

UNIVERSITÉ DE MONTRÉAL

CONSTRUCTION SIMULTANÉE D'HORAIRES DE CHAUFFEURS ET
D'ITINÉRAIRES D'AUTOBUS

LOÏC BODART

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)

MAI 2004



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97929-6

Our file Notre référence

ISBN: 0-612-97929-6

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

CONSTRUCTION SIMULTANÉE D'HORAIRES DE CHAUFFEURS ET
D'ITINÉRAIRES D'AUTOBUS

présenté par : BODART Loïc

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. SOUMIS François, Ph.D., président

M. DESAULNIERS Guy, Ph.D., membre et directeur de recherche

M. DESROSIERS Jacques, Ph.D., membre et codirecteur de recherche

M. FLEURENT Charles, Ph.D., membre

Remerciements

Je remercie avant tout mon directeur, M. Guy Desaulniers, qui a su me faire confiance initialement pour un stage de fin d'études et qui au cours de celui-ci a su me faire découvrir le domaine de la recherche opérationnelle. Ses conseils éclairés, sa disponibilité, sa patience et sa jovialité ont permis de transformer ce projet de maîtrise en une très belle expérience. Je tiens aussi à remercier son support financier tout au long de ce projet. Un gros merci Guy.

Je remercie mon codirecteur, M. Jacques Desrosiers, pour ses conseils, ses idées et sa patience. Il fut d'un grand recours pour moi surtout concernant la stabilisation des variables duales. À cette occasion, je tiens aussi à remercier Hatem Ben Amor qui fut très patient et disponible pour répondre à mes multiples questions concernant la stabilisation duale.

Au cours de l'implantation et le codage des différents modèles, j'ai eu l'aide très précieuse de Eric Vatri qui a répondu patiemment à mes nombreuses interrogations et qui a su m'enrichir de ses connaissances informatiques par la clarté de ses explications. Je voudrais profiter de l'occasion pour remercier le reste des membres du GERAD qui, de près ou de loin, ont aidé à la réussite de ce projet.

Je remercie ma famille et en particulier mes parents pour leur confiance, leur soutien constant outre-atlantique malgré des conditions délicates. Merci beaucoup.

Merci à Ann-Sophie pour m'avoir toujours soutenu et m'avoir apporté son énergie. Merci à mes amis.

RÉSUMÉ

Ce mémoire est consacré au problème de construction simultanée d'horaires de chauffeurs et d'itinéraires d'autobus. Ce problème consiste à construire des horaires de chauffeurs et des itinéraires d'autobus en minimisant le nombre de chauffeurs utilisés pour couvrir l'ensemble des voyages requis sur une ligne et sur un horizon d'une journée. Les journées de travail (horaires) des chauffeurs auront différentes caractéristiques : certaines comporteront une seule pièce de travail et d'autres comporteront deux pièces de travail entrecoupées d'un bris.

Ce type de problème est formulé à l'aide d'un modèle de partitionnement d'ensemble avec contraintes supplémentaires. Les tournées d'autobus sont construites de manière indirecte par rapport au modèle. La méthode de base proposée qui repose sur des idées de Haase *et al.* (2001) consiste à résoudre la relaxation linéaire du problème avec une technique de génération de colonnes. Les tests pour cette méthode de base ont été effectués sur des données réelles fournies par la compagnie Giro Inc. Ceux-ci démontrent que le modèle de base ne permet pas de résoudre les plus grandes instances dans des temps raisonnables. Le but de ce mémoire est donc de proposer des stratégies pour accélérer les temps de résolution.

Une première technique d'accélération concerne la résolution des sous-problèmes se basant sur des idées de Freling *et al.* (1999, 2003). Elle simplifie les réseaux en éliminant toute représentation des bris et en réduisant le nombre de ressources nécessaires de 7 à 1. Ces réseaux permettent de ne générer que des pièces de travail et non plus des journées de travail. Suite à la génération de ces pièces par un algorithme de plus court chemin avec contraintes de ressource, un second algorithme combine les

pièces générées pour fabriquer des journées de travail légales. Les résultats obtenus démontrent l'apport considérable de cette stratégie au sein de la résolution : un gain important (en moyenne 90 %) au niveau du temps de résolution.

