nouvelles colonnes de coût réduit positif (problème de maximisation), ce coût réduit est calculé en utilisant les valeurs des variables duales obtenues à l'étape précédente. Il s'agit de générer des nouvelles variables (nouveaux patterns, nouveaux chemins, ...) permettant d'améliorer la solution courante. Ce sous-problème est en général un problème difficile équivalent à un problème de plus court chemin contraint. Ces nouvelles colonnes sont alors ajoutées au problème maître restreint, la fonction objectif et les contraintes sont mises à jour en conséquence.

On itère alors ce processus, jusqu'à ce qu'il n'existe plus de colonne améliorante, i.e. qu'il n'existe pas de nouvelle colonne de coût réduit positif. La génération de colonnes est alors terminée, et la solution du problème maître restreint est optimale. C'est la solution optimale du problème maître complet. Le processus de génération de colonnes est détaillé dans la figure 5.1.

En pratique, la génération de colonnes peut s'avérer très performante [Lubbecke et Desrosiers, 2005], mais elle n'offre pas de garantie ni sur le nombre de colonnes à ajouter,

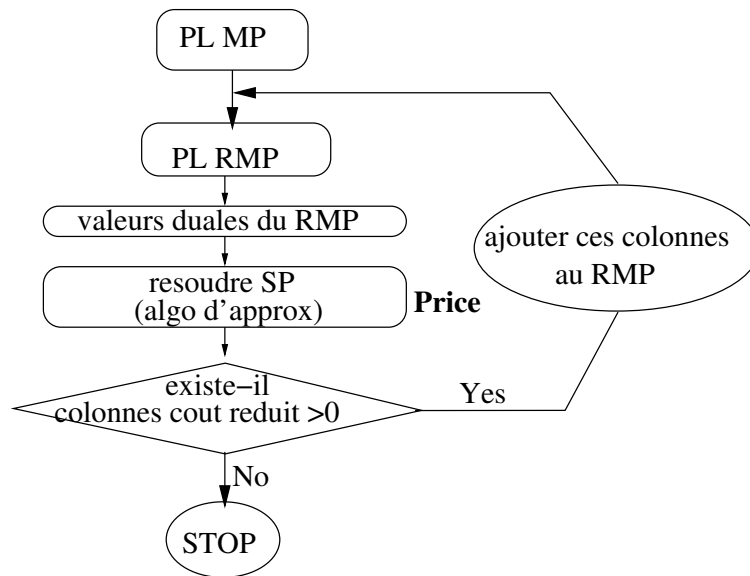


FIGURE 5.1 – Algorithme de génération de colonnes pour résoudre un programme linéaire de maximisation.

ni sur la rapidité de résolution. Des problèmes de convergences sont couramment rencontrés, notamment dus au fait qu'en début du processus de génération de colonnes, le problème maître comporte peu de colonnes et son problème dual est peu contraint, et les variables duales sont très instables. Ces problèmes de stabilité et de terminaison sont couramment rencontrés en pratique.

La génération de colonnes est source de nombreuses heuristiques pour la résolution de problèmes de très grande taille. On peut très simplement, limiter le nombre de colonnes générées, arrêter la génération si les x dernières nouvelles colonnes n'ont pas amélioré significativement la solution courante, ...

5.1.2 Principe pour résoudre un PLNE

La génération de colonnes ne permet que de résoudre des programmes linéaires. Pour résoudre un PLNE, il faut, à chaque nœud de l'arborescence de Branch&Bound, résoudre le PL correspondant. Dans un tel nœud, on a fixé certaines variables à des valeurs entières, les autres variables libres sont réelles. La figure 5.2 donne le schéma général de résolution par l'algorithme correspondant appelé Branch&Price : à chaque étape, soit on fixe une variable à une valeur entière (et on explore alors les différentes branches de l'arborescence de branchement), soit on génère des nouvelles colonnes. La solution optimale est obtenue lorsque la solution de la relaxation linéaire du problème maître restreint avec un certain nombre de variables fixées, est entière.

Afin d'accélérer le processus de branchement, il faut conjointement trouver de bonnes bornes (bornes inférieures pour un problème de maximisation) de la valeur objective op-

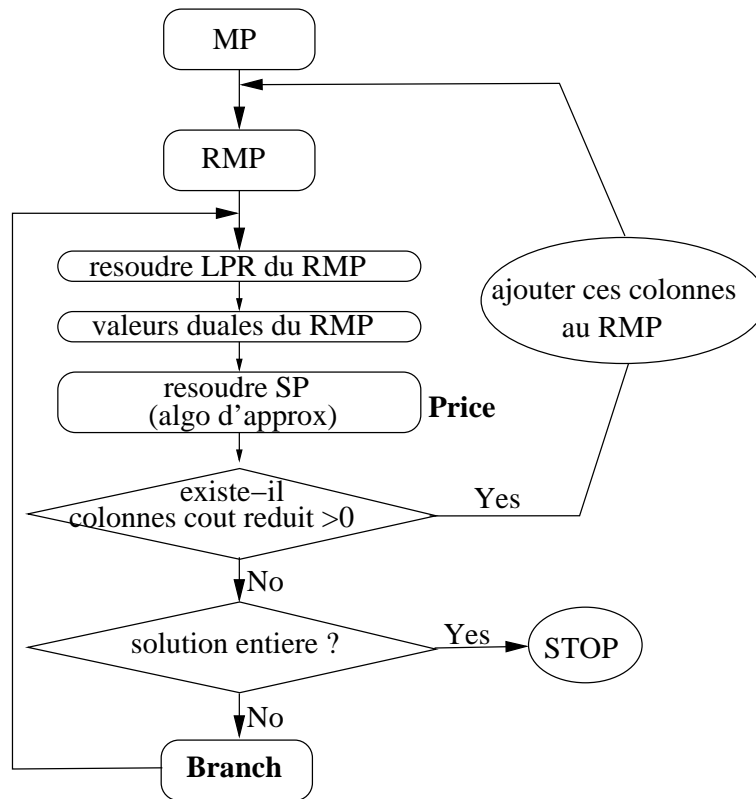


FIGURE 5.2 – Algorithme de Branch&Price pour résoudre un PLNE par génération de colonnes.

timale. Cela permet d'éviter l'exploration de branches entières, et limite grandement le nombre de nœuds à considérer. Ces bornes peuvent notamment être obtenues par des heuristiques. On retrouve dans la littérature de nombreuses heuristiques fondées sur des algorithmes de Branch&Price.

5.2 Algorithme de génération de colonnes

Dans cette section, nous nous intéressons à la génération de colonnes appliquée à notre problème de conception de réseau de service pour le transport de fret. Nous présentons ici les modèles maître et esclave pour la génération de colonnes à la fois pour les variables cycles pour les véhicules et pour les variables chemins pour les marchandises. Nous nous focalisons uniquement sur la génération de colonnes pour la formulation cycle-chemin car on peut facilement en extraire les modèles pour les formulations cycle-arc et arc-chemin.

5.2.1 Problème maître restreint

Le problème maître restreint est la relaxation linéaire du modèle en programmation linéaire mixte avec variables cycles pour les véhicules et variables chemins pour les marchandises. Le problème maître restreint ne considère qu'un nombre restreint de variables cycle et chemin : on travaille sur un sous-ensemble de variables.

Problème maître cycle-chemin

$$\begin{aligned} \text{Maximiser : } & + \sum_{k \in K} P^k \cdot z^k - \sum_{v \in V} \sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} CV^l \cdot h_v^{kl} \\ & - \sum_{v \in V} \sum_{p \in \mathcal{P}_v} CF_v^p \cdot g_v^p, \end{aligned} \quad (5.1)$$

sous contraintes

$$\sum_{v \in V} \sum_{l \in \mathcal{L}_v^k} h_v^{kl} = z^k, \quad \forall k \in K, \quad \rightarrow \sigma^k \quad (5.2)$$

$$\sum_{k \in K} \sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - \sum_{p \in \mathcal{P}_v} u_{ijv} r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, \forall v \in V, \quad \rightarrow \eta_{ijv} \quad (5.3)$$

$$\sum_{l \in \mathcal{L}_v^k} a_{ij}^{kl} h_v^{kl} - \sum_{p \in \mathcal{P}_v} b_{ijv}^k r_{ij}^p g_v^p \leq 0, \quad \forall (ij) \in A, \forall k \in K, \forall v \in V, \quad \rightarrow \rho_{ijv}^k \quad (5.4)$$

$$\sum_{p \in \mathcal{P}_v} g_v^p \leq 1, \quad \forall type, \quad \rightarrow \alpha_{type} \quad (5.5)$$

$$\begin{aligned} \sum_{v \in V_{type}} \sum_{j \in \delta^+(i)} g_v^p r_{ij}^p - \sum_{v \in V_{type}} \sum_{j \in \delta^-(i')} g_v^p r_{i'j}^p = 0, \\ \forall i \in N | T_i = 1 \ i' = (N_i, T), \forall type, \quad \rightarrow \theta_{itype} \end{aligned} \quad (5.6)$$

$$h_v^{kl} \geq 0 \quad \forall k \in K, \forall l \in \mathcal{L}_v^k, \forall v \in V, \quad (5.7)$$

$$z^k \in [0..w^k], \quad \forall k \in K, \quad (5.8)$$

$$g_v^p \in [0..1], \quad \forall p \in \mathcal{P}_v. \quad (5.9)$$

La contrainte de quotas qui assure que le temps d'utilisation du véhicule v est comprise entre deux bornes ($QuotaMin_v$ et $QuotaMax_v$) est directement incluse dans la génération des cycles.

On associe les variables duales α_{type} , σ^k , η_{ijv} , ρ_{ijv}^k et θ_{itype} aux contraintes (5.2) - (5.6). Après résolution du problème maître restreint, les variables duales sont transférées pour générer des morceaux de cycle et des chemins de coûts réduits positifs. Pour la génération de chemins pour les marchandises, il y a un sous-problème par couple marchandise-véhicule. Pour la génération de morceau de cycle, il y a un sous-problème par type de véhicule. Lorsque l'on génère simultanément cycles et chemins on peut adopter différentes stratégies quant aux *pricing* comme par exemple ne générer des cycles qu'une fois sur dix et sinon que des chemins.

5.2.2 Sous-problème de génération de cycles pour les véhicules

On cherche des cycles de coût réduit positif, pour les intégrer dans le problème maître restreint, afin d'en améliorer l'objectif. Le sous-problème détermine un morceau de cycle de coût réduit maximal. Si celui-ci est positif, ce morceau de cycle est ajouté au maître. Il faut résoudre un sous-problème par véhicule v .

Nous présentons deux modèles. Le premier se limite à générer des véritables cycles sur une période, en fin de période le cycle retourne à son terminal de départ. Dans le second, on envisage la rotation des véhicules sur plusieurs périodes.

Problème esclave de génération de cycles sur une seule période

$$\text{Maximiser : } \alpha_v - \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} y_{ijv} - \sum_{v \in V} \sum_{(ij) \in A} u_{ijv} \eta_{ijv} y_{ijv}, \quad (5.10)$$

sous contraintes

$$\sum_{(i,j) \in A: T_i \leq t < T_j} y_{ijv} \leq 1, \quad \forall t \in T, \quad (5.11)$$

$$\sum_{j \in \delta^+(i)} y_{ijv} - \sum_{j \in \delta^-(i)} y_{jiv} = 0, \quad \forall i \in N | T_i \neq 1, T_i \neq T, \quad (5.12)$$

$$\sum_{j \in \delta^+(i)} y_{ijv} - \sum_{j \in \delta^-(i')} y_{ji'v} = 0, \quad \forall i \in N | T_i = 1, i' = (N_i, T), \quad (5.13)$$

$$QuotaMin_v \leq \sum_{(ij) \in A; N_i \neq N_j} y_{ijv} t_{ijv} \leq QuotaMax_v, \quad (5.14)$$

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A. \quad (5.15)$$

La fonction objectif (5.10) correspond au coût réduit d'une colonne pour le véhicule v . Les contraintes permettent de définir un cycle pour le véhicule v . On a introduit la contrainte (5.13) pour forcer que ce cycle soit de longueur une période, de ce fait on n'a pas besoin des contraintes de cyclage dans le problème maître, ni la composante correspondante dans le calcul du coût réduit. Le cycle vérifie les contraintes de quotas (5.14).

Problème esclave de génération cycles avec rotation des véhicules

On a toujours un esclave par véhicule v , $type$ est le type du véhicule v . Le but est de trouver des morceaux de cycles qui respectent les contraintes des véhicules, et qui améliorent la solution courante (coût réduit positif).

$$\begin{aligned} \text{Maximiser : } & \alpha_v - \sum_{v \in V} \sum_{(ij) \in A} CF_{ijv} y_{ijv} - \sum_{v \in V} \sum_{(ij) \in A} u_{ijv} \eta_{ijv} y_{ijv} \\ & + \sum_{i \in N | T_i = 1} \sum_{j \in \delta^+(i)} \theta_{i,type} y_{ijv} - \sum_{i \in N | T_i = T} \sum_{j \in \delta^-(i)} \theta_{i,type} y_{jiv}, \end{aligned} \quad (5.16)$$

sous contraintes

$$\sum_{(i,j) \in A: T_i \leq t < T_j} y_{ijv} \leq 1, \quad \forall t \in T, \quad (5.17)$$

$$\sum_{j \in \delta^+(i)} y_{ijv} - \sum_{j \in \delta^-(i)} y_{jiv} = 0, \quad \forall i \in N | T_i \neq 1, T_i \neq T, \quad (5.18)$$

$$QuotaMin_v \leq \sum_{(ij) \in A; N_i \neq N_j} y_{ijv} t_{ijv} \leq QuotaMax_v, \quad (5.19)$$

$$y_{ijv} \in \{0, 1\}, \quad \forall (ij) \in A. \quad (5.20)$$

Les deux derniers termes de la fonction objectif (5.16) correspondent à la composante du coût réduit induit par les variables duales (θ) des contraintes de cyclage des véhicules. Les contraintes (5.17)-(5.19) permettent de définir un morceau de cycle pour le véhicule v .

5.2.3 Sous-problème de génération de chemins pour les marchandises

Pour chaque marchandise k , on a un sous-problème de génération de chemin. Le but est de trouver un chemin de coût réduit positif. On note a_{ijv}^k la variable binaire qui vaut 1 si la marchandise k transite à bord du véhicule v sur le tronçon (ij) .

$$\text{Maximiser : } -\sigma^k - \sum_{v \in V} \sum_{(ij) \in A} (CV_{ijv}^k + \eta_{ijv} + \rho_{ijv}^k) a_{ijv}^k, \quad (5.21)$$

sous contraintes

$$\sum_{j \in \delta^+(i)} \sum_{v \in V} a_{ijv}^k - \sum_{v \in V} \sum_{j \in \delta^-(i)} a_{jiv}^k = \begin{cases} +1 & \text{si } i = o^k, \\ -1 & \text{si } i = d^k, \\ 0 & \text{sinon,} \end{cases} \quad \forall i \in N, \quad (5.22)$$

$$a_{ijv}^k \in \{0, 1\}, \quad \forall (ij) \in A, \forall v \in V, \quad (5.23)$$

Ce programme linéaire en nombre entiers permet de trouver des chemins pour le transport d'une marchandise k depuis son origine jusqu'à sa destination. Après résolution, le chemin trouvé $l \in \mathcal{L}_v^k$ est défini par $l = \{(ij) | a_{ijv}^k = 1\}$. Son coût réduit est donné par la fonction objectif.

5.3 Résultats numériques

Nous avons implémenté différents algorithmes de génération de colonnes pour résoudre la relaxation linéaire des modèles faibles de notre problème de conception de réseau de service avec gestion des véhicules. A chaque étapes de la génération de colonnes, les sous problèmes, problèmes esclaves sont résolus directement par PLNE avec Cplex. Une fois la relaxation linéaire résolue, nous résolvons directement avec Cplex 10.2 le PLNE initial avec les colonnes générées à la racine. Nous n'avons donc pas implémenté d'algorithme de Branch&Price. Nous avons lancé des tests sur les modèles (en version faible) suivants :

- modèle cycle-arc avec cycle sur une période
- modèle cycle-arc
- modèle arc-chemin
- modèle cycle-chemin

Dans ces tests on résout les mêmes petites instances que dans le chapitre précédent. Le but est d'avoir une première idée sur la performance d'un algorithme de génération de colonnes, et de mesurer la perte en terme de profit de certaines limitations et le gain en terme de temps de calcul. Que perd-on si on ne fait pas un Branch&Price, mais que l'on génère uniquement des colonnes à la racine ? Quelle perte si la rotation des véhicules est limitée à une rotation sur une période ?

On mesure également le nombre de colonnes générées. Les colonnes sont ajoutées au pool de colonnes, mais à chaque itération on recharge complètement le modèle, l'ajout n'est pas du tout optimisé.

Profil	Tps ca GenCol	Tps ca GenCol RL	Tps ca GenCol cycles1p	Tps ca faible exact	Obj ca GenCol perte	Obj ca GenCol cycles1p	Nb Col GenCol	Nb Col GenCol cycles1p
moyenne	11,2	2,9	3,9	268,5	2,4%	8,5%	19,0	10,5
gd	3,9	2,6	1,7	11,2	0,6%	8,3%	17,0	7,8
mo	4,9	2,2	8,3	131,2	1,5%	7,8%	13,1	9,6
pt	28,0	3,8	3,6	622,3	5,7%	8,0%	26,2	15,1
min	7,8	2,8	2,2	309,4	2,5%	11,4%	19,7	9,3
v2	2,8	2,0	1,6	10,3	2,6%	6,5%	19,5	11,1
v3	6,2	2,9	2,7	151,8	1,3%	7,0%	19,0	10,3
v4	24,5	3,7	7,5	643,45	3,3%	11,4%	18,5	10,0
k10	4,2	2,5	2,2	213,9	3,7%	9,8%	20,2	11,5
k15	8,3	2,1	7,7	170,9	1,2%	9,1%	13,2	8,4
k20	27,8	4,3	7,5	477,4	1,5%	6,4%	22,2	10,5

TABLE 5.1 – Temps de résolution (en secondes) et gap entre relaxation linéaire et solution exacte en %.

Dans le tableau 5.1, on donne les résultats sur les modèles cycle-arc faibles. La première colonne donne le temps de résolution du modèle cycle-arc faible exact, avec génération de colonnes uniquement pour la résolution de la relaxation linéaire, dont le temps est donné en deuxième colonne, puis résolution directe du PLNE sur les colonnes générées. La troisième colonne donne le temps de résolution du modèle cycle-arc avec génération de cycles, lorsqu'on se limite à des morceaux de cycles qui sont de véritables cycles sur une période. Dans la colonne suivante on retrouve les résultats de la résolution exacte du modèle cycle-arc avec génération a priori de tous les morceaux de cycles. Les deux colonnes suivantes donnent la perte relative en terme de valeur objectif des heuristiques à base de génération de colonnes. Les deux dernières colonnes précisent le nombre moyen de colonnes générées.

5.3 Résultats numériques

L'algorithme heuristique à base de génération de colonne est en moyenne 24 fois plus rapide, on ne génère qu'une vingtaine de colonnes. Si on limite les colonnes aux cycles sur une seule période, le temps de calcul est encore divisé par 3 avec seulement la génération d'une dizaine de colonnes. Ce gain en temps est plus ou moins net selon la taille des demandes.

Comme on ne fait pas un Branch&Price complet, mais que les colonnes sont uniquement générées à la racine, on a une perte en objectif, cette perte est de 2,4% en moyenne. Si l'on se restreint aux cycles sur une seule période (pas de rotation entre les véhicules), la perte en objectif est en moyenne de 8,5%, et peut atteindre jusqu'à 20% sur certaines instances, la rotation de véhicules est donc une composante essentielle dans la maximisation du profit.

Le tableau 5.2 donne les résultats pour l'heuristique de génération de colonne sur le modèle arc-chemin.

Profil	Tps	Tps	Obj	Nb Col
	ap GenCol	ap faible exact	ap GenCol	ap GenCol
moyenne	120,8	463,9	3,2%	249
gd	30,3	79,2	0,34%	384
mo	243,8	392,4	2,1%	280
pt	159,3	773,3	5,9%	159
min	79,71	610,5	7,4%	172
v2	4,0	9,0	2,8%	176
v3	17,2	202,5	1,98%	246
v4	341,2	1180	4,7%	324
k10	110,8	353,5	3,5%	192
k15	143,0	382,0	2,6%	323
k20	138,6	766,3	3,6%	290

TABLE 5.2 – Temps de résolution (en secondes) et gap entre relaxation linéaire et solution exacte en %.