Une seconde technique d'accélération stabilisant de façon proximale les variables duales a été implantée afin d'accélérer les temps de résolution du problème maître. Elle repose sur des idées de Ben Amor *et al.* (2003) et consiste à borner et même fixer initialement certaines variables duales puis à stabiliser celles-ci au cours des itérations de la génération de colonnes. Les résultats obtenus avec cette stratégie, démontrent sa contribution surtout lors d'une résolution heuristique de la relaxation linéaire du problème (gain moyen de 30 % sur le temps de résolution total).

Enfin, pour obtenir des solutions entières, nous utilisons une méthode heuristique de séparation et évaluation progressive où chaque relaxation linéaire rencontrée est résolue par une approche avec arrêt prématuré de la génération de colonnes, et dans laquelle sont intégrées les deux méthodes d'accélération. Les solutions entières obtenues sont avérées optimales et les temps de résolution pour y aboutir sont fort acceptables même pour les plus grandes instances fournies par Giro Inc. En effet, la plus grande instance qui comporte 463 voyages et 926 segments de voyages a été résolue optimalement en moins de 8 heures de temps de calcul.

ABSTRACT

This master's thesis considers the construction of simultaneous vehicle and crew schedules in urban transportation. This problem consists of constructing vehicle and crew schedules with the objective of minimizing the total number of drivers subject to the constraints of covering all trips over one bus line for one day. The driver working days (schedules) have different characteristics : some of them have one piece of work and others have two pieces of work with a break in between.

This problem is formulated as a set partitioning model with side constraints. The bus schedules are not provided directly by this formulation and need to be computed a posteriori using a polynomial time procedure. The proposed basic solution methodology consist of heuristic branch-and-price algorithm adapted from the one proposed by Haase *et al.* (2001). This methodology was tested on real instances supplied by the company Giro Inc. The computed results show that this basic approach cannot solve large instances in reasonable times. The goal of this master's thesis is to propose strategies to speed up the solution process.

The first acceleration technique concerns the subproblem solution and relies on the work of Freling *et al.* (1999, 2003). Instead of directly generating driver schedules, pieces of works are generated in a first step using a resource constrained shortest path procedure. This step uses simplified networks that involve 1 resource instead of 7 and discard all break modeling. In a second step, the generated pieces of work are combined by a simple procedure to yield working days. The results obtained with this technique are very good : on average, the linear relaxation solution times are reduced by 90 %.

The second acceleration technique, based on the ideas of Ben Amor *et al.* (2003), aims at reducing the master problem solution times. It is a proximal technique that stabilizes some of the dual variables at each iteration of the column generation process. With this technique, we succeeded in speeding up the solution times especially when the linear relaxation of the problem is solved heuristically (average reductions of 30 %).

Finally, in order to derive integer solutions, we use a heuristic branch-and-bound algorithm where each linear relaxation is solved by a column generation with an early stopping rule and with the two acceleration techniques described before. With this approach, we obtained for all instances, integer solutions that are optimal in acceptable solution times. Indeed, the largest instance involving 463 trips and 926 trip segments was solved in less than 8 hours of CPU time.

Table des matières

REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xii
LISTE DES FIGURES	xiii
LISTE DES ALGORITHMES	xiv
INTRODUCTION	1
CHAPITRE 1 : LE PROBLÈME SIMULTANÉ DE TOURNÉES D'AUTOBUS ET DE JOURNÉES DE TRAVAIL	7
1.1 Description du problème	7
1.1.1 Contexte et terminologie	7

1.1.2	Les objectifs	10
1.1.3	Les contraintes	11
1.2	Revue de la littérature	12
1.2.1	Construction de tournées de véhicules	13
1.2.2	Construction de journées de travail	16
1.2.3	Construction simultanée de tournées de véhicules et de journées de travail	18
 CHAPITRE 2 : MODÈLE ET MÉTHODE DE RÉOLUTION DE BASE		22
2.1	Formulation mathématique	22
2.2	Réseaux	26
2.2.1	Les noeuds et les arcs	26
2.2.2	Ressources	34
2.3	Méthode de résolution	35
2.3.1	Calcul d'une borne inférieure par génération de colonnes . .	35
2.3.2	Recherche d'une solution entière	37
 CHAPITRE 3 : ACCÉLÉRATION DE LA RÉOLUTION DES SOUS-PROBLÈMES		40

3.1	Réseau générique	40
3.2	Création des journées de travail réalisables	42
3.3	Résultats	47
3.3.1	Description des jeux de données	48
3.3.2	Résultats pour le modèle de base	51
3.3.3	Résultats avec accélération de la résolution des sous-problèmes	53
CHAPITRE 4 : STABILISATION DES VARIABLES DUALES		56
4.1	Choix du type de la stabilisation	56
4.2	Description de la méthode de stabilisation mise en place	59
4.3	Résultats de la stabilisation	66
4.3.1	Relaxation linéaire	66
4.3.2	Résolution en nombres entiers	69
CONCLUSION		74
BIBLIOGRAPHIE		76

Liste des tableaux

Tableau 3.1	Caractéristiques des différents types de journées de travail .	49
Tableau 3.2	Tailles des problèmes avec le modèle de base	50
Tableau 3.3	Résultats de l'approche de base	52
Tableau 3.4	Résultats avec accélération de la résolution des sous-problèmes	54
Tableau 3.5	Résultats heuristique de l'approche avec accélération de la résolution des sous-problèmes pour les gros problèmes . . .	55
Tableau 4.1	Résultats avec stabilisation dynamique	68
Tableau 4.2	Résultats heuristiques de l'approche avec stabilisation . . .	69
Tableau 4.3	Résultats heuristiques en nombres entiers	71
Tableau 4.4	Résultats heuristiques en nombres entiers avec différentes va- leurs des violations	72

Liste des figures

Figure 1	Les différents niveaux de problèmes résolus en transport public	2
Figure 2.1	Les différentes possibilités pour un voyage	29
Figure 2.2	Les différentes possibilités pour un bris	31
Figure 2.3	Les sous-réseaux propres aux heures de départ	33
Figure 2.4	Méthode de résolution	36
Figure 4.1	Variation des valeurs de six variables duales	57
Figure 4.2	Descente de l'objectif de deux méthodes proximales	59
Figure 4.3	Région de confiance dans l'espace dual	62
Figure 4.4	Répartition des voyages sur une journée	63
Figure 4.5	Variation des valeurs d'une même variable duale pour une résolution avec stabilisation (trait plein) et sans stabilisation (trait pointillé).	67

Liste des algorithmes

Algorithme 3.1	Intégration de l'algorithme au sein de la génération de colonnes	42
Algorithme 3.2	ConstruireChemin(chemin) : fonction générant des pièces de travail	45
Algorithme 3.3	Fonction GenererPath2PC(SousListUnePiece,ListUnePiece)	47
Algorithme 4.1	Algorithme de la stabilisation dynamique intégré pour chaque itération i	65

Introduction

De nos jours, dans les grands centres urbains, nombreuses sont les sociétés de transport en commun qui doivent faire face à une double réalité : la baisse du financement public et une diminution de la part de marché, en faveur de l'automobile. Pour contrer cette dernière tendance, ces sociétés doivent maintenir un service de qualité (ponctualité, flexibilité, fréquence élevée du service, ...), tout en diminuant leur budget annuel. Ainsi par exemple, la Société de Transport de Montréal est confrontée à ce problème comme le souligne son directeur général, dans un discours à la Chambre de Commerce le 4 mai 2000. C'est là qu'intervient la recherche opérationnelle.

Comme mentionné dans l'article d'Odoni *et al.* (1994), les problèmes de transport public à résoudre peuvent se décomposer en plusieurs niveaux : stratégique, tactique, planification des opérations et opérations en temps réel. La figure 1 nous décrit de quelle manière se situent au sein de ce type de société les différents problèmes à solutionner afin d'organiser les actions futures.