L'algorithme heuristique à base de génération de colonne est en moyenne 4 fois plus rapide, on génère près de 250 colonnes. Comme on ne fait pas un Branch&Price complet, mais que les colonnes sont uniquement générées à la racine, on a une perte en objectif, cette perte est d'environ 3% en moyenne.

Le tableau 5.3 donne les résultats pour l'heuristique de génération de colonne sur le modèle cycle-chemin.

Profil	Tps	Tps	Obj	Nb Col
	cp	cp faible	cp	cp
	GenCol	exact	GenCol	GenCol
moyenne	6,95	334,7	3,3%	273
gd	5,47	13,2	0,36%	320
mo	5,82	132,4	2,2%	280
pt	8,9	677,6	6,1%	270
min	7,6	515,8	8,1%	225
v2	4,3	12,9	5,2%	203
v3	6,5	216,2	2,9%	275
v4	10,1	775	2,4%	343
k10	5,5	294,2	2,9	253
k15	6,3	290,1	4,5	290
k20	10,6	460,5	3,3	334

TABLE 5.3 – Temps de résolution (en secondes) et gap entre relaxation linéaire et solution exacte en %.

L'algorithme à base de génération de colonnes est très rapide, la perte en terme de fonction objectif de notre heuristique est de 3,3% en moyenne, mais peut atteindre plus de 8% lorsque les demandes sont toutes de très petite taille.

Parmi les différentes heuristiques à base de génération de colonnes, celles fondées sur des formulations cycle sont à retenir : le temps de résolution est faible et la qualité des solutions obtenue est bonne. Le modèle arc-chemin est peu intéressant. En revanche le modèle cycle-chemin est très prometteur pour les instances de plus grande taille, par contre il n'intègre pas le transbordement. On retiendra également, que le cyclage des véhicules est essentiel pour garantir un bon profit.

5.4 Résolution d'instances industrielles

La partenaire industriel pour lequel le problème de conception de réseau de service avait été résolu, souhaitait résoudre deux types d'instances. Des instances de type transport express avec des matrices de demandes pleines sur un réseau comportant peu de terminaux. Une instance typique de transport express comporte 7 terminaux, 40 demandes et se situe sur une période de temps de 1 jour. Les autres instances sont des instances sur un grand réseau comportant entre une vingtaine et une quarantaine de terminaux, avec des matrices de demandes très creuses (taux de remplissage entre 10% et 20%), sur une période de temps de quelques jours. Par opposition aux instances express, ces instances sont de type transport lent. Nous avons donc menés des tests supplémentaires, pour savoir si l'utilisation de nos heuristiques est prometteuse sur ces deux types d'instances.

5.4.1 Instances de type transport express

Le tableau 5.4 donne les résultats des heuristiques de génération de colonnes sur les modèles cycle-arc et cycle-chemin sur des instances de type transport express. On retrouve le temps de résolution pour l'heuristique de génération de toutes les colonnes à la racine puis résolution directe par Cplex limitée à 1 heure sur les deux modèles. On donne également le nombre de colonnes générées ainsi que le gap de Cplex.

Profil	Tps	Tps	Tps	Tps	Gap	Nb	Gap	Nb
	cp	cp	ca	ca	Cplex	Col	Cplex	Col
	GenColRL	GenCol	GenColRL	GenCol	cp	cp	ca	ca
n7 v3 k20 t4	55	58	206	664	0	1 126	0	245
n7 v3 k20 t8	2 489	3 744	22 263	27 454	2.2%	2 130	6.6%	1 661
n7 v3 k40 t4	87	98	949	1 020	0%	1 324	0	321
n7 v3 k40 t8	641	1 329	-	-	0	3 373	-	-
n7 v6 k20 t4	160	2 966	374	4 081	0	1 739	3.9%	232
n7 v6 k20 t8	4 635	9 326	7 846	12 654	1.4%	2 836	2.3%	615
n7 v6 k40 t4	198	1 829	681	4 218	4.5%	1 926	3.5%	258
n7 v6 k40 t8	3 616	7 898	-	-	6.2%	3 254	-	-

TABLE 5.4 – Temps de résolution (en secondes) et gap de Cplex pour les heuristiques à base de génération de colonnes.

Le nombre d'intervalles de temps, i.e. la discrétisation du temps a une très forte influence sur le temps de résolution. Ainsi la résolution de la relaxation linéaire du modèle cycle-arc avec plus de 8 périodes de temps n'est pas réalisée en moins de quelques heures (ce qui correspond entre autre aux cases vides). Le modèle cycle-chemin est très prometteur, une optimisation du code et davantage de finesse dans le processus de génération de colonnes permettraient sans doute de résoudre très efficacement des instances industrielles de type transport express.

5.4.2 Instances de type transport lent

Pour évaluer les limites des heuristiques de génération de colonnes sur les modèles à base de cycles nous avons mené quelques tests complémentaires sur des plus grandes instances de type transport lent. Nous avons généré des instances de taille croissante ; ces instances comportent entre 14 et 42 terminaux, une flotte de 8 à 16 véhicules, entre 80 et 240 demandes, et une période de temps de 2 jours discrétisés en 16 intervalles ou de 4 jours discrétisés en 24 intervalles.

Pour résoudre ces instances à l'aide de MIP, l'essentiel est de trouver le bon compromis entre nombres de contraintes et nombres de variables. Afin de limiter au maximum le nombre de contraintes nous choisissons la modélisation cycle-chemin (version faible). Nous avons implémenté l'heuristique grossière qui consiste à stopper la génération de colonnes après un nombre fixe de colonnes générées sur le modèle cycle-chemin. Ceci limite donc

le nombre de variables considérées. Dans nos tests nous avons limité ce nombre à 5000 colonnes. Nous avons résolu la relaxation linéaire du modèle cycle-chemin puis le modèle exact avec ces colonnes. Le tableau 5.5 donne les résultats qualitatifs.

n14 v8 k80 t16	solution de bonne qualité
n21 v10 k120 t16	solution acceptable
n28 v12 k160 t24	quelques instances résolues
n35 v14 k200 t24	pas de solution
n42 v16 k240 t24	problème de mémoire saturée

TABLE 5.5 – Generation de colonnes limitée à 5000 colonnes sur des grandes instances.

Notre heuristique permet de résoudre les plus petites instances de type transport lent en un temps raisonnable. Cependant les solutions obtenues ne sont pas de très bonne qualité. Sur les plus grandes instances qui comportent plus de 35 terminaux, on ne parvient pas à obtenir de solution réalisable et le modèle sature rapidement la mémoire.

Pour parvenir à résoudre ces plus grandes instances, une des heuristiques envisageables serait de résoudre, dans un premier temps, le problème sans la dimension temporelle puis de réinsérer le temps. Une exemple d'une telle heuristique est donnée dans [Teypez, 2008], il faudrait l'adapter au cas de la flotte fixée.

Synthèse

Dans cette partie, nous avons étudié un problème de conception de réseau avec gestion de flotte. Nous avons d'abord comparé quatre formulations MIP du problème, deux avec transbordement, deux sans. Pour deux d'entre-elles, les variables de conception sont exprimées sur les arcs, alors que, pour deux autres, elles sont exprimées sur des chemins (morceaux de cycles) potentiels des véhicules. Notre étude expérimentale sur de petites instances montre que les formulations à base de cycles offrent non seulement des bien meilleure relaxation linéaire, mais que leur temps de calcul est également réduit. Ce résultat est confirmé par une étude théorique de comparaison des relaxations linéaires des modèles.

Malheureusement, la taille des problèmes résolus dans cette étude se limite à 5 terminaux, 20 demandes, 4 véhicules et une discrétisation en 4 intervalles de temps. Pour des instances de taille plus grande, la génération exhaustive de tous les chemins potentiels engendre très rapidement des modèles qui ne tiennent plus en mémoire. Ces modèles ne peuvent donc pas être résolus directement pour les instances réelles qui comportent des dizaines de terminaux, des centaines de demandes, une dizaine de véhicules et surtout une discrétisation de la période en 10 à 50 intervalles de temps.

Afin de proposer des solutions à notre problème industriel avec flotte fixée sur de telles instances, un algorithme de génération de colonnes semble prometteur. Nous avons déjà proposé une esquisse d'un tel algorithme, et évalué la qualité de certaines heuristiques envisageables. Cependant pour résoudre des instances de taille réelles, des contraintes additionnelles comme par exemple des fenêtres de temps permettant de limiter la combinatoire, ainsi qu'un pas de discrétisation du temps bien choisi seront sans doute nécessaires.

Troisième partie

Gestion de perturbation dans le milieu aérien

Dans cette partie nous présentons le travail réalisé pour le challenge ROADEF'2009, challenge auquel j'ai participé avec trois autres doctorants et ex-doctorants du laboratoire G-SCOP entre février 2008 et février 2009.

Le sujet était proposé par la société Amadeus¹ et concerne la gestion de perturbations dans le domaine aérien. Le but du challenge est de proposer et d'implémenter en quelques mois un algorithme qui parvient à calculer les meilleures solutions possibles sur des instances fournies par l'industriel.

Les caractéristiques du problème de gestion de perturbation sont détaillées dans le chapitre 6, notre approche de résolution est présentée dans le chapitre 7. Finalement, le chapitre 8 donne les résultats numériques obtenus. Pour faciliter la lecture, l'ensemble des notations et paramètres du problèmes sont récapitulés dans l'annexe B.

1. <http://www.amadeus.com/>

Chapitre 6

Gestion de perturbation dans le milieu aérien

Sommaire

6.1	Description du problème	104
6.1.1	Données	104
6.1.2	Contraintes	110
6.1.3	Fonction d'optimisation	112
6.1.4	Structure d'une solution	113
6.2	Etat de l'art	116
6.2.1	Planification dans le domaine aérien : processus et modèles	116
6.2.2	Recouvrement d'une seule ressource	120
6.2.3	Recouvrement intégrant plusieurs ressources	121
6.3	Les méthodes proposées par les finalistes	122

Les compagnies aériennes commerciales opèrent leurs vols selon un programme de vols publié, optimisé du point de vue des revenus engendrés. Ce programme de vols est, en général, calculé plusieurs mois à l'avance grâce à des logiciels puissants fondés sur des méthodes et outils d'optimisation, notamment de Recherche Opérationnelle. Cependant, des événements extérieurs, tels des pannes mécaniques, une grève du personnel, des conditions météorologiques défavorables, des problèmes techniques ou opérationnels, perturbent régulièrement le bon déroulement de ce programme. Les compagnies aériennes doivent alors trouver des solutions performantes qui minimisent l'impact des perturbations c'est-à-dire qui minimisent le surcoût pour la compagnie aérienne et qui permettent de revenir rapidement au programme de vol initialement prévu. Cette replanification très complexe devant être faite en seulement quelques minutes, les logiciels actuels ne produisent pas encore de solutions satisfaisantes gérant toutes, ou partie des ressources (appareils, équipages, passagers, opérations au sol (gestion des bagages, service de restauration), allocation des *slots* (créneau horaire dédié et alloué pour un aéronef)).

De façon traditionnelle, comme lors de leur première affectation, les ressources sont réaffectées au cours d'un processus séquentiel selon l'ordre hiérarchique : flotte d'appareils,

équipages, opérations au sol, et passagers. Un bref aperçu de ce contexte est donné dans la section 6.2.1. Les problèmes simplifiés de réaffectation concernant une seule ressource ont été largement traités dans la littérature, un certain nombre de travaux sont présentés dans l'état de l'art section 6.2.2. Une réaffectation des appareils de la flotte, qui modifie le programme de vols initial, risque de conduire à une impossibilité de correspondance pour un équipage ou un ensemble de passagers. De ce fait, une meilleure optimisation de la réaffectation des ressources ne peut se faire uniquement avec des décisions au niveau local et ne concernant qu'une seule ressource. Il est nécessaire d'intégrer les différents niveaux de décision. Les travaux dans ce contexte sont en pleine croissance, une étude bibliographique en donne un aperçu dans la section 6.2.3. Le sujet du challenge s'inscrit dans ce courant. Sa spécificité est de chercher, en cas de perturbations, à réaffecter de façon simultanée la flotte d'appareils et les passagers.

6.1 Description du problème

Cette section a pour objet la description des principales caractéristiques du problème. Pour plus de détails concernant les données, les coûts, les contraintes et les instances mises à disposition le lecteur est invité à lire le sujet complet du challenge [Palpant *et al.*, 2008]. Le but du challenge est d'absorber le plus rapidement possible les perturbations affectant un programme de vols. Le programme de vols initialement prévu, doit pouvoir être réappliqué, sans aucune modification supplémentaire, après une période de temps donnée, appelée *période de recouvrement*. Ainsi, pour absorber la perturbation, il faut générer un nouveau programme de vols pour la période définie par la période de recouvrement. Il s'agit donc de déterminer la rotation de chacun des appareils de la flotte et la réacommodation des passagers sur les vols opérés. Le temps de calcul pour obtenir une telle solution est limité à 10 minutes sur un ordinateur de bureau standard.

La description du problème ne se veut pas trop technique, l'essentiel étant de se familiariser avec le problème et de bien noter les contraintes prises en compte pour ce challenge. Nous ne décrivons donc pas ici la syntaxe complexe et variée de tous les fichiers d'entrée. Cependant, un bon nombre d'information est nécessaire soit à la description même d'une instance, soit au calcul du coût d'une solution. Un exemple illustratif complet est donné au fil de cette description. Un exemple de solution est proposé dans la section 6.1.4.

6.1.1 Données

Une instance du problème contient les informations concernant les aéroports, les appareils, les passagers et les vols sur une période englobant la *période de recouvrement* définie par ses dates et heures de début et fin.

Exemple. La période de recouvrement débute le 03/01/09 à 12:00 et se termine le 04/01/09 à 04:00. Le plan de vols est donné entre le 03/01/09 à 08:00 et le 04/01/09 à 04:00.

Une instance contient également le descriptif des perturbations ainsi que les paramètres

6.1 Description du problème

et coûts nécessaires au calcul du coût d'une solution. Plus précisément, une instance du problème contient les données suivantes :

- *Aéroports et capacités aéroportuaires* : Chaque aéroport est associé à des capacités de décollage et atterrissage par créneau horaire $[H, H + 1[$.

Exemple. Aéroports avec capacité de décollage et d'atterrissage par créneau horaire

```
AMS 1 1 00:00 00:00
CDG 1 1 00:00 06:00 4 4 06:00 22:00 0 1 22:00 00:00
IST 0 0 00:00 06:00 2 2 06:00 00:00
JFK 1 2 00:00 04:00 3 3 04:00 00:00
LHR 3 4 00:00 05:00 6 6 05:00 23:00 3 4 23:00 00:00
NCE 0 0 00:00 05:00 3 3 05:00 22:00 1 2 22:00 00:00
```

La ligne correspondant à l'aéroport CDG signifie que l'aéroport Charles de Gaulle est limité à 1 décollage et 1 atterrissage par créneau horaire entre 00:00 et 06:00 (créneaux [00:00, 01:00]; [01:00, 02:00]; ...; [05:00, 06:00]), 4 décollages et 4 atterrissages par créneau horaire entre 06:00 et 22:00, aucun décollage ne peut avoir lieu entre 22:00 et 00:00 et un seul atterrissage par créneau horaire est autorisé sur cette tranche horaire.

La *distance entre deux aéroports* est le temps de vol nominal inter-aéroportuaire, le *type de vol* (domestique (D), continental (C) ou intercontinental (I)) dépend de ces aéroports. On peut noter que bon nombre de vols opérés, ont un temps de vol légèrement inférieur à ce temps de vol nominal du fait, par exemple, de la vitesse de l'appareil qui l'opère. La distance correspond en fait à la distance maximale que l'on peut retrouver dans le plan de vols.

Exemple. Distance et type de vol

```
AMS NCE 120 C
CDG NCE 90 C
CDG IST 200 I
LHR JFK 446 I
```

Un vol entre Amsterdam (AMS) et Nice (NCE) dure au plus 120 minutes, il s'agit d'un vol de type continental.

- *Appareils et maintenance* : La flotte est constituée par l'ensemble des appareils qu'une compagnie aérienne opère. Chaque appareil est défini par un numéro de série unique, un *modèle* (Boeing 747, A320, ...) et une *configuration* de sièges, c'est-à-dire la répartition des sièges dans chaque cabine (*First/Business/Economic*). Les *temps de réengagements* sont communs à tous les appareils du même modèle, ils correspondent au temps minimum de préparation de l'appareil pour un nouveau vol : débarquement/embarquement des passagers et de leurs bagages, nettoyage de l'appareil, changement éventuel d'équipage... Des sous-ensembles d'appareils possédant des caractéristiques communes sont regroupés au sein de *familles* (par exemple les modèles A318, A319, A320, et A321 constituent la famille "AirbusSmall").

On connaît les appareils qui doivent subir une *maintenance* ainsi que le potentiel de vol restant de chaque appareil avant maintenance.

Exemple. Appareils et maintenance

```
A320#1 A320 AirbusSmall 0/20/160 2000.0 40 LHR
A320#2 A320 AirbusSmall 0/20/160 2000.0 40 LHR
A321#4 A321 AirbusSmall 10/10/160 2000.0 40 NCE
B767#5 B767 BoeingSmall 10/20/170 2000.0 40 CDG maintenance à CDG du 03/01/09
à 16:00 au 04/01/09 à 03:30, 180 minutes au plus d'heures de vols restantes
avant le début de la maintenance.
```

La description des caractéristiques de l'appareil Airbus A320#1 se lit comme suit : l'appareil comporte 0 place en cabine *First*, 20 places en cabine *Business* et 160 places en *Economic*. Une heure de vol coûte 2000€. Le temps de réengagement est de 40 minutes. Par défaut l'appareil est positionné à Londres Heathrow en début de période, et cet appareil n'a pas de maintenance prévue pendant la période de recouvrement.

Position des appareils à la fin de la période de recouvrement : le nombre d'appareils de chaque type (modèle et configuration) requis à chaque aéroport à la fin de la période de recouvrement est également donné en entrée.

Exemple. Positionnement des avions à la fin de la période de recouvrement

```
CDG A320 0/20/160 2 B767 10/20/170 1
JFK A320 0/20/160 1
```

En fin de période, à l'aéroport Charles de Gaulle, on doit avoir au moins 2 A320 de configuration de sièges 0/20/160 et au moins 1 B767 avec la configuration 10/20/170, sous peine de pénalités.

- *Vols et rotations* : L'ensemble des vols réalisés par la compagnie aérienne, aussi appelé, *programme de vols* ou *plan de vols* est connu sur la période de recouvrement. Chaque vol est défini de façon unique par son numéro de vol, les aéroports de départ et d'arrivée, les dates et horaires de départ et d'arrivée. On appelle *rotation* la séquence des vols alloués à un appareil.

Exemple. Vols et rotations des avions

```
1 NCE LHR 11:30 13:15 03/01/09 A320#1
2 LHR JFK 18:00 01:26+1 03/01/09 A320#1
3 NCE CDG 08:30 10:00 03/01/09 A320#2
4 CDG LHR 11:55 12:55 03/01/09 A320#2
5 LHR CDG 13:35 14:35 03/01/09 A320#2
6 CDG IST 15:15 18:35 03/01/09 A320#2
7 IST CDG 19:35 22:55 03/01/09 A320#2
8 NCE CDG 13:20 14:50 03/01/09 A321#4
```

6.1 Description du problème

9 CDG LHR 21:30 22:30 03/01/09 A321#4
 10 LHR CDG 23:10 00:10+1 03/01/09 A321#4
 11 AMS NCE 09:00 11:00 03/01/09 B767#5
 12 NCE CDG 12:20 13:50 03/01/09 B767#5

Le vol numéro 2 part le 03/01/09 à 18:00 de Londres Heathrow et arrive le lendemain à 01:26 à New York. Il est opéré par l'appareil A320#1.

La figure 6.1 représente la rotation de l'appareil A320#2 entre 08h00 et 00h00. L'appareil A320#2 opère les vols numéro 3,4,5,6 et 7. On peut noter qu'un intervalle horaire, correspondant au moins au temps de réengagement de l'appareil, sépare deux vols successifs.