Le premier niveau de décisions est la stratégie. Cela concerne la planification à long terme de la conception du réseau d'autobus. L'objectif ici sera d'apporter des modifications aux lignes d'autobus, comme l'ajout ou le retrait d'une ligne. Dans ce but, sont mis en place les outils nécessaires à son évaluation pour estimer la direction des changements ultérieurs. Ainsi par exemple, Nguyen *et al.* (2001) proposent un modèle qui a pour but de déterminer l'affectation des passagers dans un réseau de transport, tout en respectant les capacités maximales et les besoins de la clientèle. Les transformations apportées au niveau stratégique auront un grand impact sur le fonctionnement du réseau. De ce fait, les solutions doivent estimer au mieux les exigences futures.

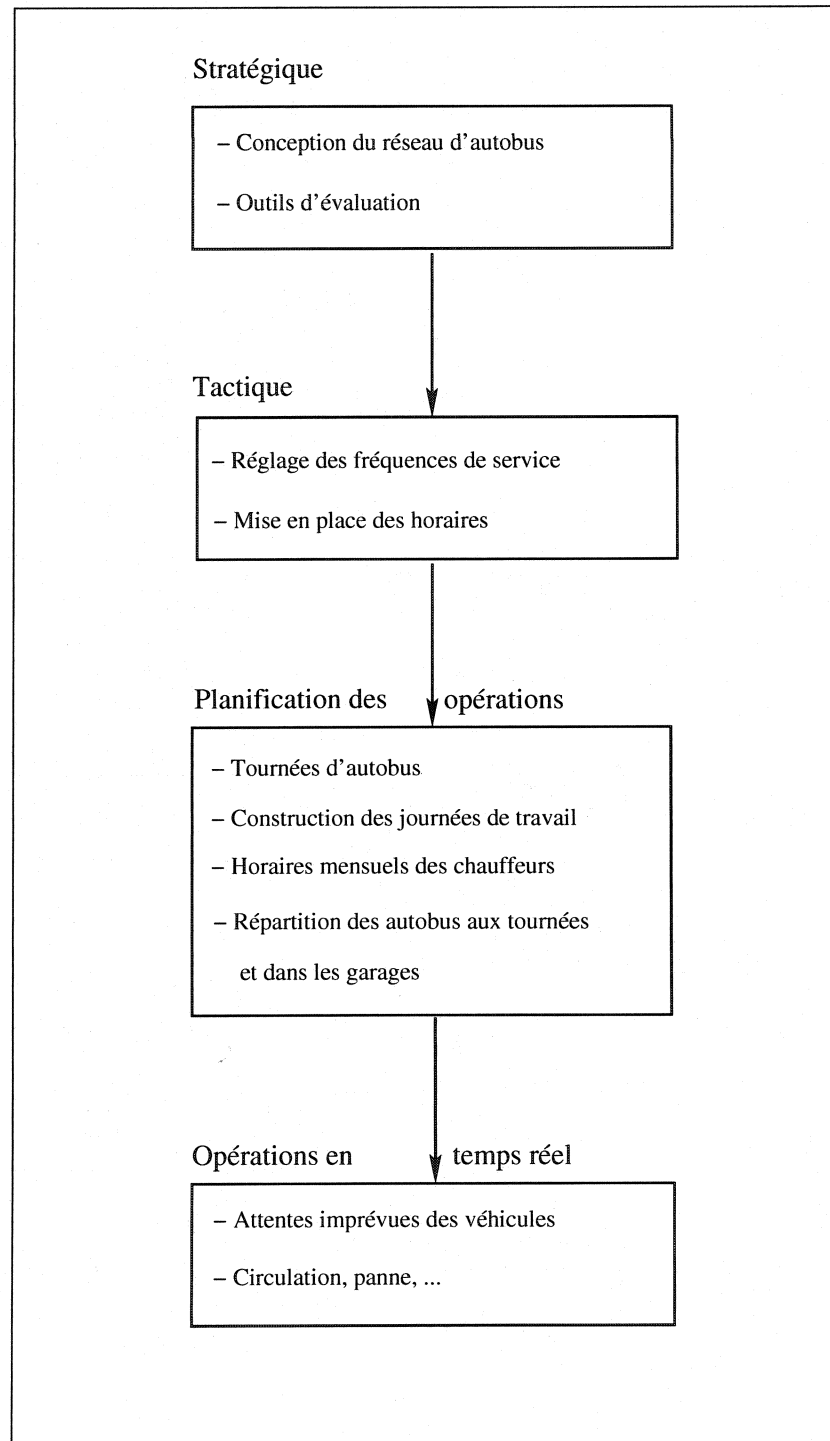


Figure 1 – Les différents niveaux de problèmes résolus en transport public

En deuxième niveau, vient la planification tactique. Ici, il s'agit de s'interroger sur la fréquence de service et sur les heures de passage des autobus à chaque arrêt. Pour le premier paramètre, Ceder (1984) propose une façon de trouver la fréquence appropriée le long d'une ligne pour satisfaire aux différents taux d'arrivée des passagers aux arrêts et respecter les capacités maximales des autobus. Concernant la mise en place des horaires des véhicules le long des lignes, la résolution devra tenir compte des fréquences établies et optimiser la synchronisation des correspondances (voir Ceder, 1986). De manière générale, le degré de répercussion sur le réseau sera moindre comparé au niveau précédent. De ce fait, ce type de planification est modifié un peu plus souvent.

Dans un troisième niveau arrive la planification des opérations. Cette tâche consiste à mettre en place les tournées d'autobus, à construire les différentes journées de travail et les horaires mensuels des chauffeurs, et enfin à établir la répartition adéquate des autobus aux divers voyages et à l'intérieur des garages. Le premier problème consiste à minimiser le nombre d'autobus pour effectuer les différents trajets requis et à construire les tournées à coût minimum. Le deuxième porte sur la minimisation du nombre de chauffeurs pour réaliser les tournées d'autobus exigées et construire les journées de travail en tenant compte de la convention collective. Mais ces deux premiers problèmes étant l'objet de ce mémoire, ils seront détaillés plus loin.