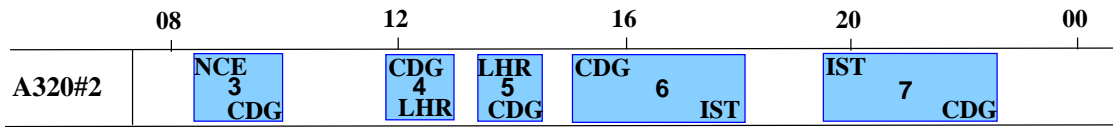


FIGURE 6.1 – La rotation de l'A320#2 entre 08h00 et 00h00.

Le programme de vols complet est représenté figure 6.2. Une barre grise plus fine matérialise la période de maintenance de l'appareil B767#5. Les vols représentés au-delà de la fin de la période de recouvrement illustrent la reprise du programme de vols initialement prévu.

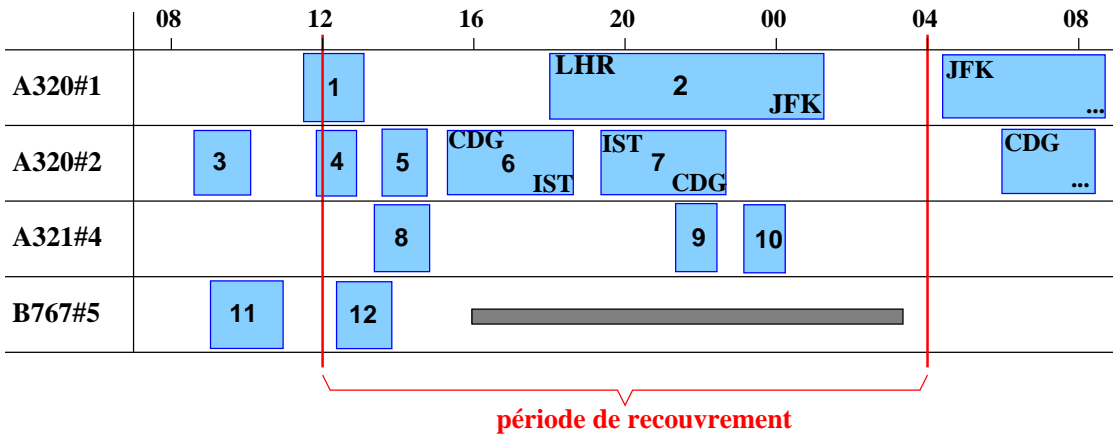


FIGURE 6.2 – Programme de vols initialement prévu.

Dans le challenge, des "vols" particuliers, dits *vols de proximité* sont ajoutés pour modéliser les acheminements par transport en communs (non aérien). Ces "vols" additionnels sont opérés par des appareils dédiés et ne sont pas modifiables. Ils correspondent typiquement au trajet d'une navette qui relie régulièrement les aéroports parisiens Charles de Gaulle et Paris Orly. Leur prise en compte permet d'élargir la complexité et la combinatoire du problème, notamment en ce qui concerne les cor-

respondances à Paris.

Par souci de clarté, nous ignorons dans ce rapport les vols dits *vols multi-tronçons*. Il s'agit en fait de vols avec escale. L'avantage est de réduire légèrement le temps de préparation de l'avion entre ces deux vols, appelé alors *temps de transit*.

Il est à noter qu'usuellement la plupart des vols sont répétés avec le même numéro de jour en jour.

- *Itinéraires* : Les passagers effectuent des réservations sur les vols proposés par la compagnie aérienne. Les itinéraires regroupent les passagers ayant effectués les mêmes réservations. Une réservation est définie par un numéro de réservation unique, le caractère aller ou retour du voyage, le nombre de passagers concernés par la réservation, le prix moyen payé par passager et la description de l'itinéraire, composé d'un ou de plusieurs vols à des dates précises, associé chacun à une cabine (*first* (F), *business* (B) ou *economic* (E)).

Exemple. Passagers et itinéraires

```

1 A 284.21 120 1 03/01/09 E
2 A 451.50 10 1 03/01/09 B
3 A 1423.20 84 2 03/01/09 E
4 R 1752.50 15 2 03/01/09 B
5 R 1845.00 25 1 03/01/09 E 2 03/01/09 E
6 A 1510.10 14 4 03/01/09 E 2 03/01/09 E
7 A 852.50 55 5 03/01/09 E 6 03/01/09 E
8 A 758.30 25 8 03/01/09 E 6 03/01/09 E
9 R 812.00 22 12 03/01/09 B 6 03/01/09 E
...
    
```

Par souci de lisibilité, on ne liste ici qu'une partie des passagers. Les passagers identifiés par le numéro de réservation 7, sont en voyage aller, ils sont 55 sur cette même réservation, et ont payé en moyenne 852.50 euros pour leur billet. Ils voyagent à bord des vols 5 puis 6 du 03/01/09 et sont en classe économique sur ces deux vols.

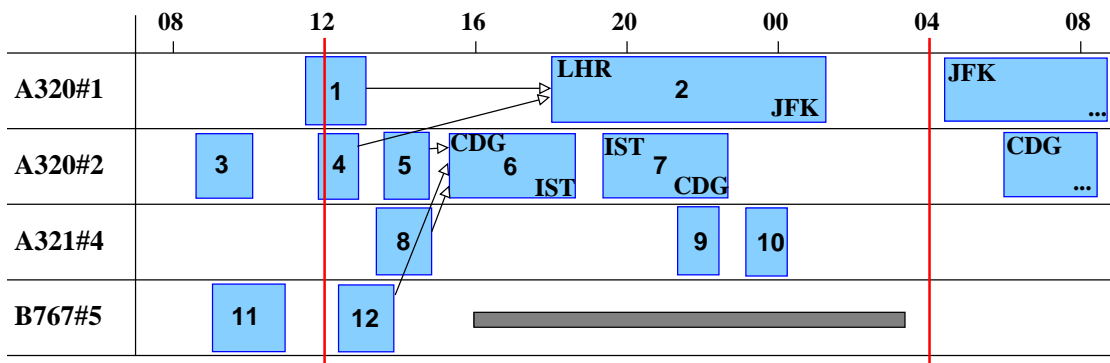


FIGURE 6.3 – Représentation des correspondances.

Ces différents passagers lient certains vols via des correspondances. Par exemple, les passagers de la réservation numéro 5, lient les vols 1 et 2. Cela signifie qu'un retard sur le vol 1, peut nécessiter un retard sur le vol 2 pour attendre les passagers en correspondance. Ou alors il faut proposer un itinéraire alternatif ou au pire annuler le voyage de ces passagers. Les liens entre les vols engendrés par les correspondances des passagers sont représentés par des flèches dans la figure 6.3.

Ces données décrivent le plan de vols initialement prévu, optimisé du point de vue des revenus de la compagnie aérienne.

Perturbations Le but du challenge est de gérer les impacts résultant d'une perturbation sur le plan de vols. On envisage les perturbations suivantes :

- *Vols perturbés* on envisage le retard ou l'annulation de vols. Un retard peut être dû à l'attente d'un passager ou d'un équipage en correspondance, ou bien à un problème opérationnel au sol. L'annulation de vol est en général due à un problème technique ou opérationnel de plus grande importance.
- Chaque *appareil perturbé* correspond à l'indisponibilité de l'appareil pour une période déterminée. Aucun vol ne peut lui être affecté pendant toute la période de son indisponibilité.
- Les *perturbations aéroportuaires* correspondent à des réductions éventuelles des capacités de décollage et d'atterrissage des aéroports. Ce cas de figure apparaît typiquement en cas de condition météorologique défavorable (brouillard), ou de grève du personnel. Ceci a notamment été le cas lorsque le nuage de fumée dû à l'éruption d'un volcan en Islande a paralysé une grande partie du trafic aérien européen. La fermeture d'aéroport n'a pas uniquement des conséquences locales, elle entraîne également des indisponibilités d'appareils (ceux-ci n'ayant pas le droit de décoller), et des perturbations de vols en cascade.

Exemple. Perturbations sur le plan de vols initialement prévu.

Retard et annulation de vols

4 03/01/09 330

11 03/01/09 -1

Le vol numéro 4 est retardé de 330 minutes, et le vol numéro 11 est annulé.

Pas de perturbations sur les appareils

Perturbations sur les aéroports

LHR 03/01/09 23:00 04/01/09 00:00 0 0

Aucun décollage ni atterrissage ne peut avoir lieu à Londres Heathrow entre 23:00 et 00:00 le 03/01/09.

Ces perturbations impactent un certain nombre de vols et de passagers, compromettant notamment certaines correspondances. Pour représenter ces impacts on a entouré les vols perturbés ainsi que les correspondances affectées (flèches) dans la figure 6.4.

Le but du challenge est de proposer et d'implémenter un algorithme de résolution qui génère un nouveau programme de vols pour la période définie par la période de recouvre-

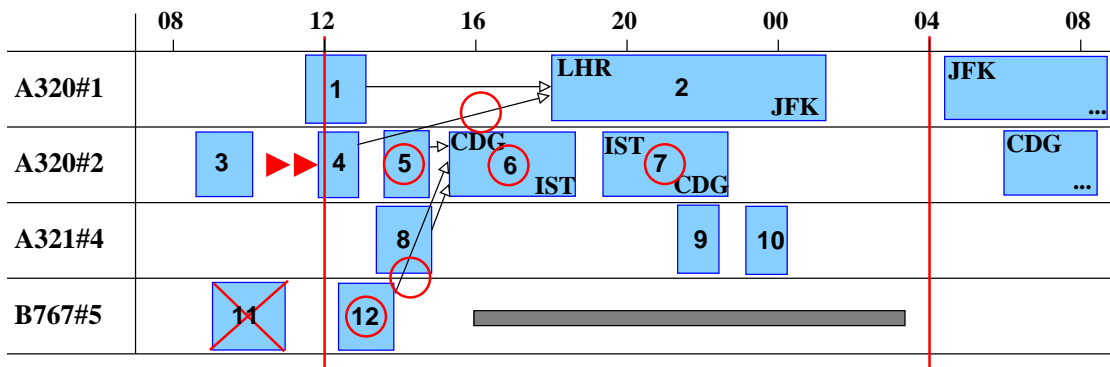


FIGURE 6.4 – Impact des perturbations.

ment ainsi que des nouveaux itinéraires pour les passagers. Ce nouveau plan de vols doit respecter un certains nombre de contraintes décrites dans la section 6.1.2. Cette solution doit minimiser les surcoûts induits aussi bien du point de vue opérationnel que du point de vue des passagers. Une description de ces surcoûts est donnée dans la section 6.1.3.

6.1.2 Contraintes

Contraintes inviolables. Les appareils, aéroports, itinéraires et rotations sont soumis à des contraintes. Celle-ci sont inviolables : le non respect d'une de ces contraintes rend la solution irréalisable.

Contraintes sur les appareils. – *Maintenance et indisponibilité* : la contrainte de maintenance oblige un avion d'être positionné dans un aéroport indiqué à la date fixée pour sa maintenance, et de le "clouer au sol" dans cet aéroport pour toute la durée de la maintenance. De plus, le nombre d'heures de vol entre le début de la période de recouvrement et le début de la maintenance est limité.

Un avion *indisponible* doit rester cloué au sol, peu importe dans quel aéroport, pendant toute sa période d'indisponibilité.

- *Capacité des cabines* : pour chaque vol, le nombre de passagers voyageant dans chacune des cabines ne doit pas excéder la capacité, en nombre de sièges, de cette cabine.
- *Changement d'avion* : un changement d'avion pour l'opération d'un vol ne peut se faire qu'à l'intérieur de la même famille d'appareils. Les équipages étant interchangeables sur des appareils de la même famille, cette contrainte facilite leur affectation sur le nouveau plan de vols (la rotation et l'affectation des équipages ne sont pas pour autant intégrés dans notre problème).
- *Temps minimum de réengagement* : deux vols consécutifs sur une rotation donnée doivent être espacés par un temps correspondant au temps minimum de réengagement de l'appareil opérant ces vols.

Contraintes aéroportuaires. – Les *capacités aéroportuaires* imposent une limite sur le nombre de décollages et d’atterrissages possibles par créneau horaire à chaque aéroport. Cette contrainte crée une dépendance entre les rotations des différents avions, et compte parmi les contraintes difficiles du problème. On peut cependant noter que l’on ignore les capacités de stationnement dans les aéroports, ainsi que bon nombre d’autres limitations aéroportuaires (disponibilité des portes, du personnel au sol, ...).

Contraintes sur les rotations. – *Contrainte liée à la période de recouvrement* : Tout vol étant déjà arrivé ou encore en cours au début de la période de recouvrement ne peut absolument pas être modifié. En particulier même un vol dont le départ est initialement prévu avant le début de la période de recouvrement, qui suite à un retard se trouve dans la période de recouvrement ne peut être modifié.

Contraintes passagers. Ces contraintes concernent les passagers dont l’itinéraire a été modifié. Si elles ne peuvent pas toutes être satisfaites, le voyage des passagers correspondant doit être annulé.

- Le *temps minimum de correspondance* doit être respecté pour tous les itinéraires passagers : deux vols consécutifs sur un itinéraire donné doivent être espacés d’au moins 30 minutes à l’aéroport de correspondance.
- Les passagers doivent être réaccommodés en conservant l’origine et la destination de leur itinéraire initial.
- Les itinéraires de remplacement ne doivent pas débiter avant l’heure de départ du premier vol de l’itinéraire initial.
- Le *retard global maximal* à l’arrivée à la destination finale d’un itinéraire ne doit pas excéder 18 heures pour un vol domestique ou continental et 36 heures pour un vol intercontinental. Cette contrainte s’applique à tous les passagers exceptés ceux qui sont en correspondance (ayant déjà débuté leur voyage au début de la période de recouvrement) ou ceux étant sur un itinéraire retour.

Toute violation de l’une de ces contraintes produit une solution irréalisable.

Bon nombre de contraintes paraissent assez naturelles et une simple propagation de retard avec, si nécessaire, des annulations de vols, suffit à les satisfaire. En pratique, c’est la combinaison entre les contraintes de maintenance et les capacités aéroportuaires qui est la plus difficile à satisfaire.

Contrainte avec pénalités de violation. Le *positionnement des appareils à la fin de la période de recouvrement* est la seule contrainte violable, sa violation impliquant une pénalité intégrée dans le terme *coûts de mauvais positionnement* de la fonction d’optimisation. Ces coûts sont décrits dans la section 6.1.3.

D'autres contraintes, comme par exemple, la réacommodation prioritaire des passagers en correspondance ou ceux étant sur un itinéraire retour est impliquée par les coûts d'annulation de ces différents passagers. Le non respect de cette priorité n'altère pas la réalisabilité de la solution, mais affecte seulement son coût.

6.1.3 Fonction d'optimisation

La fonction d'optimisation intègre des paramètres liés aux coûts ou gains induits par le nouveau programme de vols, des pénalités en cas de non-respect des contraintes de positionnement des appareils à la fin de la période de recouvrement ainsi qu'une mesure de la désutilité (notion expliquée ci-après) des passagers. Le but étant de minimiser une somme pondérée de ces différentes composantes.

Les *coûts opérationnels* incluent tous les frais associés à la réalisation d'un vol : ils sont estimés en fonction de l'appareil et du nombre d'heures de vol. Les coûts de retard et d'annulation pour les passagers sont inspirés de la réglementation européenne : 15 euros pour boissons et repas, 60 euros pour une nuit d'hôtel, leur attribution dépend du retard subi et de la durée initiale du trajet des passagers. En cas d'annulation, la compagnie aérienne doit rembourser le prix du billet, et apporter une compensation financière variable selon la durée initiale du trajet.

La *désutilité d'un passager* modélise le ressenti passager, indépendamment des compensations légales mentionnées ci-dessus. La désutilité, difficilement mesurable, est ramenée à une valeur en euros. Elle est divisée en deux parties : le coût de retard et le coût de déclassement. Le coût de retard est modélisé comme une fonction linéaire du retard global à l'arrivée avec un coefficient qui dépend du type d'itinéraire (type du vol le plus long emprunté par l'itinéraire) et de la cabine de référence de l'itinéraire (cabine de réservation la plus élevée des vols de l'itinéraire). Le coût de déclassement est appliqué en cas de déclassement sur un ou plusieurs tronçons de l'itinéraire.

Les *coûts de mauvais positionnement* des appareils à la fin de la période de recouvrement dépendent de la nature de la violation : violation de la configuration, du modèle ou de la famille de l'appareil requis.

Exemple. Nous ne détaillons pas ici tous les coûts, mais nous donnons un ordre de grandeur pour chacun d'eux.

- Coûts opérationnels :
 - coûts d'exploitation de l'appareil : 2 000€ par heure de vol, récupérés en cas d'annulation d'un vol, à payer pour créer un vol.
 - remboursement du billet en cas d'annulation : entre 284€ et 1 845€ par passager.
 - compensation financière légale en cas de retard : 15€ pour boissons et repas, 60€ pour une nuit d'hôtel par passager.
 - compensation financière légale en cas d'annulation : entre 200€ et 600€ par passager selon la durée initiale du trajet.
- Mesure de la désutilité des passagers :
 - coût de retard (désutilité) : 0.33€ par minute de retard pour un passager en classe Eco sur un vol domestique.

- coût d’annulation (désutilité) : entre 350€ et 3 000€ sur un vol Aller, entre 1 000€ et 9 000€ sur un vol Retour.
- coût de déclassement : entre 100€ et 750€.
- Coût de mauvais positionnement des appareils :
 - entre 1 000€ et 20 000€ selon la nature de la violation.

Il est clair qu’annuler le voyage d’un passager coûte très cher, et ce d’autant plus qu’il est en voyage retour, sur un trajet long et en classe de confort élevée. Dans tous les cas, un retard, même très important, coûte beaucoup moins cher qu’une annulation.

6.1.4 Structure d’une solution

Une solution du problème est composée d’un nouveau programme de vols et de nouveaux itinéraires pour l’ensemble des passagers. Elle reprend l’intégralité des informations disponibles dans les fichiers d’entrées des rotations et itinéraires en y incluant les modifications requises pour répondre aux perturbations ainsi que d’éventuels nouveaux vols et de nouveaux itinéraires. Le nouveau programme de vols résulte d’un certain nombre de décisions quant aux vols du programme initial : *annulations et retards* intentionnels, *changement d’appareils* au sein de la même famille et *créations* éventuelles de nouveaux vols. Les nouveaux itinéraires peuvent, quant à eux, résulter d’*annulations et retards* intentionnels, de *changements de vols* en respectant l’origine et la destination initiale, de *déclassements et surclassements* éventuels.

L’objectif est de trouver une solution qui engendre un coût le moins élevé possible pour la compagnie aérienne. On cherche donc une solution qui réalise le meilleur compromis entre le nouveau plan de vols, son impact sur les passagers et son impact sur la reprise normale des opérations après la fin de la période de recouvrement.

Exemple. Solution sans gestion de la perturbation, simple propagation des retards.

Sans gestion "intelligente", la perturbation entraîne retards et annulations en cascades et peut même produire une solution irréalisable, notamment si un appareil ne peut pas être amené à temps à son aéroport de maintenance. La figure 6.5 représente le plan de vols que l’on obtient en propageant les retards et annulations, et en annulant le vol 7 qui déborde au delà de la période de recouvrement. Ce plan de vols est réalisable, mais de nombreux voyages de passagers doivent être annulés, soit à cause d’une annulation de vol, soit à cause d’une correspondance qui ne peut plus être réalisée. Cette solution coûterait donc très cher à la compagnie aérienne.

Exemple. Solution pour le recouvrement des perturbations.

Afin d’absorber, au moins partiellement, le retard en cascade subi par les vols succédant au vol 4, on échange partiellement les rotations des appareils A320#2 et A321#4. On retarde les vols 2, 6 et 7 pour permettre la correspondance des passagers retardés. Le vol 10 est, quant à lui, retardé pour cause de capacités aéroportuaires insuffisantes. L’appareil B767#5 n’étant pas arrivé à l’aéroport de départ du vol 12, celui-ci est annulé.

Le nouveau programme de vol est représenté figure 6.6, il est également donné de façon

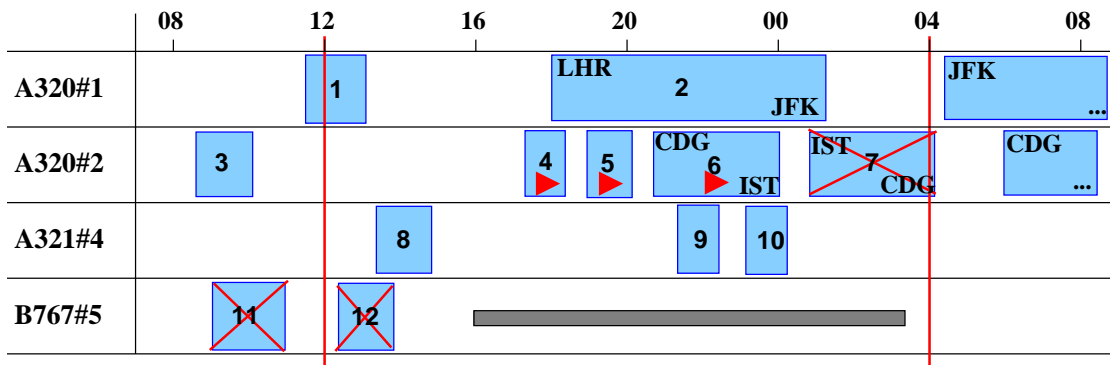


FIGURE 6.5 – Solution avec propagation des retards.

explicite. La réacommodation des passagers est détaillée, de nombreux passagers sont retardés mais un seul voit son voyage annulé. Cette solution coûte donc beaucoup moins cher à la compagnie aérienne que la solution précédente. Nous détaillons une partie de son coût ci-après.

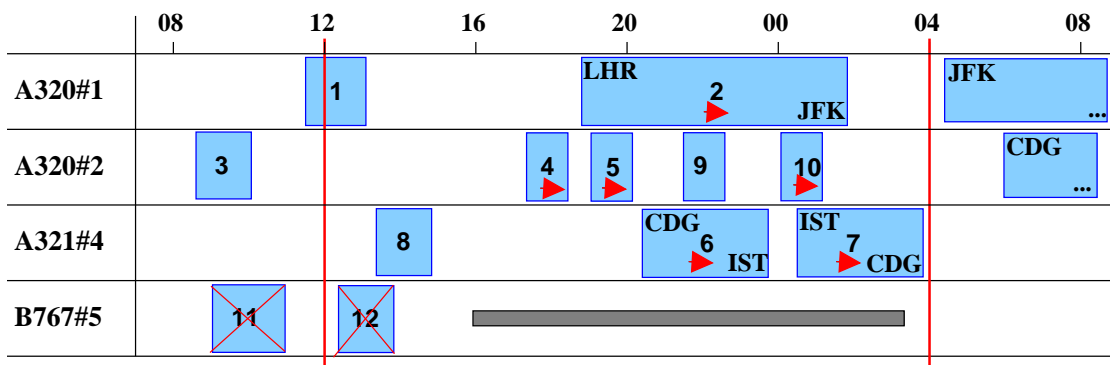


FIGURE 6.6 – Solution pour un meilleur recouvrement possible des perturbations.

Solution : vols et rotations des avions.

- 1 NCE LHR 11:30 13:15 03/01/09 A320#1
- 2 LHR JFK 18:55 02:21+1 03/01/09 A320#1 (vol retardé de 55 minutes)
- 3 NCE CDG 08:30 10:00 03/01/09 A320#2
- 4 CDG LHR 17:25 18:25 03/01/09 A320#2 (vol retardé de 330 minutes)
- 5 LHR CDG 19:05 20:05 03/01/09 A320#2 (vol retardé de 330 minutes)
- 6 CDG IST 20:35 23:55 03/01/09 A321#4 (changement d'appareil et retard 320 minutes)
- 7 IST CDG 00:35 03:55 03/01/09 A321#4 (changement d'appareil et retard 300 minutes)
- 8 NCE CDG 13:20 14:50 03/01/09 A321#4
- 9 CDG LHR 21:30 22:30 03/01/09 A320#2 (changement d'appareil)
- 10 LHR CDG 00:00+1 01:00+1 03/01/09 A320#2 (changement d'appareil et retard 50 minutes)
- 11 AMS NCE 09:00 11:00 03/01/09 cancelled (vol annulé)

6.1 Description du problème

12 NCE CDG 12:20 13:50 03/01/09 cancelled (vol annulé)

Solution : réacommodation des passagers.

1 A 284.21 120 1 03/01/09 E
 2 A 451.50 10 1 03/01/09 B
 3 A 1423.20 84 2 03/01/09 E (retard 55 minutes)
 4 R 1752.50 15 2 03/01/09 B (retard 55 minutes)
 5 R 1845.00 25 1 03/01/09 E 2 03/01/09 E (retard 55 minutes)
 6 A 1510.10 14 4 03/01/09 E 2 03/01/09 E (retard 55 minutes)
 7 A 852.50 55 5 03/01/09 E 6 03/01/09 E (retard 320 minutes)
 8 A 758.30 159 8 03/01/09 E 6 03/01/09 E (retard 320 minutes)
 9 R 812.00 10 8 03/01/09 B 6 03/01/09 B (changement 1^{er} vol et retard 320 min)
 9 R 812.00 10 8 03/01/09 F 6 03/01/09 F (surclassement et retard 320 min)
 9 R 812.00 1 8 03/01/09 E 6 03/01/09 E (déclassement et retard 320 min)
 9 R 812.00 1 cancelled (voyage annulé) ...

Coût pour la compagnie aérienne Dans le tableau 6.1, on a détaillé le coût de la solution décrite ci-dessus.

Coût opérationnel		
exploitation appareil	annulation vols 11 et 12 : 2 000€ * 3.5h	-7 000.00
remboursement billet	annulation passager 9A	+812.00
compensation financière	9A trajet initial 4h50	+600.00
Coût désutilité		
retard légal	repas 9RD	+15.00
retard légal	hotel 9RD	+60.00
retard désutilité	9RD business, intercontinental 1€/min * 320min	+320.00
annulation désutilité	9A business, intercontinental, retour	+6 000.00
déclassement	9RD déclassement B→ E, intercontinental	+300.00
Coût repositionnement		
mauvais positionnement	aucun	+0.00

TABLE 6.1 – Coût de la solution en considérant les appareils et seulement un passager annulé et un passager retardé et déclassé.

Pour plus de lisibilité, on ne détaille ici que les coûts concernant les deux derniers passagers de la réservation 9 : l'un est déclassé et retardé de 320 minutes, l'autre voit son voyage annulé. On notera ces passagers respectivement 9RD et 9A. Ces passagers sont sur un vol intercontinental, en voyage retour et leur cabine de référence est business. On ne calcule pas le coût engendré par le retard des autres passagers de la réservation 9, ni par les passagers des réservations 3 à 8.

6.2 Etat de l'art

Le but de cet état de l'art est double. Il vise, d'une part, à placer le contexte du domaine aérien, à présenter le processus de planification des compagnies aériennes commerciales et à citer quelques travaux essentiels de la Recherche Opérationnelle dans ce domaine. D'autre part, il rassemble une partie des travaux récents concernant la gestion de perturbation. Pour connaître les dernières avancées et applications de la Recherche Opérationnelle dans l'industrie des compagnies aérienne, nous invitons le lecteur à consulter le site internet de l'AGIFORS *Airline Group of the International Federation of Operational Research Societies*¹. L'AGIFORS est une des sociétés qui montre le lien fort entre la Recherche Opérationnelle et le domaine aérien.

C'est dès le début des années 50 qu'Air France a mis en place un service de Recherche Opérationnelle [Roy, 2006]. Cependant, étant donnée la puissance des "ordinateurs" de l'époque, une grande partie des travaux d'optimisation, notamment pour la gestion de perturbation, se faisait manuellement. Cette tradition est encore bien ancrée. La Recherche Opérationnelle est utilisée comme un outil d'aide à la décision : elle propose des solutions permettant à la compagnie d'être la plus performante possible dans l'utilisation de ses ressources, mais c'est le centre de contrôle des opérations qui prend les décisions.

6.2.1 Planification dans le domaine aérien : processus et modèles

Processus de planification. La mise au point des programmes de vols nécessite à la fois beaucoup d'anticipation et de nombreux réajustements et raffinements. De manière générale, la planification suit un ordre séquentiel : la priorité est aux appareils qui constituent la première richesse d'une compagnie aérienne (c'est son outil de production). En second lieu on gère les équipages, et finalement les passagers.

Le schéma figure 6.7 donne une vision globale de cette planification. Il permet de replacer les différentes étapes de la planification aussi bien dans le temps que dans les niveaux de planification et même dans leur impact sur le *revenue management*.

En amont, un premier ensemble de vols est déterminé en se fondant notamment sur les prévisions de la demande passager. On assigne ensuite des types d'avions à ces vols, puis on établit la rotation des avions avec respect des maintenances. On construit alors la rotation des équipages ainsi que les plannings des équipages. Finalement, on affecte les passagers sur les vols opérés et on construit tous les plannings opérationnels (affectation des avions aux portes, ...). Ces différentes étapes se déroulent séquentiellement mais ne sont bien entendu pas indépendantes. Un certain nombre de liens sont représentés figure 6.7. Dans le cas idéal, les vols effectifs suivent ce planning optimisé. Mais, en cas de perturbation majeure affectant significativement plusieurs vols, le calcul d'une solution de recouvrement nécessite de reconsidérer les décisions faites aux trois niveaux de planifications. Le but est alors clairement de "limiter la casse", en minimisant au maximum les conséquences des perturbations.

Dans le cadre du challenge, nous nous intéressons exclusivement à la construction du

1. <http://www.agifors.org>

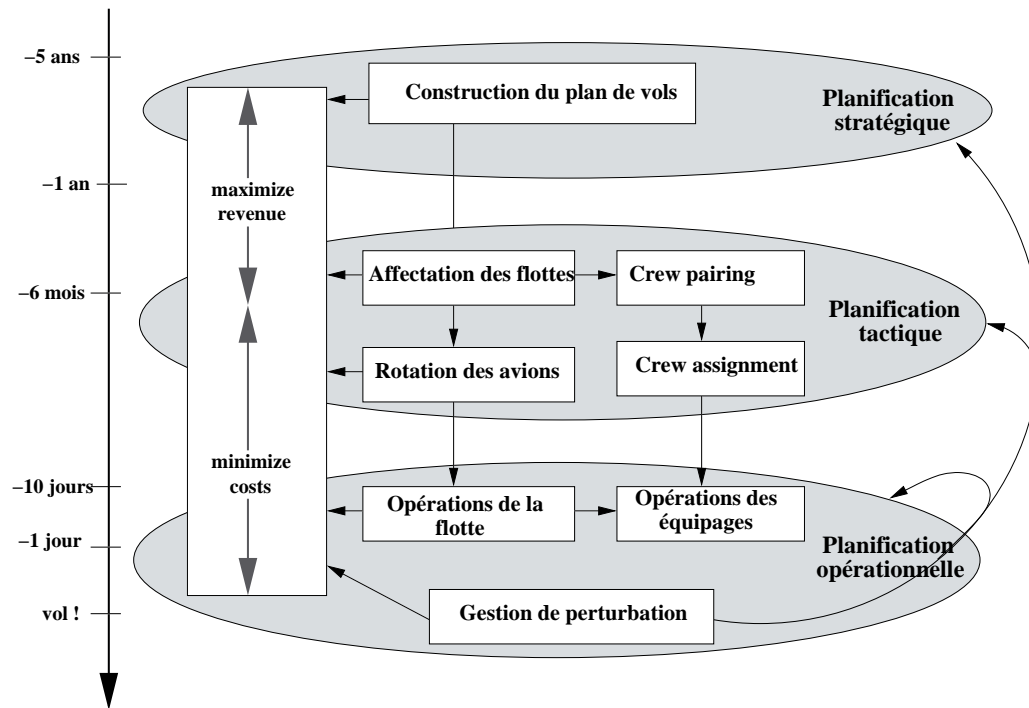


FIGURE 6.7 – Processus de planification dans le domaine aérien.

nouveau plan de vols et à la réaffectation des passagers. Nous présentons ici les modèles couramment utilisés pour construire les rotations des avions lors de la phase de planification. Ces modèles peuvent, bien entendu, être adaptés pour reconstruire ces rotations en cas de perturbation. Nous citons également quelques articles fondamentaux traitant d'une ou de plusieurs étapes de la planification.

Modèles de réseaux. Dans les problèmes de planification du domaine aérien, les modèles d'optimisation sont principalement fondés sur deux types de réseau selon que les actions (vols) sont représentées par les nœuds ou par les arcs du réseau. Nous illustrons les différents réseaux classiquement utilisés avec pour exemple simple les rotations des avions A320#2 et A321#4.

- Le *connection network*, ou *réseau des connexions* (figure 6.8) a pour but de représenter les différentes possibilités pour construire les rotations des avions. Les nœuds de ce réseau correspondent aux différents vols (tronçons), un arc joint deux nœuds s'il est possible d'enchaîner les vols correspondant avec le même appareil (respect du temps de réengagement et même aéroport intermédiaire). On ajoute des nœuds qui indiquent la position des appareils au début et à la fin de la période de temps considérée, et on connecte ces nœuds aux nœuds précédents si le vol est possible à partir de cette position initiale (respectivement vers cette position finale). Pour l'exemple on se limite aux rotations des appareils A320#2 et A321#4 entre

08:00 le 03/01/09 et 01:00 le lendemain.

Exemple. 3 NCE CDG 08:30 10:00 03/01/09 A320#2
 4 CDG LHR 11:55 12:55 03/01/09 A320#2
 5 LHR CDG 13:35 14:35 03/01/09 A320#2
 6 CDG IST 15:15 18:35 03/01/09 A320#2
 7 IST CDG 19:35 22:55 03/01/09 A320#2
 8 NCE CDG 13:20 14:50 03/01/09 A321#4
 9 CDG LHR 21:30 22:30 03/01/09 A321#4
 10 LHR CDG 23:10 00:10+1 03/01/09 A321#4

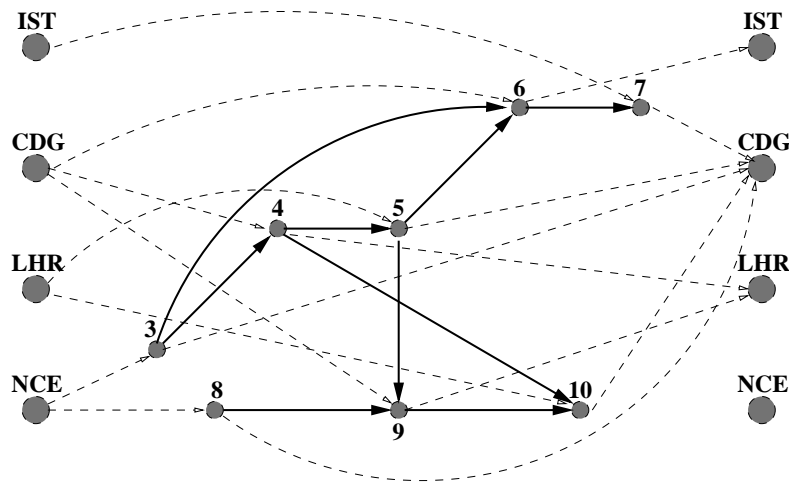
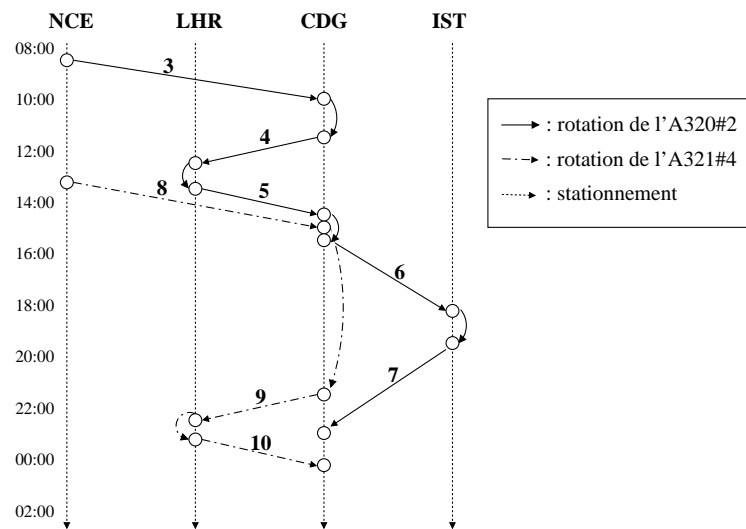


FIGURE 6.8 – Réseau des connexions. La rotation de l'appareil A320#2 correspond au chemin NCE-3-4-5-6-7-CDG. La rotation de l'appareil A321#4 correspond au chemin NCE-8-9-10-CDG.

Un chemin dans ce réseau correspond à un morceau de rotation. Les maintenances peuvent s'intégrer dans ce réseau, par exemple en considérant la maintenance comme un vol fictif. On voit toutefois que le nombre de chemins possibles est important, et ce nombre augmente d'autant plus que la fenêtre de temps se rallonge.

- Le *time line network* (figure 6.9) a une structure de réseau espace-temps. Il permet de représenter les différents vols et horaires de façon plus explicite que le réseau des connexions.

Les nœuds de ce réseau correspondent aux différents couples (aéroport de départ, heure de départ) et (aéroport d'arrivée, heure d'arrivée) des différents vols. Un arc dans ce réseau représente soit un vol réalisable entre deux aéroports à des horaires donnés, soit une attente dans un même aéroport. De même que pour le réseau de connexions, un chemin dans ce réseau correspond à un morceau de rotation. Les

FIGURE 6.9 – Réseau *time line network*.

maintenances s'intègrent par exemple en incluant le temps de la maintenance dans le temps de parcours du vol la précédant.

- Le *time band network* a lui aussi une structure de réseau espace-temps. C'est l'adaptation du *time line network* pour gérer d'éventuelles perturbations. Ce réseau est construit de façon dynamique lors d'une perturbation. Les nœuds de ce réseau correspondent aux nœuds d'un réseau espace-temps agrégés par bande de temps (bande de 30 minutes par exemple). On ajoute à ces nœuds des nœuds puits à la fin de la période de recouvrement. Chaque nœud est labellisé par l'heure de disponibilité d'un appareil dans ce nœud. Un vol prévu est représenté par un arc dans ce réseau. Cet arc est dupliqué sur tous les nœuds futurs à partir desquels ce vol est potentiellement réalisable (dans lesquels un appareil est disponible et à partir desquels le vol arrive bien avant la fin de la période de recouvrement). On construit ainsi des arcs parallèles.

Un flot dans ce réseau donne une solution de recouvrement : les arcs initiaux qui n'ont pas de flot correspondent aux vols annulés, les arcs dupliqués ayant du flot correspondent à des vols retardés.

Le modèle que nous proposons pour résoudre notre problème est en partie inspiré de ce type de réseau.

Travaux existants. Nous citons ici quelques articles essentiels concernant la planification dans le domaine aérien. Les travaux dans ce domaine traitent en général d'un seul ou de deux des problèmes apparaissant aux différents niveaux de planification.

Les problèmes de construction des rotations des appareils et d'affectation des flottes sont traités dans [Clarke *et al.*, 1997] (relaxation lagrangienne) et [Barnhart *et al.*, 1998]

(*Branch-and-Price*).

Les problèmes de gestion des équipages, *crew pairing* et *crew assignment*, sont en général modélisés sous la forme de problème de partitionnement et de recouvrement. Les méthodes de résolution exploitent donc cette structure. Nous ne citons ici que [Hoffman et Padberg, 1993] qui propose un algorithme de *Branch-and-Cut* pour résoudre le problème de planification pour les équipages.

Ce problème a aussi été résolu simultanément avec la construction des rotations d'appareils dans [Cordeau *et al.*, 2001]. Ce travail constitue une première vraie intégration de flotte d'appareils et équipages, la résolution est fondée sur une décomposition de Benders.

La construction de planifications robustes est elle aussi en vogue. L'idée est d'intégrer dans le planning la possibilité d'absorber des perturbations ou au moins de faciliter le recouvrement des perturbations. Dans la pratique, cette robustesse est largement intégrée dans les plannings pour des "petites perturbations" (pour absorber par exemple un retard de 10 minutes); cependant face à des perturbations plus importantes l'optimisation reste nécessaire. Nous présentons ici quelques travaux, notamment ceux qui nous ont inspirés pour modéliser et résoudre le problème du challenge. Une revue de la littérature plus complète sur les problèmes de gestion de perturbation est réalisée dans [Clausen *et al.*, 2009]. On distingue les travaux qui ne traitent que d'une seule ressource et ceux qui, comme pour le challenge, affichent la volonté d'un recouvrement intégrant simultanément plusieurs ressources.