Au sujet des horaires mensuels, définissons un peu mieux ce terme : c'est une suite de journées de travail et de journées de congés s'étalant sur un mois. Ainsi, le problème ici sera de combiner au mieux ces différents types de journées pour obtenir l'horaire mensuel des chauffeurs. Celui-ci devra respecter les besoins journaliers relatifs aux journées de travail et les règles de la convention collective qui régissent entre autres les repos à allouer au cours du mois. Enfin vient le problème du rangement des autobus dans les garages à la fin de la journée. En effet, l'espace réduit des garages oblige les entreprises à s'interroger sur la façon de minimiser le coût de manoeuvre des

autobus devant être affectés le matin aux diverses tournées d'autobus. Gallo et Di Miele (2001) proposent une manière pour optimiser la disposition des autobus garés dans les garages dans le but de débiter au mieux les tournées matinales.

Pour terminer, les opérations en temps réel vont finaliser cette structure. Cette dernière partie fait face à toutes les complications provoquées par les imprévus temporels : mauvais temps, circulation dense, pannes, demandes élevées et imprévues ... En effet, des difficultés se posent face à ces événements et elles doivent être résolues efficacement pour effectuer des modifications sur les opérations planifiées. Ainsi Hickman (2001) propose un modèle analytique pour résoudre les problèmes de retenu de véhicules afin de parer à certains de ces contretemps : il s'agit de mettre en attente certains autobus aux terminus ou aux arrêts de façon à éviter l'empilement des véhicules le long de la même ligne.

Le double problème auquel nous nous attaquons dans ce mémoire se situe dans le troisième niveau. Cela concerne la construction des tournées d'autobus et des journées de travail des chauffeurs. La difficulté qui s'ajoute est de savoir s'il est judicieux de combiner les deux problèmes en un.

Tout d'abord, il est bon de déceler la liaison entre ces deux problèmes : ce sont les voyages à vide (voyages sans passager pour repositionner l'autobus d'un terminus à un autre ou d'un garage à un terminus). En effet, ces voyages à vide ne sont pas connus à l'avance et ils nécessitent un autobus et un chauffeur. Par conséquent, lorsque les problèmes sont résolus séparément, les voyages à vide sont déterminés lors de la construction des tournées de véhicules, sans connaître l'impact qu'ils auront sur le problème de construction des journées.