6.2.2 Recouvrement d'une seule ressource

Pour la gestion des perturbations dans le domaine aérien, les problèmes académiques traités dans la littérature sont souvent des versions très simplifiées des problèmes réels du terrain. En effet, la plupart des travaux ne s'intéressent qu'à une seule ressource et les passagers sont rarement considérés [Kohl *et al.*, 2007]. La littérature traite de façon souvent séparée les principaux sous-problèmes de gestion de perturbations (*disruption management*): réaffectation des appareils, réaffectation des équipages et (plus rarement) réacommodation des passagers.

Une grande part des travaux de gestion de perturbation porte sur le recouvrement des appareils. En effet, c'est la ressource la plus chère et la plus rare pour une compagnie aérienne. Le problème de recouvrement des appareils, ou *aircraft recovery problem* a été largement étudié dans les quinze dernières années, et bon nombre de techniques de Recherche Opérationnelle ont été appliquées. Les différents problèmes de recouvrement des appareils diffèrent par les contraintes prises en compte ou non (couvre-feu d'aéroports, capacités aéroportuaires, type de perturbation, prise en compte des maintenances, ...) et par les fonctions d'optimisation choisies (minimiser le retard total, minimiser le nombre de vols annulés, ...).

[Jarrah *et al.*, 1993] modélisent le problème de recouvrement des appareils avec deux modèles de flot. Le premier a pour objectif de minimiser les coûts de retard des vols, alors

que dans le second, l'objectif est de minimiser les coûts d'annulation des vols. Ce travail a ensuite été repris en intégrant les deux critères. Les modèles à base de multiflot [Andersson, 2001, Andersson et Värbrand, 2004] sont aussi couramment utilisés, mais l'impact de la solution de recouvrement sur les passagers est ignoré.

On retrouve bien entendu les techniques de recherche locale [Love *et al.*, 2002], l'utilisation de métaheuristiques [Andersson, 2006] ainsi que la génération de colonnes [Rosenberger *et al.*, 2003].

Le problème de recouvrement des équipages est également largement étudié. Ce problème est encore plus complexe que le précédent du fait du nombre d'équipages (*flight crew* et *cabin crew*) et des nombreuses règles, contraintes, législations et préférences personnelles à prendre en compte. Nous référons le lecteur intéressé à ce problème à l'état de l'art présenté dans [Clausen *et al.*, 2009].

Très peu de travaux portent sur le problème de recouvrement des passagers seuls [Lettovsky, 1997]. En effet, une fois que toutes les décisions concernant les appareils et les vols sont prises, et que le planning de recouvrement est calculé, la réaffectation des passagers est un "simple" multiflot. Par contre, comme nous le verrons dans la prochaine section, certains problèmes de recouvrement intégrant plusieurs ressources intègrent aussi les passagers. Cependant, la plupart du temps cette intégration reste implicite : elle est uniquement fondée sur une approximation des coûts de retard.

6.2.3 Recouvrement intégrant plusieurs ressources

L'intégration de plusieurs ressources dans le recouvrement a, la première fois, été traitée par [Teodorović et Stojković, 1995]. Il s'agit du recouvrement intégré des appareils et des équipages avec les contraintes de maintenance. D'autres travaux concernant le recouvrement intégré des appareils et des équipages sont ceux de Stokjovic et Soumis. On ne cite ici que [Stojković et Soumis, 2001].

La thèse de [Lettovsky, 1997] propose un modèle linéaire mixte intégrant les trois ressources principales : appareils, équipages et passagers. Le problème est décomposé en trois sous-problèmes, un par ressource, et ces trois parties sont contrôlées par un problème maître. La technique de résolution utilise des décompositions de Benders mais l'implémentation ne reste que partielle.

Dans leurs travaux, [Klincewicz et Rosenwein, 1995] adaptent le plan de vols ainsi que la rotation des appareils afin de faire face à une demande variable de la part des passagers. C'est en quelque sorte un problème de recouvrement appareils/passagers. Ni les maintenances, ni les équipages ne sont intégrés dans ce problème. La résolution utilise des modèles à base de flot pour générer une nouvelle affectation des appareils. [Bratu et Barnhart, 2006] a pour objectif premier le recouvrement des passagers, mais intègre également des contraintes concernant les appareils et les équipages. C'est un des premiers travaux qui place véritablement les passagers au cœur du recouvrement.

6.3 Les méthodes proposées par les finalistes

Les méthodes proposées par les différentes équipes participantes sont des méthodes heuristiques résolvant le problème en le décomposant en plusieurs étapes et utilisant une grande variété d'approches. Nous présentons ici brièvement les méthodes utilisées par les 8 autres équipes finalistes. Pour plus de détails, le lecteur est invité à consulter les actes de la conférence ROADEF 2009. Les schémas généraux de résolution se répartissent en algorithmes purement gloutons, méthodes de descente avec redémarrage, recuit simulé, méthodes hybrides de programmation linéaire en nombres entiers et de recherche locale, méthodes de génération de colonnes. Les algorithmes élémentaires de flots et de chemins optimaux sont largement utilisés. De nombreuses solutions intègrent une procédure d'amélioration, en général elle vise à réaccommoder des passagers en créant de nouveaux vols selon les besoins.

Voici brièvement les méthodes de résolution utilisées par 9 équipes finalistes (dont la notre). Les méthodes sont données dans l'ordre du classement du challenge.

1. **Une méthode de recherche à grand voisinage** - S. Bisailon, J.-F. Cordeau, G. Laporte et F. Pasin - *Montréal*.

La méthode de recherche à grand voisinage est une approche itérative qui exploite au maximum le temps imparti, chaque itération se décompose en trois phases.

- (a) Une phase de construction d'une solution initiale avec respect des maintenances par suppressions et réinsertions de séquences de vols. La solution viole éventuellement les capacités aéroportuaires, les modifications sur les itinéraires sont gérées simultanément (annulations, retards). Ceci est fait avec un ordre aléatoire sur les avions, ordre modifié à chaque itération.
- (b) Une phase de réparation de la solution pour satisfaire les contraintes de capacités aéroportuaires par annulation de vols et des itinéraires correspondant, ceci est fait avec le même ordre que l'étape précédente.
- (c) Une phase d'amélioration locale : insertions de vols, décalages de vols, et algorithme de plus court chemin pour réaccommoder passagers.

2. **Stratégie d'oscillation** - S. Hanafi, C. Wilbaut, R. Mansi et F. Clautiaux - *Lille*.

La première étape est de construire une solution initiale en relaxant les contraintes de capacité des cabines et des aéroports mais qui satisfait les contraintes dures de maintenance. La solution obtenue est alors améliorée en alternant des phases constructives et destructives. L'amélioration est obtenue en ajoutant et supprimant itérativement des routes et des itinéraires. Pour finir, une phase de post-optimisation permet de créer des vols pour réaccommoder les derniers passagers.

3. **Statistical Analysis of Propagation of Incidents (SAPI)** - R. Acuna-Agost, P. Michelon, D. Feillet et S. Gueye - *Avignon*.

La méthode SAPI a été développée initialement pour résoudre le problème de réordonnancement de plan de circulation ferroviaire en cas d'incidents. La méthode a

été adaptée et appliquée au problème de gestion de perturbation dans le domaine aérien. Le problème est modélisé intégralement via un MIP comportant deux multiflots (un sur les appareils, un concernant les passagers). Ce modèle est de très grande taille (taille du réseau, nombre de vols, nombre de passagers), et ne peut être résolu directement avec Cplex. La méthode SAPI consiste à utiliser un modèle statistique pour évaluer la probabilité qu'un vol soit affecté par les perturbations et ainsi pouvoir diminuer la complexité du problème en fixant les variables selon ces probabilités.

4. **La méthode TUE** - C. Eggermont, M. Firat, C. Hurkens et M. Modelski - *Eindhoven*.

L'approche est fondée sur une décomposition du problème, chaque étape améliore un aspect de la solution. Les différentes étapes sont résolues de façon séquentielle.

- (a) Réparation de la continuité dans les rotations des avions.
- (b) Respect des maintenances (en priorité) conjointement aux capacités aéroportuaires.
- (c) Retard supplémentaires sur vols pour garantir plus de correspondances aux passagers.
- (d) Réacommodation finale des passagers.

5. **Notre approche** - J. Darlay, L.-P. Kronek, S. Schrenk, L. Zaourar - *Grenoble*.

En quelques mots, nous avons décomposé le problème en deux grandes étapes résolues de façon séquentielle.

- (a) Calcul d'un nouveau plan de vols proche du plan de vol initial. Le plan de vols est calculé par morceaux. On traite d'un nombre limité d'avions à chaque fois, et on reconstruit un plan de vols pour ces avions en résolvant un modèle de multiflot dans un réseau (arc=vols) avec contraintes additionnelles. En priorité, on détermine les rotations des appareils ayant une maintenance, avec si besoin création de vols.
- (b) Réacommodation des passagers sur le nouveau plan de vols, en résolvant un problème de multiflot.

6. **Un recuit simulé** - J. Peekstok et E. Kuipers - *Pays-Bas*.

On commence par construire une solution initiale, non réalisable. On cherche ensuite par recherche locale à améliorer cette solution et à la rendre réalisable, pour ce faire les contraintes violées entraînent des pénalités dans la fonction objectif. Les mouvements considérés dans la recherche locale sont :

- avancer/retarder un vol, changer l'appareil qui opère un vol, modifier un vol dans l'itinéraire d'un passager,
- suppression/création de vols pour obtenir solution réalisable,
- ajout/suppression/modification d'un vol dans l'itinéraire d'un passager.

Finalement, on applique une phase d'amélioration qui consiste à créer des vols et à réacommoder les passagers, ceci en boucle.

7. **La méthode NCF** - N. Jozefowicz, C. Mancel et F. Mora-Camino - *Toulouse*.

Il s'agit d'une approche heuristique à trois phases. La première phase génère une solution réalisable après avoir considéré séquentiellement toutes les perturbations du problème. La solution est composée d'un ensemble de rotations d'avions qui respecte toutes les contraintes y compris les maintenances prévues. A la fin de cette étape, certains itinéraires peuvent avoir été annulés à cause des perturbations. Une deuxième phase améliore la solution courante en faisant intervenir différents modules de modification des rotations et réacommodation de passagers. La partie clé de cette phase est l'insertion de séquences de vols réalisables afin d'acheminer les passagers à leurs destinations finales. La troisième phase consiste à créer de nouveaux vols pour minimiser des pénalités en cas de non-respect des contraintes de positionnement des appareils à la fin de la période de recouvrement.

8. **La méthode australienne** - S. Dickson, O. Smith et W. Li - *Melbourne*.

La méthode développée est un processus à deux phases fondé sur un découpage appareils/passagers.

- (a) La première phase s'attache à réaffecter les appareils, de changer l'horaire ou d'annuler des vols en vue de minimiser les perturbations sur les itinéraires de passagers. Cette étape est réalisée en résolvant un MIP dans un réseau des connexions (nœuds = vols), avec en particulier des variables de connexions entre vols et des variables continues pour les retards de vols. Pour limiter sa taille le modèle est paramétré, on peut ainsi limiter le retard maximal, limiter le temps possible au sol entre 2 vols connectés, ...
- (b) La seconde phase optimise les itinéraires des passagers par rapport au plan de vols fixé dans la première phase. Elle est fondée sur la résolution d'un multiflot dont l'objectif est de maximiser la valeur des itinéraires transitant à travers le réseau des vols opérés après la première phase.

9. **Une génération de colonnes dans un réseau spécifique** - N. Eggenberg et M. Salani - *Lausanne*.

La méthode a initialement été développée pour créer des plannings robustes, elle est adaptée et appliquée au problème de gestion de perturbation dans le domaine aérien. Il s'agit d'un algorithme à base de génération de colonnes qui résout un modèle MIP spécifique visant à chercher la meilleure solution de recouvrement. Le problème maître consiste à optimiser l'utilisation des appareils et minimiser le nombre de vols annulés. Le problème esclave vise à proposer d'autres alternatives pour le recouvrement (génération de nouvelles routes pour les avions (rotations) via un plus court chemin contraint). Une fois ce maître résolu, les passagers sont réacommodés par ordre de priorité via un modèle de flot.

Le tableau 6.2 classe les méthodes utilisées par les 9 équipes finalistes. On retrouve dans la plupart des méthodes une décomposition du problème. Les approches fondées sur

6.3 Les méthodes proposées par les finalistes

ces décompositions sont de deux types. On a d'abord les approches séquentielles dans lesquelles, on résout un étape de la décomposition à la fois et on enchaîne sur l'étape suivante. Par opposition à ces méthodes directes, on a les méthodes itératives qui consistent à résoudre le problème en calculant successivement des solutions de meilleure qualité à partir d'une solution initiale. Dans les méthodes itératives, on revient donc sur ses choix.

	1	2	3	4	5	6	7	8	9
Méthode									
MIP			×		×			×	×
Recherche locale	×	×		×		×	×		
Décomposition									
itérative	×	×				×			×
séquentielle				×	×		×	×	
Solveur	?	×	×		×			×	×

TABLE 6.2 – Approches de résolution des différents finalistes.

Toutes les équipes ont noté la difficulté des contraintes de maintenance. De manière générale, les méthodes exploitent au maximum les plannings initiaux, les vols sont décalés dans le temps ou annulés, et en général de nouveaux vols sont créés pour satisfaire les contraintes de maintenance et éventuellement en post-optimisation.

Chapitre 7

Notre approche de résolution

Sommaire

7.1	Structure de l'algorithme de résolution	127
7.2	Construction d'un nouveau plan de vols	128
7.2.1	Schéma général et modélisation	129
7.2.2	Décomposition en paquets.	133
7.2.3	Création de nouveaux vols pour les maintenances.	133
7.2.4	Coûts de réaffectation	135
7.3	Réaccommodation des passagers	136
7.3.1	Itinéraires alternatifs pour la réaccommodation	136
7.3.2	Modélisation	137

Le but de ce chapitre est de présenter notre heuristique de résolution. Nous proposons un algorithme constructif, qui produit un nouveau programme de vols réalisable ainsi qu'une réaccommodation des passagers, et qui vise à minimiser les coûts opérationnels et fonctionnels. Notre algorithme utilise principalement des modèles de flots, ceux-ci ont été présentés dans la partie I.

Nous rappelons que dans le contexte du challenge, il a fallu trouver une méthode de résolution efficace qui résoud des instances de grande taille en moins de 10 minutes. De plus, il a fallu implémenter et fournir un programme en quelques mois seulement. La méthode de résolution n'est pas très sophistiquée, mais elle a le mérite de nous avoir permis d'être classés 5ème parmi les 25 équipes en lice. En effet, lors de cette compétition en temps limité, proposer une approche originale et l'implémenter "from scratch" est quasi mission impossible.

7.1 Structure de l'algorithme de résolution

Notre approche est fondée sur la décomposition du problème en deux étapes. La première traite des rotations des avions, alors que la seconde s'intéresse à la réaccommodation des passagers.

Afin de construire des solutions de bonne qualité, les deux étapes s'inspirent au maximum du planning initial. La solution imite au mieux le planning non perturbé, ce qui limite les coûts opérationnels et fonctionnels.

Les algorithmes utilisés dans les deux étapes sont simples, ce sont des algorithmes fondés sur des flots dans des réseaux. La qualité des résultats obtenus reflète encore une fois le grand potentiel de ces modèles. De plus, toutes les contraintes peuvent être directement intégrées dans ces algorithmes de construction pour obtenir directement une solution réalisable.

Le schéma général de l'algorithme de résolution est donné dans la figure 7.1.

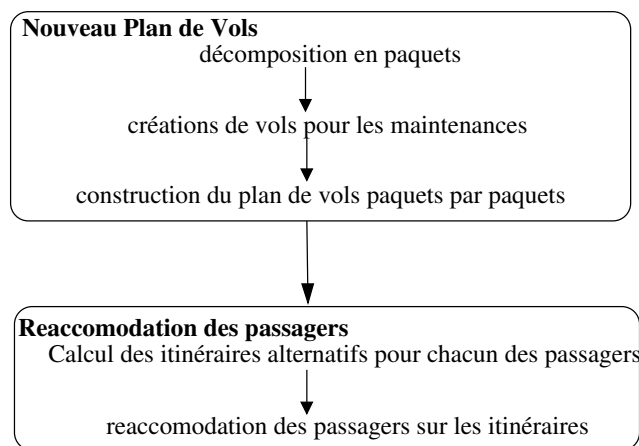


FIGURE 7.1 – Algorithme de résolution.

Dans les prochaines sections nous détaillons chacune des étapes. Nous rappelons qu'afin de faciliter la lecture, toutes les notations sont récapitulées dans l'annexe B.

7.2 Construction d'un nouveau plan de vols

Dans un premier temps, on intègre les perturbations dans le plan de vols initial en retardant les vols qui sont retardés et en annulant les vols annulés. Nous identifions ensuite tous les vols fixes : il s'agit des vols dont l'heure de départ était prévue avant le début de la période de recouvrement dans le planning initial. Même si après un retard imposé par une perturbation ces vols se retrouvent dans la période de recouvrement, ils ne peuvent pas être modifiés. On ne peut ni retarder, ni annuler, ni changer l'appareil qui opère un vol fixe. On peut d'ores et déjà ôter les capacités aéroportuaires utilisées par ces vols. Il n'y a plus aucune marge de manœuvre sur ces vols, dans la suite on s'intéresse donc uniquement aux vols non fixes.

Nous allons alors réaffecter les avions à ces vols non fixes, et construire ainsi un nouveau plan de vols. La solution qui consiste à annuler tous les vols non fixes respecte toutes les contraintes, sauf éventuellement les contraintes de maintenances. Dans des cas simples,

l'utilisation des vols non fixes suffit à amener les avions à leur maintenance. Pour des instances plus complexes, il faut créer de nouveaux vols pour amener les avions à leur aéroport de maintenance. Plus de détails à ce sujet sont donnés dans la section 7.2.3.

Pour des raisons à la fois techniques et de temps de calcul, la construction du nouveau plan de vols ne se fait pas simultanément sur toutes les ressources. Nous avons décomposé le problème en reconstruisant le plan de vols sur des plus petits paquets, un paquet étant composé d'un certain nombre d'avions et d'un certain nombre de rotations à faire opérer par ces avions. Cette décomposition est présentée dans la section 7.2.2. La reconstruction du plan de vols sur un paquet utilise une méthode générique qui est présentée dans la section 7.2.1.

7.2.1 Schéma général et modélisation

La réaffectation des appareils aux vols est faite en utilisant une méthode générique, qui réassigne des appareils à des vols en respectant les contraintes liées aux appareils. Elle produit une solution qui sera a priori de faible coût. Plus précisément la méthode traite le problème suivant.

ENTRÉES : – une fenêtre de temps débutant au temps 0 et finissant au temps TE ;
– un ensemble K d'avions ayant tous le même temps de réengagement (τ^k pour $k \in K$) et de la même famille (du même modèle par exemple), avec pour chaque avion k son aéroport de disponibilité AD^k et son heure de disponibilité TD^k ;
– un ensemble F de vols (rotations) à assigner aux avions, pour chaque vol f on connaît l'aéroport de départ AD_f , l'aéroport d'arrivée AA_f , l'heure initiale de départ TD_f et l'heure initiale d'arrivée TA_f ;
– un ensemble D de retards possibles pour retarder davantage certaines rotations ;
– une fonction coût c_{fd}^k donnant le coût d'affecter l'avion k au vol f retardé de d minutes. Et la fonction coût c_f donnant le coût d'annulation du vol f .

CONTRAINTES : – respecter les temps de réengagement des appareils,
– respecter les maintenances et le nombre maximal d'heures de vol autorisé avant maintenance de l'avion k , noté $MaxFlightHours^k$;
– ne pas utiliser un avion indisponible (perturbation indisponibilité d'appareil) ;
– respecter les capacités de décollage et d'atterrissage des aéroports.

SORTIE : un plan de vols respectant toutes les contraintes et minimisant le coût total de réaffectation plus le coût d'annulation.

Nous avons formalisé ce problème via un programme linéaire à variables entières fondé sur un modèle de multiflot. Ceci nous permet à la fois d'exploiter la structure de réseau, et d'intégrer des contraintes spécifiques liées à l'application.

Le temps est intégré dans le réseau via un réseau espace-temps. Le problème est représenté, avec une discrétisation du temps dans la figure 7.2. Le pas de la discrétisation du temps, Δ_t , est un des paramètres flexibles : trop grand on perd en performance, trop petit la taille des modèles explose.

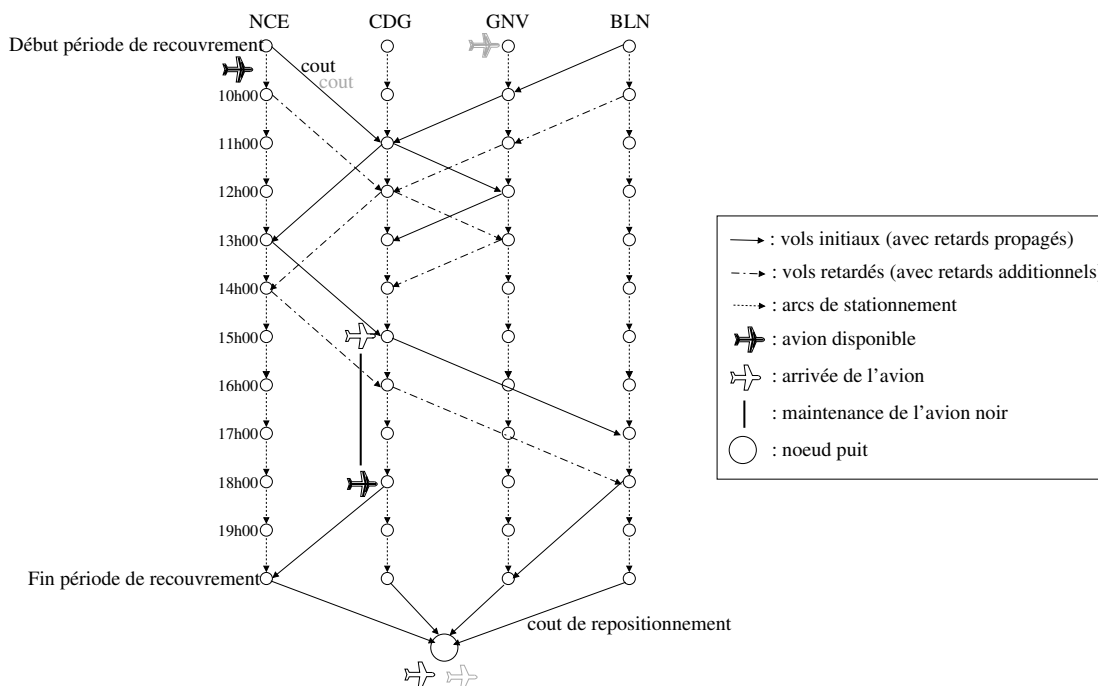


FIGURE 7.2 – Représentation graphique de la réaffectation des appareils.

Le réseau espace-temps est enrichi par différents noeuds spécifiques. Il comporte les noeuds suivants.

1. Noeuds classiques : la majorité de noeuds correspondent à un aéroport spécifique à un temps donné.
2. Noeuds de disponibilité des avions : chaque avion est disponible à un certain aéroport à un instant de la période de recouvrement donné.
3. Noeuds d'arrivée des avions : des noeuds puits spécifiques pour que les avions concernés respectent des contraintes de maintenance. Dans l'exemple de la figure 7.2, l'avion noir doit être à son aéroport de maintenance *CDG* au plus tard à 15h00.
4. Noeud puits : Un noeud est ajouté à la fin de la période de recouvrement, c'est un aéroport artificiel de destination des avions qui garantit que les avions ne sont plus en vols à la fin de la période.

Les arcs du réseau lient les différents noeuds. Ces arcs sont de différentes natures.

1. Arcs de vols : Les vols prévus sont représentés par des arcs reliant l'aéroport de départ à l'heure de départ à l'aéroport d'arrivée à l'heure d'arrivée plus le temps de réengagement. Les temps de réengagement sont inclus dans le temps de vol, ainsi deux vols peuvent se succéder.
2. Arcs de stationnements : Les arcs de stationnements (arcs verticaux) correspondent à l'attente d'avions dans un aéroport pendant un certain temps. Le flot sur un tel arc est le nombre d'avions stationnés à un aéroport. Dans notre problème, ces arcs

7.2 Construction d'un nouveau plan de vols

ne sont pas limités en capacité. De plus le stationnement n'induit ni de coût ni de revenus supplémentaires.

3. Arcs de retards : Les arcs de retard reflètent la stratégie de retards supplémentaires. Ce sont des arcs parallèles aux arcs de vols, vols correspondants à ces mêmes vols retardés des délais potentiels issus de D . Dans l'exemple, on a envisagé un seul retard supplémentaire de 1 heure.

Le problème de la réaffectation des appareils revient à calculer un flot de coût minimal avec contraintes additionnelles dans ce réseau espace-temps.

La **fonction objectif** consiste à minimiser les coûts induits par les vols annulés c_f , et les coûts résultant de l'opération d'un vol avec un retard ou un appareil différent de l'appareil initial c_{fd}^k .

Les variables de décisions, toutes binaires, sont les suivantes :

- $x_{fd}^k = 1$ si l'avion k opère le vol f avec un retard de d minutes.
- $X_f = 1$ si le vol f est annulé (opéré par aucun appareil).
- $y_{it}^k = 1$ si l'avion k reste stationné dans l'aéroport i entre les temps t et $t + \Delta_t$.

Le modèle à résoudre s'écrit comme suit.

$$\text{Minimiser : } \sum_{\substack{k \in K \\ f \in F, d \in D}} c_{fd}^k x_{fd}^k + c_f X_f, \quad (7.1)$$

Les contraintes (7.3) et (7.4) sont les contraintes classiques de **conservation du flot** et conditions aux nœuds sources et puits. La contrainte (7.2) garantit qu'un vol est opéré au plus une fois, soit à l'heure soit avec un retard.

$$\sum_{k \in K} \sum_{\substack{d \in D \\ TA_f + d \leq TE}} x_{fd}^k \leq 1, \quad \forall f \in F, \quad (7.2)$$

$$\sum_{\substack{f \in F, d \in D \\ AA_f = j \\ TA_f + d = t - \tau^k}} x_{fd}^k + y_{j, t - \Delta_t}^k - \sum_{\substack{f \in F, d \in D \\ AD_f = j \\ TD_f + d = t}} x_{fd}^k - y_{j, t}^k = \begin{cases} -1 & \text{si } j = AD^k, \\ 0 & \text{sinon,} \end{cases} \quad \begin{array}{l} \forall j \in A, \\ \forall k \in K, \\ \forall t \leq TE - \tau^k, \end{array} \quad (7.3)$$

$$\sum_{k \in K} \sum_{\substack{t \in T \\ t \leq \tau^k}} \sum_{\substack{f \in F, d \in D \\ AA_f = j \\ TA_f + d = TE - \tau^k + t}} x_{fd}^k + \sum_{k \in K} y_{f, TE - \Delta_t}^k \geq |K|, \quad (7.4)$$

Les **contraintes de capacité aéroportuaire** (7.5) et (7.6) lient différents nœuds et vols. Si la capacité à un aéroport est limitée, il faut choisir certains vols et en annuler d'autres.

Ces contraintes liantes renforcent la difficulté de résolution du problème.

$$\sum_{k \in K} \sum_{\substack{f \in F, d \in D \\ AD_f = i \\ h \leq TD_f + d < h+1}} x_{fd}^k \leq CapTakeOff[i, h], \quad \forall i \in A, \forall h \in H, \quad (7.5)$$

$$\sum_{k \in K} \sum_{\substack{f \in F, d \in D \\ AA_f = j \\ h \leq TA_f + d < h+1}} x_{fd}^k \leq CapLanding[j, h], \quad \forall j \in A, \forall h \in H, \quad (7.6)$$

Les **contraintes de maintenance** (7.7) et (7.8) sont de deux types. La première garantit qu'un avion soit dans l'aéroport de sa maintenance pendant toute la durée de celle-ci. La seconde limite le temps de vol dont un avion dispose avant de se rendre à sa maintenance. Les **contraintes d'indisponibilités** (7.9) forcent un avion indisponible à rester cloué au sol pendant tout le temps de son indisponibilité, aucun vol ne peut être affecté à un avion indisponible.

$$y_{AM^k t}^k = 1, \quad \forall k \in KM, \forall t \in [TMB; TME], \quad (7.7)$$

$$\sum_{\substack{f \in F, d \in D \\ TA_f + d \leq TMB^k}} (TA_f - TD_f) x_{fd}^k \leq MaxFlightHours^k, \quad \forall k \in KM, \quad (7.8)$$

$$\sum_{i \in A} y_{it}^k = 1, \quad \forall k \in KU, \forall t \in [TUB; TUE], \quad (7.9)$$

Toutes les **variables** sont binaires.

$$x_{fd}^k \in \{0, 1\}, \quad \forall k \in K, \forall f \in F, \forall d \in D, \quad (7.10)$$

$$X_f = 1 - \sum_{\substack{k \in K, \\ d \in D}} x_{fd}^k \in \{0, 1\}, \quad \forall f \in F, \quad (7.11)$$

$$y_{jt}^k \in \{0, 1\}, \quad \forall k \in K, \forall j \in A, \forall t \in T. \quad (7.12)$$

Du fait notamment de la décomposition du temps, ce modèle est très vite amené à être de très grande taille. Ceci compromet non seulement sa résolution rapide, mais même son chargement en mémoire.

La plus petite instance, notée A01 comporte $|A| = 35$ aéroports, $|K| = 85$ avions, 608 vols dont 400 sont non fixes, sur une période de 16 heures, soit 192 intervalles de temps de 5 minutes. Le nombre de retards additionnels dépend de la politique de retard. Au minimum, on autorise aucun retard additionnel, i.e. seulement le retard de 0 minutes. En pratique on utilise entre 1 et 20 retards additionnels. Dans le tableau 7.1, un ordre de grandeur de la taille du modèle de réaffectation des appareils pour ce type d'instance est donné. Ce tableau permet de mettre en évidence le grand nombre de contraintes et de variables binaires. Le modèle comporte surtout des contraintes de flot qu'il faut écrire à chaque nœud du réseau espace-temps. La discrétisation du temps fait donc exploser la taille du modèle.

Concevoir et implémenter ce modèle en C++ a été une de mes tâches dans le challenge.

7.2 Construction d'un nouveau plan de vols

	tous les modèles 10 retards possibles	les A320 10 retards possibles	les A320 pas de retard 1 retard possible
vols non fixes	400 vols	100 vols	100 vols
variables binaires	340 000	85 000	8 500
contraintes	550 000	550 000	550 000

TABLE 7.1 – Taille du modèle pour différentes politiques de retard.

7.2.2 Décomposition en paquets.

Une instance comporte entre 608 et 2844 rotations et entre 85 et 255 avions. Le changement d'un avion qui opère un vol ne peut se faire que dans la même famille. Nous avons donc décomposé la construction du plan de rotation, famille par famille d'appareils. Ceci permet notamment de respecter la contrainte de changement d'avion.

Cette première décomposition n'est pas suffisante pour toutes les instances. La famille *AirbusSmall* contient encore beaucoup d'avions (jusqu'à 144 pour une instance). Nous décomposons donc plus finement la construction du plan de rotation par modèle d'appareils. Ceci présente un avantage : tous les avions d'un même modèle ont le même temps de réengagement, on peut donc facilement l'intégrer au temps de vols, indépendamment de l'avion qui opérera le vol. Cela simplifie l'implémentation.

Malheureusement, certains modèles comportant beaucoup d'avions, produisent encore des programmes mathématiques trop grands (il y a 60 appareils du modèle A320 dans l'instance B10, ces avions opérant près de 2000 vols). Notre algorithme ne peut produire de réaffectation pour tous les avions d'un tel modèle pris d'un bloc en moins de 1 minute, ce qui compromet l'obtention d'une solution finale en moins de 10 minutes. Nous proposons donc une décomposition plus fine, en prenant des sous-ensembles d'appareils de même modèle et les rotations correspondantes pour constituer un paquet. En pratique, nous avons fait des tests sur les instances fournies pour trouver le bon compromis entre temps de résolution et taille des paquets. Cela a été ma seconde tâche dans le challenge.

On réaffecte alors de façon séquentielle les appareils aux vols, paquet par paquet. Les paquets sont ordonnés : on traite en priorité ceux qui comportent des avions ayant des maintenances, i.e. des contraintes inviolables. En effet, c'est dans ces paquets que la réaffectation des avions est la plus délicate et risque de produire une solution non réalisable. Traiter ces paquets en priorité permet de disposer encore de capacités aéroportuaires, notamment pour atterrir à l'aéroport de maintenance. On ajoute quelques nouveaux vols dans ces paquets, créés spécifiquement pour pouvoir amener rapidement un avion à son aéroport de maintenance. Plus de détails sur ces vols sont donnés dans la section suivante.

7.2.3 Création de nouveaux vols pour les maintenances.

Considérons un paquet qui comporte au moins un avion nécessitant une maintenance. Les vols de ce paquet permettraient, avant perturbation, d'amener les avions à leur aéroport de maintenance. Cependant, du fait des perturbations, notamment des retards ou des annulations de ces vols et des réductions de capacités aéroportuaires, il se peut que l'on ne

puisse pas amener un avion à sa maintenance avec seulement les vols du paquet. On ajoute donc de nouveaux vols, créés spécifiquement pour amener ces avions à leur aéroport de maintenance. En théorie, le choix de ces vols est sans doute un problème difficile. En effet, il faut jongler avec les capacités aéroportuaires encore disponibles, le temps de vol encore disponible, la distance maximale franchissable (ou rayon d'action) de l'appareil et surtout arriver à temps à l'aéroport de la maintenance. Avec un peu de malice, on peut construire des instances, qui atteignent les limites d'heuristiques simples de créations de vols pour les maintenances.

La figure 7.3 donne un exemple de rotation d'un avion qui a une maintenance. Le plan de vols de cet appareil est perturbé par de très fortes réductions de capacités aéroportuaires dans certains aéroports.

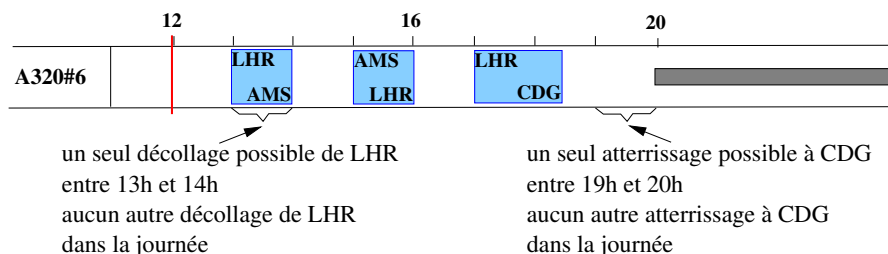


FIGURE 7.3 – Plan de vols initial et perturbation.

Pour amener l'avion à sa maintenance, les faibles capacités aéroportuaires rendent impossible la création d'un vol direct de l'aéroport de disponibilité *LHR* vers l'aéroport de maintenance *CDG*. Une solution pour amener l'avion à sa maintenance, est de le faire transiter par un troisième aéroport. Cette solution est présentée dans la figure 7.4.

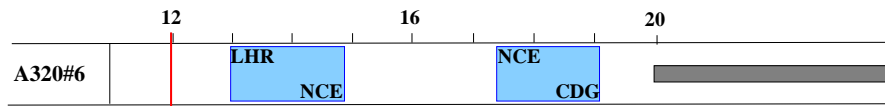


FIGURE 7.4 – Création de vols pour amener un avion à sa maintenance.

Nous avons proposé un algorithme (algorithme 7.1) simple de création de vols pour se rendre aux maintenances. Le principe de cet algorithme est de créer des vols directs et des vols avec une escale de leur aéroport de disponibilité vers l'aéroport de maintenance.

Algorithme 7.1 : Creation de nouveaux vols pour les maintenances avec zéro ou une escale.

```

1 pour avion  $k$  avec maintenance ( $k \in KM$ ) faire
2   | Calculer  $AD^k$  et  $TD^k$ ;
3   | Lire  $AM^k$ ,  $TMB^k$  et  $MaxFlightHours^k$ ;
4 fin
5 si  $Dist(AD^k, AM^k) \leq MaxDist^k$  et qu'il y a capacité aéroportuaire encore
   | disponible) alors
6   | créer un vol direct de  $AD^k$  à  $AM^k$  le plus tôt possible et arrivant avant le début
   | de la maintenance  $TMB^k$ ;
7 sinon
8   | pour aéroports  $a \in A$  faire
9     | créer (si possible) un vol direct  $f$  de  $AD^k$  à  $a$  le plus tôt possible;
10    | créer (si possible) un vol direct  $f'$  de  $a$  à  $AM^k$  dès que possible après  $f$ ;
11    fin
12 fin

```

7.2.4 Coûts de réaffectation

La fonction coût est définie de sorte à ce que le plan de vols obtenu reste proche du plan de vols initial. Les passagers pourront alors être réaffectés à priori sans trop de changement, ce qui limitera leur désagrément. La précision de cette fonction coût est donc fondamentale pour la qualité des solutions finale. Cette fonction intègre à la fois des coûts opérationnels et des coûts fonctionnels.

Le coût d'annulation c_f du vol f est égal au coût d'annulation de tous les passagers initialement à bord de ce vol. Ce coût est composé du coût des billets payés par les passagers à rembourser, des coûts de compensation financière et des coûts de pénalité.

Le coût c_{fd}^k de réaffecter l'avion k au vol f retardé de d minutes est composé de deux parties. D'une part, le changement de l'avion opérant le vol peut entraîner un changement de configuration et de ce fait des déclassements voire des annulations si le nouvel avion est plus petit. Ces conséquences sont intégrées dans la fonction coût. D'autre part, la fonction coût contient les coûts de retard.

Idéalement, la fonction coût devrait également prendre en compte l'impact qu'une perturbation a sur les correspondances. En effet, en pratique, les passagers avec correspondances engendrent une grande part des revenus. Cependant, n'ayant pas trouvé d'approximation linéaire satisfaisante de ces coûts, nous ne les avons pas intégrés dans notre formulation.

7.3 Réacommodation des passagers

Le but de la deuxième étape de l'algorithme est de réacommoder les passagers sur les vols finalement opérés. Pour cela nous commençons par calculer un ensemble d'itinéraires alternatifs pour chacun des passagers. La réacommodation se fait par le calcul d'un flot de coût minimal avec contraintes de capacité. Pour cela nous avons choisi une formulation chemin du problème de flot.

7.3.1 Itinéraires alternatifs pour la réacommodation

Pour chaque passager p on calcule un ensemble Q^p d'itinéraires alternatifs. Chacun de ces itinéraires respecte les contraintes de réacommodation : un itinéraire alternatif a la même origine et la même destination que l'itinéraire initial, l'heure de départ est postérieure à l'heure de départ initiale, le retard à l'arrivée n'excède pas le retard maximal $MaxDelayPassenger^p$. La classe de confort (F/B/E) est intégrée dans ces itinéraires, on envisage toutes les combinaisons de classes possibles sur les différents tronçons d'un itinéraire, cela multiplie le nombre d'itinéraire par 3 pour chaque tronçon. Lors de la construction de ces itinéraires alternatifs, on calcule également le coût C_q^p de réacommoder le passager p sur l'itinéraire alternatif q . Ce coût est composé des coûts de déclassement, des coûts de retard et des coûts de désagréments.

La génération des itinéraires alternatifs pour les passagers, ainsi que celle des coûts correspondants, sont donnés par l'algorithme 7.2.

Algorithme 7.2 : Calcul des différents itinéraires possibles pour les passagers.

```

1 pour passagers  $p$  in  $P$  faire
2   Lire  $AD^p$ ,  $AA^p$ ,  $TD^p$ ,  $TA^p$  et les aéroports d'escalés;
3   Calculer des itinéraires alternatifs de  $AD^p$  à  $AA^p$  (recherche en profondeur);
4   Affecter toutes les combinaisons de classes possibles à ces itinéraires;
5   Calculer les coûts  $C_q^p$ ;
6   Ajouter ces itinéraires à  $Q^p$ ;
7   pour aéroport  $a \in A$ ,  $a$  hub faire
8     Calculer des itinéraires alternatifs de  $AD^p$  à  $AA^p$  via le hub  $a$ ;
9     Affecter toutes les combinaisons de classes possibles à ces itinéraires;
10    Calculer les coûts  $C_q^p$ ;
11    Ajouter à  $Q^p$ ;
12  fin
13 fin

```

Dans notre étude, nous avons générés ces itinéraires une fois pour toute. L'ensemble de ces itinéraires est alors utilisé pour réacommoder les passagers, c'est la base des variables chemin du problème de flot présenté dans le paragraphe suivant.

7.3.2 Modélisation

Nous présentons dans cette section le MIP de flot en formulation chemins, qui modélise le problème de réacommodation des passagers sur les vols opérés.