De ce fait, s'il y a de nombreuses tournées pour les autobus, la construction des journées de travail se réalisera dans une plus grande flexibilité. Dans le cas contraire,

il devient difficile de résoudre séparément le problème de construction de journées de travail, où on obtient des solutions coûteuses. En effet, Freling (1997) a montré dans sa thèse que des économies substantielles peuvent être réalisées en combinant les deux problèmes.

Le problème qui nous intéresse (données issues d'une ville méditerranéenne) présente peu de flexibilité. En effet, les autobus et les chauffeurs sont affectés à une seule ligne d'autobus par jour. Puis le nombre d'autobus est tout de même assez limité. Enfin, les règles de la convention collective sont très restrictives. Par exemple, la durée d'un type de journée doit être comprise entre 7 heures et 7 heures et 30 minutes. Par conséquent, il semble essentiel de résoudre pour ce cas les deux problèmes simultanément.

Le but du mémoire est de pouvoir résoudre les plus grandes instances de ce double problème en un temps raisonnable. En implantant un modèle de base pour ce type de problème en utilisant la génération de colonnes, il en ressort un souci majeur : le temps de résolution des sous-problèmes est beaucoup trop grand. De ce fait, nous proposons une méthode d'accélération des sous-problèmes qui va se révéler très fructueuse. Ensuite, nous proposons une méthode de stabilisation des variables duales afin de réduire principalement les temps de résolution du problème maître. Finalement nous obtiendrons des résultats satisfaisants en insérant ces différentes méthodes au sein d'une méthode heuristique de séparation et évaluation progressive pour obtenir des solutions en nombres entiers.

L'organisation de ce mémoire est la suivante. Le premier chapitre aborde la description et une revue de littérature du problème simultané de tournées d'autobus et de journées de travail. Le deuxième chapitre présente un modèle de base pour ce type de problème. Le chapitre suivant décrit une méthode d'accélération des sous-problèmes au sein de la génération de colonnes. Le dernier chapitre détaille une méthode de

stabilisation des variables duales avant de présenter les résultats obtenus par une résolution heuristique en nombres entiers. Une brève conclusion achève ce mémoire.

Chapitre 1

LE PROBLÈME SIMULTANÉ DE TOURNÉES D'AUTOBUS ET DE JOURNÉES DE TRAVAIL

Dans ce chapitre, nous décrirons dans un premier temps, le problème auquel nous sommes confrontés puis, dans un second temps, nous effectuons une revue de la littérature sur ce même problème. L'acronyme PSTAJT sera utilisé pour désigner le problème simultané de tournée d'autobus et de journées de travail.

1.1 Description du problème

Nous présentons, en premier lieu, le contexte et la terminologie du PSTAJT et nous énumérons ensuite les contraintes et les objectifs du problème.

1.1.1 Contexte et terminologie

Tout d'abord, le PSTAJT considère plusieurs *lignes* d'autobus durant une journée. Une ligne représente le parcours des autobus entre deux *terminaux* (points extrêmes de chaque ligne) en passant par des *points de relève intermédiaires*. Ces points sont propres au sens de parcours des diverses lignes. Les lignes ont aussi un temps de

parcours spécifique qui varie selon l'heure de la journée. Cette caractéristique se retrouve pour tous les temps de parcours entre les différents *points stratégiques* (points qui ont une utilité quelconque dans le PSTAJT) du problème. En effet, la diversité de la densité de la circulation est représentée par ces écarts.

Il existe deux autres types de lieux stratégiques : le *garage* et le *stationnement*. Le premier type de point est l'endroit où se retrouvent les autobus en début et en fin de journée, de même qu'occasionnellement en milieu de journée. Le deuxième type, est un lieu où les autobus peuvent procéder au cours de la journée à des arrêts prolongés. Le PSTAJT considéré dans ce mémoire comporte un seul garage et un nombre quelconque de stationnements. À chacun des points stratégiques (terminaux, points de relève, garage et stationnements), un changement de chauffeurs est possible. C'est d'ailleurs la seule particularité qui justifie l'existence des points de relève intermédiaires sur la ligne.