La variable entière z_q^p correspond au nombre de passagers de la réservation p , réacommodés sur l'itinéraire q . Le coût de cette réacommodation est noté C_q^p . Le nombre total de passagers de la réservation p est noté n^p . Si un passager de la réservation p n'est pas réacommodé, il voit son voyage annulé. Le coût d'une telle annulation est noté CC^p .

On note δ_{fq}^{kcl} le booléen qui vaut 1 si le vol f opéré par l'avion k en classe de confort cl est dans l'itinéraire q . $cap^k(cl)$ est le nombre de siège (capacité) de l'avion k en classe de confort cl .

Le modèle linéaire en nombres entiers à résoudre est le suivant.

$$\text{Minimiser : } \sum_{p \in P} \sum_{q \in Q^p} C_q^p z_q^p + \sum_p CC^p \left(n^p - \sum_{q \in Q^p} z_q^p \right), \quad (7.13)$$

s.c.

$$\sum_{p \in P} \sum_{q \in Q^p} \delta_{fq}^{kcl} z_q^p \leq cap^k(cl), \quad \forall f \in F, \forall cl \in CL, \quad (7.14)$$

$$\sum_{q \in Q^p} z_q^p \leq n^p, \quad \forall p \in P, \quad (7.15)$$

$$z_q^p \in \mathbb{N}. \quad (7.16)$$

La fonction objectif (7.13) comporte deux termes, le premier concerne les coûts de réacommodation des passagers, le second les coûts liés à l'annulation des passagers non réacommodés. La contrainte (7.14) assure que la capacité de l'avion k est respectée, et ce dans chaque classe de confort. Le nombre total de passagers d'une réservation p réacommodés est limité par le nombre initial des passagers de p grâce à la contrainte (7.15). Toutes les autres contraintes sont déjà incluses dans la génération des itinéraires alternatifs.

En théorie, si l'on génère tous les itinéraires alternatifs possibles, la résolution de ce modèle nous donne la réacommodation optimale des passagers sur les vols opérés. En pratique, nous avons fait le choix de ne générer que de bons candidats pour ces itinéraires. Il serait plus performant d'utiliser un algorithme à base de génération de colonnes, qui génère dynamiquement de bons nouveaux itinéraires selon les besoins. Mais de toute façon, la qualité de la solution obtenue après la réacommodation des passagers dépend des vols opérés après la première étape de l'algorithme de résolution. Il faudrait donc pouvoir revenir sur ces choix lors de la réaffectation des passagers, afin d'améliorer davantage le plan de vols par rapport aux besoins des passagers.

Chapitre 8

Résultats expérimentaux

Sommaire

8.1	Instances	139
8.2	Temps de calcul	140
8.3	Qualité des solutions obtenues	141

Dans ce chapitre, nous présentons nos résultats numériques. Ces résultats sont améliorés par rapport au programme fourni pour le challenge. En effet, en améliorant légèrement notre code, après la fin du challenge, nous avons réussi à obtenir des solutions réalisables sur toutes les instances.

Le challenge s'est déroulé en deux phases. Une première phase de qualification, entre février 2008 et septembre 2008, dans laquelle 29 équipes se sont lancées. Ces équipes se partagent en deux catégories : la catégorie junior avec des équipes qui comportent au plus un docteur et la catégorie senior avec des équipes avec au moins deux docteurs plus ou moins expérimentés. Quatorze pays des Pays-Bas à l'Australie, en passant par le Québec étaient représentés. A la fin des qualifications, en Septembre 2008, notre équipe a terminé première de la catégorie junior et quatrième au classement général sur les instances d'une base A "facile". Au total 11 équipes (7 de la catégorie senior et 4 de la catégorie Junior) ont été qualifiées et ont continué leur travail sur la base B "plus difficile" jusqu'en février 2009. Les résultats finaux ont été annoncés le 11 février 2009 lors du congrès annuel de la ROADEF, qui cette année là se déroulait à Nancy. Notre équipe a été classée deuxième de la catégorie junior et cinquième au classement général.

8.1 Instances

Lors du challenge, la société Amadeus a fourni trois jeux d'instances au fur et à mesure de l'avancée du challenge.

- le premier jeu d'instances (Base *A*), disponible dès le début du challenge utilisé pour arbitrer la sélection des candidats pour la finale ;
- le deuxième jeu d'instances (Base *B*) disponible à l'issue de la phase de qualification ;

- les derniers jeux d’instances (Bases XA , XB et X) divulgués après l’annonce des résultats finaux.

	vols	appareils	itinéraires	passagers
jeu A et XA	600	85	2 000	36 000
	1 200		4 000	100 000
jeu B et XB	2 500	251	11 500	250 000
jeu X	6 500	600	30 000	700 000

TABLE 8.1 – Taille des instances.

Dans cette section, nous présentons nos résultats numériques obtenus sur ces jeux d’instances. Un ordre de grandeur de la taille des instances dans les différents jeux est donné dans le tableau 8.1. Les instances X01 à X04 sont des instances de très grande taille qui n’ont pas été prises en compte pour le classement (un seul programme a proposé des solutions réalisables sur ces instances). Le classement final des candidats aux challenge ROADEF’2009 a été fait sur les instances B , XA et XB (18 instances différentes). Ces instances comportent 2844 rotations opérée par 255 avions et environ 11,000 itinéraires qui correspondent à plus de 200,000 passagers. Pour chacune de ces instances, on disposait de 10 minutes de temps de calcul sur un ordinateur avec un processeur AMD Turion64x2 possédant 2Go de RAM. Les systèmes d’exploitation supportés sont Windows XP Professional SP2 (32 bits) et Fedora 9 (64 bits). Nous avons opté pour Windows XP Professional SP2 (32 bits).

8.2 Temps de calcul

Le tableau 8.2 donne les temps de calcul de chacune des deux parties de notre algorithme. Pour chacune des instances, on donne le temps de calcul nécessaire à la construction du nouveau plan de vols dans la colonne *Avions*, et le temps nécessaire pour la réacommodation des passagers dans la colonne *Passagers*. La dernière colonne donne le temps de calcul total (en secondes) nécessaire à l’obtention de notre solution. Les temps de chargement en mémoire des instances, et d’écriture de la solution dans les fichiers sorties sont négligeables (moins de quelques secondes).

D’après les temps du tableau 8.2, environ 20% du temps de calcul est utilisé pour reconstruire un nouveau plan de vols, et 80% est utilisé pour la réacommodation des passagers. A l’inverse, sur des instances de petite taille, comme $XA01$ et $XA03$, la réacommodation des passagers nécessite moins de 10 secondes de temps de calcul et c’est la reconstruction du plan de vols qui prend relativement plus de temps. Ceci s’explique par la taille des modèles MIP utilisés dans chacune des fonctions. La construction du plan de vols résout plusieurs petit MIP, alors que la réacommodation des passagers ne résout qu’un seul et gros PLNE. Pour de petites instances ce dernier programme reste lui aussi de petite taille.

8.3 Qualité des solutions obtenues

Instance	Avions(%)	Passagers (%)	Temps total (s)
B01	17,71	82,29	336,985
B02	19,68	80,32	302,641
B03	18,17	81,83	329,109
B04	18,25	81,75	322,672
B05	19,89	80,11	465,578
B06	19,49	80,51	367,188
B07	19,95	80,05	357,782
B08	18,04	81,96	393,594
B09	17,44	82,56	406,203
B10	19,38	80,62	483,141
XA01	93,07	6,93	18,032
XA02	29,74	70,26	219,000
XA03	93,21	6,79	18,656
XA04	26,68	73,32	191,781
XB01	18,51	81,49	320,750
XB02	19,9	80,1	466,797
XB03	20,11	79,89	353,875
XB04	19,82	80,18	475,250

TABLE 8.2 – Détails des temps de calcul.

8.3 Qualité des solutions obtenues

Le tableau 8.3 présente les résultats obtenus sur les instances utilisées pour le classement. Les colonnes *Meilleure* et *Pire* correspondent au coût de la meilleure (resp. pire) solution parmi l'ensemble des solutions des participants. La colonne *Résultat* correspond au coût de la solution obtenue par notre programme. Le classement est effectué sur la base du score normalisé, présenté dans la dernière colonne. Ce score est obtenu par le calcul suivant :

$$Score = \frac{Pire - Resultat}{Pire - Meilleure}$$

Une modification mineure du code délivré pour le challenge nous a permis d'obtenir des solutions réalisables sur toutes les instances. Pour cela nous avons simplement ajouté une nouvelle contrainte dans le modèle de réaffectation des avions, sans changer les autres parties du code. Cette nouvelle version nous permet d'avoir un score normalisé moyen de 87%. La contrainte ajoutée permet la gestion des vols qui changent de jours du fait d'un retard trop important. Notre algorithme a donc le mérite de produire des solutions réalisables et de bonne qualité en moins de 10 minutes. Ceci illustre encore une fois la puissance des techniques simples à bas de flots et multiflots. La simplicité de la méthode la rend très flexible : si besoin, il est facile d'intégrer de nouvelles contraintes.

Instance	Meilleure	Pire	Résultat	Score
B01	971 182,50	43 169 547,75	2 480 629,85	0,96
B02	1 220 708,30	31 420 941,20	5 741 677,20	0,85
B03	1 007 565,70	47 509 155,15	2 579 411,00	0,97
B04	1 101 394,80	46 400 734,65	2 575 466,95	0,97
B05	9 653 780,05	94 278 109,15	29 235 529,65	0,77
B06	3 218 000,10	66 101 253,95	8 323 534,40	0,92
B07	5 039 744,20	56 785 261,80	16 318 727,85	0,78
B08	3 509 318,00	62 391 786,00	8 211 402,55	0,92
B09	3 967 344,70	68 668 311,00	10 101 024,65	0,91
B10	34 523 605,00	124 900 519,50	66 458 200,35	0,65
XA01	116 195,20	7 486 622,70	264 756,30	0,98
XA02	1 475 322,10	38 313 615,30	3 582 627,75	0,94
XA03	285 287,05	10 092 302,00	604 065,45	0,97
XA04	4 112 262,60	13 937 165,80	10 487 502,00	0,35
XB01	1 352 823,05	105 894 332,10	3 156 666,95	0,98
XB02	11 297 822,20	200 281 943,50	29 259 788,40	0,90
XB03	6 463 354,30	135 863 963,60	12 609 345,75	0,95
XB04	34 331 225,80	240 102 702,80	63 522 137,85	0,86
Moyenne				0,87

TABLE 8.3 – Résultats numériques.

Le tableau 8.4 donne le classement finale des différentes équipes finalistes. La version de notre code donnée pour le challenge nous classait cinquième au classement général, deuxième de la catégorie junior, avec le score normalisé moyen de 70,62% dû à quelques solutions non réalisables. La légère modification du code, qui nous permet d'obtenir des solutions réalisables sur toutes les instances nous classerait troisième au classement général et premier junior !

Equipe	Catégorie	Rang (rang dans la catégorie)	Score Moyen (%)
Bisaillon, Cordeau, Laporte, Pasin	Senior	1(1)	95,90
Hanafi, Wilbaut, Mansi, Clautiaux	Senior	2(2)	92,73
Darlay, Kronek, Schrenk, Zaourar	Junior	3(1)	87,00
Acuna-Agost, Michelon, Feillet, Gueye	Senior	3(3)	74,26
Eggermont, Firat, Hurkens, Modelski	Junior	4(1)	72,01
Darlay, Kronek, Schrenk, Zaourar	Junior	5(2)	70,62
Peekstok, Kuipers	Senior	6(4)	70,31
Jozefowicz, Mancel, Mora-Camino	Senior	7(5)	64,02
Dickson, Smith, Li	Junior	8(3)	42,02
Eggenberg, Salani	Junior	9(4)	20,48

TABLE 8.4 – Classement final.

Synthèse

Dans cette troisième partie, nous avons étudié le problème de gestion de perturbation dans le domaine aérien, sujet du challenge ROADEF'2009. Nous avons d'abord proposé une présentation illustrée du problème. Nous avons replacé le problème dans le contexte de la planification dans le domaine aérien et présenté un état de l'art. En raison de la taille des instances et du temps de calcul restreint de 10 minutes, nous avons proposé et implémenté une heuristique à deux phases pour résoudre le problème. Chacune de ces phases est modélisée via un modèle de multiflot avec contraintes dans un réseau, respectivement avec un modèle à base d'arcs ou de chemins.

Nous avons tenté de relever le défi posé en résolvant séquentiellement ces modèles de multiflot avec contraintes en utilisant un solveur de MIP. Notre méthode construit ainsi directement une solution réalisable et optimisée. Simple, rapide, et flexible, notre algorithme nous a permis d'atteindre la deuxième place dans la catégorie junior. Ceci illustre encore une fois le potentiel des méthodes de flot.

La première place de la catégorie junior n'est pas passée loin... et notre méthode pourrait être encore améliorée, à la fois en temps et en qualité. En effet, un algorithme de génération de colonnes pour la réacommodation des passagers permettrait de manipuler seulement les itinéraires utiles dans le modèle chemin. Une deuxième amélioration, non moindre, mais plus difficile à mettre en place, serait d'avoir une boucle de retour de la réacommodation des passagers afin d'améliorer progressivement la solution.

Conclusions et perspectives

Dans cette thèse, nous nous sommes intéressés à deux problèmes industriels dans le domaine du transport. Le premier est un problème de conception de réseau de service avec gestion de ressources pour un transport de fret. Le second est le problème de gestion de perturbation dans le domaine aérien, sujet du challenge ROADEF'2009. Dans les deux cas, il s'agit de problèmes pratiques difficiles qui comportent des contraintes complexes non standard. Nos contributions ont porté sur trois thèmes de Recherche Opérationnelle : combinatoire, modélisation et résolution.

Dans la première partie, nous avons étudié la complexité de problèmes combinatoires purs, sous-problèmes au cœur de nos problèmes industriels. Nous avons d'abord étudié les problèmes polynomiaux de flots dans les réseaux de distribution et plus particulièrement le problème de transport de base. Nous avons proposé quelques extensions, notamment autour du paradoxe de transport, et caractérisé les réseaux bien équilibrés. Puis nous avons étudié la complexité de sous-problèmes pour comprendre d'où vient la difficulté des problèmes. Une de nos contributions majeures a été de démontrer que, même dans le cas le plus simple le problème de transport de base avec coût fixe est déjà difficile. Nous avons également mis en évidence la difficulté engendrée par les contraintes pratiques de gestion des véhicules. Ces problèmes s'interprètent comme des problèmes de graphes, certains d'entre-eux sont encore ouverts à ce jour, comme notamment le problème de partitionnement des arcs d'un graphe en chemin de longueur comprise entre $a \geq 3$ et $b > a$.

La deuxième partie, cœur de cette thèse, était consacrée au problème de conception d'un réseau de service avec gestion de flotte modélisant un problème industriel de planification d'un service régulier de transport de fret. Des flux de marchandises doivent être déplacés sur un réseau physique (route, chemin de fer, réseau maritime, réseau aérien) en mobilisant les véhicules d'une flotte donnée. Nous avons présenté différentes formulations MIP du problème. En effet, les décisions concernant les trajets des véhicules peuvent être prises soit au niveau des arcs du réseau, soit directement sur des chemins morceaux d'un cycle d'un véhicule. De même, les variables déterminant le flot des marchandises peuvent être définies sur des arcs ou des chemins du réseau. Nous avons ainsi obtenu 4 formulations : arc-arc, cycle-arc, arc-chemin, cycle-chemin. Nous les avons comparé analytiquement et expérimentalement sur de petites instances. Il ressort de notre étude que les formulations à base de cycles sont très prometteuses, elles procurent des relaxations linéaires de bonne qualité. A la fin de cette deuxième partie, nous avons ébauché un algorithme à base de génération de colonnes capable de résoudre le problème pour de plus grandes instances.

Nous avons également présenté quelques tests préliminaires pour évaluer le potentiel d'un tel algorithme ainsi que ses limites. Une perspective naturelle serait d'implémenter de façon performante des heuristiques à base de génération de colonnes pour résoudre des instances réelles. Des contraintes industrielles supplémentaires, telles des fenêtres de temps sur les collectes et livraisons permettraient de limiter la combinatoire dans le processus de génération des colonnes. Le pas de discrétisation du temps, est alors également un paramètre à contrôler, trop petit il fait exploser la taille des modèles, trop grand il ne permet pas d'envisager suffisamment de services et entraîne beaucoup de temps mort.

Enfin, la dernière partie présente le travail réalisé pour répondre au challenge ROA-DEF'2009, challenge auquel j'ai participé avec trois autres doctorants et ex-doctorants du laboratoire G-SCOP entre février 2008 et février 2009. Le sujet proposé par la société Amadeus concerne la gestion de perturbations dans le domaine aérien. J'ai présenté notre approche de résolution qui reconstruit des solutions respectant toutes les contraintes et de bonne qualité en résolvant des modèles de multiflot avec contraintes additionnelles. Notre méthode a permis d'implémenter rapidement une heuristique ad hoc efficace dans le cadre des conditions du challenge. On peut la voir comme une méthode qui construit une première solution. Grâce à notre travail nous avons réussi à remporter la deuxième place de la catégorie junior ! Une piste d'amélioration possible serait la remise en cause, après la réacommodation des passagers, des choix effectués dans la reconstruction du plan de vols. Il faudrait donc insérer notre heuristique dans un processus itératif plus complexe.

C'est ainsi que nous avons contribué à améliorer la compréhension, la modélisation et la résolution de problèmes industriels complexes. La recherche de solutions encore meilleures nous réservera sûrement de belles surprises.

A Notations pour le problème de conception de réseau de service

Ensembles:

N = ensemble des nœuds du réseau espace temps, indexé par i ou j . Pour $i \in N$, on note $i = (N_i, T_i)$, N_i est le nœuds du réseau physique, T_i est le temps.

A = ensemble des arcs de services.

K = ensemble des demandes de transport (marchandises), indexé par k .

V = ensemble de tous les véhicules, indexé par v .

V_{type} = ensemble des véhicules de type $type$.

\mathcal{P}_v = ensemble des chemins, morceaux de cycles, potentiels pour le véhicule v .

Données et paramètres:

T = date de fin de période de planification.

τ = temps de réengagement des véhicules k .

t_{ijv} = temps de trajet entre les nœuds i et j du réseau espace temps.

CV_{ijv} = coût variable d'opérer le service (ij) avec le véhicule v .

CF_{ijv} = coût fixe d'ouverture du service (ij) avec le véhicule v .

CF_v^p = coût fixe d'ouverture de la route p avec le véhicule v .

P^k = profit unitaire pour la satisfaction d'une unité de demande k .

σ^k, d^k = terminal d'origine, respectivement de destination de la demande k .

w^k = quantité totale de la demande k .

$QuotaMin_v, QuotaMax_v$ = temps d'utilisation minimal et maximal du véhicule v , si il est utilisé.

u_{ijv} = capacité du véhicule v lorsqu'il opère le service (ij) .

$r_{ij}^p = 1$ si l'arc (ij) est dans le chemin p , 0 sinon. $b_{ijv}^k = \min\{u_{ijv}, w^k\}$.

Variables:

$\delta_v = 1$ si le véhicule v est utilisé.

x_{ijv}^k = quantité de marchandise de la demande k qui transite à bord du véhicule v sur (ij) aux horaires correspondants.

z^k = quantité totale de la demande k satisfaite sur une période de planification.

$y_{ijv} = 1$ si le véhicule v opère le service (ij) .

$g_v^p = 1$ si le véhicule v opère le chemin p .

variables duales α_{type} , σ^k , η_{ijv} , ρ_{ijv}^k et $\theta_{i'type}$.

B Notations pour le problème de gestion de perturbation dans le milieu aérien

Ensembles:

A = ensemble des aéroports, indexé par i ou j .

K = ensemble des avions, indexé par k . KM (respectivement KU) est le sous-ensemble des avions ayant une maintenance (respectivement une indisponibilité).

F = ensemble des vols, indexé par f .

D = ensemble de tous les retards additionnels possibles, indexé par d .

H = ensemble de toutes les heures, indexé par h .

T = ensemble de tous les instants (temps discrétisé en minutes) commençant au temps 0, finissant au temps TE (fin de la période de recouvrement), indexé par t .

P = ensemble de tous les itinéraires des passagers (agrégés), indexé par p .

CL = ensemble des différentes classes de confort business (B), first (F) et economic (E), indexé par cl .

Q^p = ensemble de tous les itinéraires alternatifs pour réacommoder un passager de l'itinéraire p , indexé par q .

Données et paramètres:

Δ_t = pas de discrétisation du temps (par exemple 5 minutes).

τ^k = temps de réengagement de l'avion k .

c_{fd}^k = coût d'opérer le vol f avec l'avion k et un retard supplémentaire de d minutes.

c_f = coût d'annulation du vol f .

$cap^k(cl)$ = nombre de siège (capacité) de l'avion k en classe de confort cl .

$AD_f, AA_f, AD^p, AA^p, AD^k, AA^k$ = aéroports de départ et d'arrivée du vol f , des passagers de l'itinéraires p , et de l'avion k respectivement.

TD_f, TA_f, TD^p, TA^p = heure de départ et d'arrivée du vol f et des passagers de l'itinéraire p respectivement.

$CapTakeOff[i, h]$ = nombre maximal de décollages à partir de l'aéroport i dans l'intervalle horaire $[h, h + 1[$.

$CapLanding[j, h]$ = nombre maximal d'atterrissages à l'aéroport j dans l'intervalle horaire $[h, h + 1[$.

TMB^k, TME^k = temps de début et de fin de la maintenance de l'avion k .

AM^k = aéroport de maintenance de l'avion k .

$MaxFlightHours^k$ = nombre maximal d'heures de vols avant le début de la maintenance pour l'avion k .

TUB^k, TUE^k = temps de début et de fin de l'indisponibilité de l'avion k .

n^p = nombre de passagers de l'itinéraire p .

$MaxDelayPassenger^p$ = retard maximal à l'arrivée autorisé pour les passagers de l'itinéraire p .

C_q^p = coût pour la réacommodation d'un passager de l'itinéraire p sur l'itinéraire q .

CC^p = coût d'annulation du voyage d'un passager de l'itinéraire p .

δ_{fq}^{kcl} = 1 si le vol f opéré par l'avion k dans la classe de confort cl est dans l'itinéraire q .

Variables de décision :

x_{fd}^k = 1 si l'avion k opère le vol f avec un retard de d minutes.

X_f = 1 si le vol f est annulé.

y_{it}^k = 1 si l'avion k reste stationné à l'aéroport i entre les temps t et $t + \Delta_t$.

z_q^p = nombre de passagers de l'itinéraire p réacommodés sur l'itinéraire q .

Bibliographie

- [Adlakha et Kowalski, 2003] ADLAKHA, V. et KOWALSKI, K. (2003). A simple heuristic for solving small fixed-charge transportation problems. *Omega*, 31(3):205–211.
- [Ahuja *et al.*, 1993] AHUJA, R. K., MAGNANTI, T. L. et ORLIN, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.
- [Andersen *et al.*, 2009a] ANDERSEN, J., CHRISTIANSEN, M., GRØNHAUG, R. et CRAINIC, T. G. (2009a). Branch-and-price for service network design with asset management constraints. Rapport technique 58, Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport, Canada.
- [Andersen *et al.*, 2009b] ANDERSEN, J., CRAINIC, T. G. et CHRISTIANSEN, M. (2009b). Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C*, 17(2):197–207.
- [Andersson, 2001] ANDERSSON, T. (2001). *The flight perturbation problem: operational aircraft rescheduling*. Thèse de doctorat, Linköping university, Sweden.
- [Andersson, 2006] ANDERSSON, T. (2006). Solving the flight perturbation problem with meta heuristics. *Journal of Heuristics*, 12(1):37–53.
- [Andersson et Värbrand, 2004] ANDERSSON, T. et VÄRBRAND, P. (2004). The flight perturbation problem. *Transportation Planning and Technology*, 27(2):91–117.
- [Archetti *et al.*, 2008] ARCHETTI, C., FEILLET, D., HERTZ, A. et SPERANZA, M. G. (2008). The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*.
- [Assad, 1978] ASSAD, A. A. (1978). Multicommodity network flows—a survey. *Networks*, 8(1):37–91.
- [Balakrishnan *et al.*, 1997] BALAKRISHNAN, A., MAGNANTI, T. L. et MIRCHANDANI, P. (1997). Network design. *Annotated bibliographies in combinatorial optimization*, pages 311–334.
- [Balakrishnan *et al.*, 1989] BALAKRISHNAN, A., MAGNANTI, T. L. et WONG, R. T. (1989). A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740.
- [Balinski, 1961] BALINSKI, M. L. (1961). Fixed cost transportation problems. *Naval Research Logistics Quarterly*, 8:41–54.

-
- [Barnhart *et al.*, 1998] BARNHART, C., BOLAND, N. L., CLARKE, L. W., JOHNSON, E. L., NEMHAUSER, G. L. et SHENOI, R. G. (1998). Flight string models for aircraft fleetling and routing. *Transportation Science*, 32(3):208.
- [Barnhart *et al.*, 2000] BARNHART, C., JIN, H. et VANCE, P. H. (2000). RailRoad Blocking: A Network Design Application. *Operations Research*, 48(4):603–614.
- [Barnhart *et al.*, 2002] BARNHART, C., KRISHNAN, N., KIM, D. et WARE, K. (2002). Network design for express shipment delivery. *Computational Optimization and Applications*, 21(3):239–262.
- [Barnhart et Schneur, 1996] BARNHART, C. et SCHNEUR, R. (1996). Network design for express freight service. *Operations Research*, 12(6):852–863.
- [Berger *et al.*, 2000] BERGER, D., GENDRON, B., POTVIN, J. Y., RAGHAVAN, S. et SORIANO, P. (2000). Tabu search for a network loading problem with multiple facilities. *Journal of Heuristics*, 6(2):253–267.
- [Bratu et Barnhart, 2006] BRATU, S. et BARNHART, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9(3):279–298.
- [Brauner *et al.*, 2007] BRAUNER, N., CRAMA, Y., GRIGORIEV, A. et VAN DE KLUNDERT, J. (2007). Multiplicity and complexity issues in contemporary production scheduling. *Statistica Neerlandica*, 61(1):75–91.
- [Charnes *et al.*, 1987] CHARNES, A., DUFFUAA, S. et RYAN, M. (1987). The more-for-less paradox in linear programming. *European Journal of Operational Research*, 31:194–197.
- [Charnes et Klingman, 1971] CHARNES, A. et KLINGMAN, D. (1971). The 'more-for-less' paradox in the distribution models. *Cahier du Centre d'Etude de Recherche Opérationnelle*, 13:11–22.
- [Chou *et al.*, 2003] CHOU, M., SONG, M. et TEO, C. (2003). Inventory-routing problem in sea freight: Direct versus transshipment model. Rapport technique, The Logistics Institute - Asia Pacific.
- [Clarke *et al.*, 1997] CLARKE, L., JOHNSON, E., NEMHAUSER, G. et ZHU, Z. (1997). The aircraft rotation problem. *Annals of Operations Research*, 69:33–46.
- [Clausen *et al.*, 2009] CLAUSEN, J., LARSEN, A., LARSEN, J. et REZANOVA, N. J. (2009). Disruption management in the airline industry - Concepts, models and methods. *Computers and Operations Research*.
- [Cordeau *et al.*, 2001] CORDEAU, J. F., STOJKOVIC, G., SOUMIS, F. et DESROSIERS, J. (2001). Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation science*, 35(4):375.
- [Crainic, 2000] CRAINIC, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288.
- [Crainic *et al.*, 2001] CRAINIC, T. G., FRANGIONI, A. et GENDRON, B. (2001). Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1-3):73–99.

- [Crainic et Laporte, 1997] CRAINIC, T. G. et LAPORTE, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438.
- [Crainic et Semet, 2005] CRAINIC, T. G. et SEMET, F. (2005). Recherche opérationnelle et transport de marchandises. *Optimisation combinatoire: applications*, pages 47–115.
- [Crama *et al.*, 1994] CRAMA, Y., KOLEN, A. W. J., OERLEMANS, A. G. et SPIEKSMAN, F. C. R. (1994). Minimizing the number of tool switches on a flexible machine. *International Journal of Flexible Manufacturing Systems*, 6(1):33–54.
- [Cung *et al.*, 2000] CUNG, V.-D., HIFI, M. et LE CUN, B. (2000). Constrained two-dimensional cutting stock problems a best-first branch-and-bound algorithm. *International Transactions in Operational Research*, 7(3):185–210.
- [Dantzig, 1951] DANTZIG, G. B. (1951). Application of the simplex method to a transportation problem. *Activity Analysis of Production and Allocation*, pages 359–373.
- [Dantzig et Wolfe, 1960] DANTZIG, G. B. et WOLFE, P. (1960). Decomposition principle for linear programs. *Operations research*, 8(1):101–111.
- [Deineko *et al.*, 2003] DEINEKO, V. G., KLINZ, B. et WOEGINGER, G. J. (2003). Which matrices are immune against the transportation paradox? *Discrete Applied Mathematics*, 130:495–501.
- [Dor et Tarsi, 1997] DOR, D. et TARSİ, M. (1997). Graph decomposition is np-complete: A complete proof of holyer’s conjecture. *SIAM journal on computing(Print)*, 26(4):1166–1187.
- [Even *et al.*, 1976] EVEN, S., ITAI, A. et SHAMIR, A. (1976). On the complexity of timetable and multicommodity flow problems. *SIAM Journal on Computing*, 5:691.
- [Farvolden et Powell, 1994] FARVOLDEN, J. M. et POWELL, W. B. (1994). Subgradient methods for the service network design problem. *Transportation Science*, 28(3):256.
- [Feillet *et al.*, 2005] FEILLET, D., DEJAX, P. et GENDREAU, M. (2005). Traveling Salesman Problems with Profits. *Transportation Science*, 39(2):188–205.
- [Feremans *et al.*, 2003] FEREMANS, C., LABBE, M. et LAPORTE, G. (2003). Generalized network design problems. *European Journal of Operational Research*, 148(1):1–13.
- [Finke, 1978] FINKE, G. (1978). A unified approach to reshipment, overshipment and postoptimization problems. *Lecture notes in Control and Information Science*, 7:201–208.
- [Finke, 1981] FINKE, G. (1981). Overshipments in linear network problems. *Methods of Operations Research*, 43:205–215.
- [Finke, 1983] FINKE, G. (1983). Minimizing overshipments in bottleneck transportation problems. *INFOR*, 21:121–135.
- [Gendron *et al.*, 1998] GENDRON, B., CRAINIC, T. G. et FRANGIONI, A. (1998). Multi-commodity capacitated network design. *Telecommunications Network Planning*, pages 1–19.
- [Gilmore et Gomory, 1963] GILMORE, P. C. et GOMORY, R. E. (1963). A linear programming approach to the cutting stock problem-Part II. *Operations research*, 11(6):863–888.

-
- [Guisewite et Pardalos, 1990] GUISEWITE, G. M. et PARDALOS, P. M. (1990). Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 25(1):75–99.
- [Hirsch et Danzig, 1954] HIRSCH, W. M. et DANZIG, G. B. (1954). The fixed charge problem. RAND Corporation Rept. RM-1383.
- [Hirsch et Danzig, 1968] HIRSCH, W. M. et DANZIG, G. B. (1968). The fixed charge problem. *Naval Research Logistics Quarterly*, 15:413–424.
- [Hitchcock, 1941] HITCHCOCK, F. L. (1941). The distribution of a product from several sources to numerous localities. *Journal of Mathematical Physics*, 20:224–230.
- [Hoffman et Padberg, 1993] HOFFMAN, K. L. et PADBERG, M. (1993). Solving airline crew scheduling problems by branch-and-cut. *Management Science*, pages 657–682.
- [Holyer, 1981] HOLYER, I. (1981). The np-completeness of some edge-partition problems. *SIAM J. Comput.*, 10(4):713–717.
- [Irnich, 2002] IRNICH, S. (2002). Netzwerk-Design für zweistufige Transportsysteme und ein Branch-and-Price Verfahren für das gemischte Direkt- und Hubflugproblem.
- [Jaillet et al., 1996] JAILLET, P., SONG, G. et YU, G. (1996). Airline network design and hub location problems. *Location Science*, 4(3):195–212.
- [Jarrah et al., 1993] JARRAH, A. I. Z., YU, G., KRISHNAMURTHY, N. et RAKSHIT, A. (1993). A decision support framework for airline flight cancellations and delays. *Transportation Science*, 27(3):266.
- [Karp, 1972] KARP, R. M. (1972). Reducibility among combinatorial problems. *Complexity of computer computations*, 43:85–103.
- [Kim, 1997] KIM, D. (1997). *Large Scale Transportation Service Network Design: Models, Algorithms and Applications*. Thèse de doctorat, Massachusetts Institute of Technology.
- [Klincewicz et Rosenwein, 1995] KLINCEWICZ, J. G. et ROSENWEIN, M. B. (1995). The airline exception scheduling problem. *Transportation Science*, 29(1):4.
- [Kohl et al., 2007] KOHL, N., LARSEN, A., LARSEN, J., ROSS, A. et TIOURINE, S. (2007). Airline disruption management-Perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162.
- [Lai et Lo, 2004] LAI, M. F. et LO, H. K. (2004). Ferry service network design: optimal fleet size, routing, and scheduling. *Transportation Research Part A*, 38:305–328.
- [Laporte, 1992] LAPORTE, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358.
- [Letovsky, 1997] LETOVSKY, L. (1997). *Airline operations recovery: An optimization approach*. Thèse de doctorat, Georgia Institute of Technology, Atlanta, USA.
- [Love et al., 2002] LOVE, M., SORENSEN, K., LARSEN, J. et CLAUSEN, J. (2002). Disruption management for an airline-rescheduling of aircraft. *Lecture Notes in Computer Science*, pages 315–324.
- [Lubbecke et Desrosiers, 2005] LUBBECKE, M. E. et DESROSIERS, J. (2005). Selected topics in column generation. *Operations Research*, 53(6):1007–1023.

- [Magnanti et Wong, 1984] MAGNANTI et WONG (1984). Network design and transportation planning: models and algorithms. *Transportation science*, 18(1):1–55.
- [Magnanti *et al.*, 1995] MAGNANTI, T. L., MIRCHANDANI, P. et VACHANI, R. (1995). Modeling and solving the two-facility capacitated network loading problem. *Operations Research*, 43(1):142–157.
- [Minoux, 1989] MINOUX, M. (1989). Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360.
- [Palpant *et al.*, 2008] PALPANT, M., BOUDIA, M., ROBELIN, C. A., GABTENI, S. et LABURTHER, F. (2008). Roadef 2009 challenge: Disruption management for commercial aviation.
- [Pedersen *et al.*, 2007] PEDERSEN, M. B., CRAINIC, T. G. et MADSEN, O. B. G. (2007). Models and tabu search meta-heuristics for service network design with asset management requirements. 49, Centre Interuniversitaire de Recherche sur les Réseaux d'Entreprise, la Logistique et le Transport.
- [Privault et Finke, 1993] PRIVAULT, C. et FINKE, G. (1993). Tool management on machines. *Proceeding International Conference on Industrial Engineering and Production Management*, pages 667–676.
- [Privault et Finke, 2000] PRIVAULT, C. et FINKE, G. (2000). k-server problems with bulk requests: an application to tool switching in manufacturing. *Annals of Operations Research*, 96(1):255–269.
- [Robb, 1990] ROBB, D. J. (1990). The 'More for Less' Paradox in Distribution Models: An Intuitive Explanation. *IIE Transactions*, 22(4):377–378.
- [Rosenberger *et al.*, 2003] ROSENBERGER, J. M., JOHNSON, E. L. et NEMHAUSER, G. L. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408.
- [Roy, 2006] ROY, B. (2006). Regard historique sur la place de la recherche opérationnelle et de l'aide à la décision en France. *Mathématiques et sciences humaines*, 175.
- [Schrenk *et al.*, 2010] SCHRENK, S., CRAINIC, T. G., CUNG, V.-D. et FINKE, G. (2010). *Comparaison de différentes formulations de conception de réseaux pour un problème de transport de fret avec gestion de véhicules*. Studia Informatica Universalis Hermann.
- [Schrijver, 1998] SCHRIJVER, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons Inc.
- [Sharma, 1977] SHARMA, J. K. (1977). Extensions and special cases of transportation problem : a survey. Mathematics Department, D. N. Postgraduate College, Meerut.
- [Stojković et Soumis, 2001] STOJKOVIĆ, M. et SOUMIS, F. (2001). An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, pages 1290–1305.
- [Szwarc, 1971] SZWARC, W. (1971). The transportation paradox. *Naval Research Logistics Quarterly*, 18(2):185–202.
- [Tang et Denardo, 1988] TANG, C. S. et DENARDO, E. V. (1988). Models arising from a flexible manufacturing machine, Part I: minimization of the number of tool switches. *Operations Research*, 36(5):767–777.

-
- [Teodorović et Stojković, 1995] TEODOROVIĆ, D. et STOJKOVIĆ, G. (1995). Model to reduce airline schedule disturbances. *Journal of Transportation Engineering*, 121:324.
- [Teypez, 2008] TEYPAZ, N. (2008). *Contribution à la résolution de problèmes de transport urbain et de fret : modélisations et algorithmes*. Thèse de doctorat, Institut National Polytechnique de Grenoble.
- [Teypez et Rapine, 2008] TEYPAZ, N. et RAPINE, C. (2008). Graph decomposition into paths under length constraints. *Cahier Leibniz*, n°165.
- [Teypez et al., 2010] TEYPAZ, N., SCHRENK, S. et CUNG, V.-D. (2010). A decomposition scheme for large-scale Service Network Design with asset management. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):156–170.
- [Toth et Vigo, 2002] TOTH, P. et VIGO, D. (2002). The Vehicle Routing Problem. SIAM monographs on discrete mathematics and applications. *Society for Industrial and Applied Mathematics*.
- [Wang et Lo, 2008] WANG, D. Z. W. et LO, H. K. (2008). Multi-fleet ferry service network design with passenger preferences for differential services. *Transportation Research Part B*, 42(9):798–822.
- [Wieberneit, 2008] WIEBERNEIT, N. (2008). Service network design for freight transportation: a review. *OR Spectrum*, 30:77–112.
- [Yan et al., 2005] YAN, S., CHEN, S.-C. et CHEN, C.-H. (2005). Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E*, 42(5):409–430.

Titre : Contributions à la conception de réseau de service en transport.

Résumé : Dans cette thèse, nous nous sommes intéressés à deux problèmes industriels dans le domaine du transport. Le premier est un problème de conception de réseau de service avec gestion de ressources pour un transport régulier de fret. Le second est le problème de gestion de perturbation dans le domaine aérien, sujet du challenge ROADEF'2009. Dans les deux cas, il s'agit de problèmes pratiques difficiles qui comportent des contraintes complexes non standard. Le défi est d'autant plus marqué que les instances à résoudre sont de grandes tailles et que les problèmes comportent une dimension temporelle forte. Nous avons analysé la complexité des problèmes en étudiant la complexité de problèmes combinatoires purs, sous-problèmes au cœur de nos problèmes industriels. Nous présentons différentes formulations MIP du problème de conception d'un réseau de service avec gestion de flotte. Il ressort de notre étude que les formulations à base de cycles pour les véhicules sont très prometteuses. Finalement, nous présentons notre contribution au challenge ROADEF'2009. Nous proposons une méthode de résolution rapide, basée sur une décomposition, permettant de trouver de bonnes solutions à un problème industriel complexe en temps limité.

Mots clés : Conception de réseau de service avec gestion de ressources, Gestion de perturbation dans le domaine aérien, Analyse de complexité, Modélisation MIP, Transport et Logistique, Recherche Opérationnelle.

Title: Contributions to the design of transportation service network.

Abstract: In this thesis, we focused on two industrial problems in the field of transportation. The first is a service network design problem with asset management for regular freight transport. The second is the disruption management problem for commercial aviation from challenge ROADEF'2009. In both cases, we deal with difficult practical problems involving complex non standard constraints. The difficulty is even more pronounced that the instances to be solved are large scale and that the problems have an inherent time dimension.

We have analyzed the complexity of the problems by studying the complexity of pure combinatorial problems, sub-problems at the heart of our industrial problems. We have presented different MIP formulations of the service network design with asset management problem. It appears from our study that the formulations based on cycle variables for the vehicles are very promising. Finally, we have presented our contribution to challenge ROADEF'2009 proposing a fast solving method based on decomposition to find good solutions to a complex industrial problem within limited computational time.

Keywords: Service network design with asset management, Disruption management for commercial aviation, Complexity analysis, MIP, Transportation and Logistics, Operations Research.