La ligne est servie par des autobus selon une certaine fréquence spécifique à l'heure de la journée : durant les heures de pointe la fréquence est la plus haute. De ce fait, la ligne possède un propre horaire journalier dans chacune des deux directions qu'il faut nécessairement respecter. Cela caractérise un ensemble de *voyages* à effectuer sur la ligne durant la journée. Ces voyages désignent plus précisément des parcours de la ligne (dans un sens comme de l'autre) effectués à une certaine heure de la journée. L'autobus doit rester le même tout au long d'un voyage pour un souci de respect et de bien être de la clientèle. Du point de vue des chauffeurs, un voyage peut être scindé en plusieurs parties appelées des *segments de voyage*. Ceux-ci définissent un parcours entre deux points stratégiques (un terminal ou un point de relève intermédiaire) consécutifs le long du voyages à une certaine heure de la journée.

L'autobus doit effectuer une *tourné d'autobus* durant la journée. Cette tournée désigne plus exactement le parcours journalier d'un autobus donné : celui débutera et

finira obligatoirement au garage, et il sera composé d'une succession de voyages, de *voyages à vide* et de *battements*. On désigne par ce dernier mot le temps d'attente à un terminal entre deux voyages. L'expression *voyage à vide* définit un parcours d'autobus sans passager. Ce trajet peut être entre deux terminaux, ou entre le garage et un terminal, ou encore entre un stationnement et un terminal (dans un sens comme dans un autre pour chaque possibilité).

Un chauffeur doit réaliser plusieurs segments de voyage dans sa *journée de travail*. Ce terme désigne les tâches que doit réaliser un chauffeur selon un horaire bien précis durant une journée. Pendant celle-ci, le chauffeur peut partir en pause pour une certaine durée puis poursuivre ensuite sa journée de travail. Celle-ci sera alors scindée en plusieurs parties que nous définissons comme des *pièces de travail*. Nous appelons un *bris* le temps de repos entre deux de ces pièces. A noter que ces définitions induisent qu'un chauffeur effectue sa pièce de travail sans le moindre changement d'autobus.

L'existence des pièces de travail engendre plusieurs combinaisons possibles pour établir une journée de travail. En effet, certaines d'entre elles auront une pièce de travail, d'autres deux. De plus, ces pièces peuvent être réalisées à divers moments de la journée. Cela nous amène à définir plusieurs *types de journée de travail*. Ils sont alors caractérisés par le nombre de pièces que contient une journée de travail et par le moment de la journée où ces pièces sont réalisées. Par exemple, dans les jeux de données que nous avons utilisés, il existe des journées de types « AM » ayant une seule pièce devant être réalisée le matin.

Finalement, les diverses journées de travail seront effectuées par les chauffeurs en une durée propre à chacune que l'on définit par le *temps de travail*. Il comprend tout notamment le *temps de préparation*, sauf les temps de bris et ceux de *déplacements à pied*. Ce premier terme désigne la durée mise par le chauffeur pour préparer

son véhicule avant d'entrer ou sortir du garage ou d'un stationnement. Ensuite, on caractérise par le deuxième terme, le déplacement hors des autobus nécessaire aux chauffeurs avant (ou après) de débiter (ou terminer) leurs pièces de travail. Enfin, une dernière caractéristique qui ressort des journées de travail est l'*amplitude de travail*. C'est en fait la différence entre l'heure de la fin de journée et l'heure de début de la journée.

Étant donné un ensemble de voyages prévus à l'horaire, le PSTAJT consiste à fabriquer des tournées d'autobus et des journées de travail afin d'effectuer ces voyages. Voyons les objectifs visés et les contraintes à respecter pour ce problème.

1.1.2 Les objectifs

Pour le PSTAJT, différents objectifs pourraient être utilisés. Dans les divers problèmes qui nous ont été proposés, le principal objectif est le suivant : couvrir la totalité des voyages à effectuer avec un coût minimum. Comme les coûts sont engendrés par les chauffeurs et les autobus et que les chauffeurs ont un salaire horaire identique, une première partie de l'objectif revient à minimiser la durée totale du temps de travail des chauffeurs. Cependant, ces journées de travail ont à peu de choses près la même durée dans les problèmes étudiés. Par conséquent, l'objectif revient approximativement à minimiser le nombre de journées de travail des chauffeurs.

Deuxièmement, il est nécessaire de minimiser le coût total des autobus. Cependant, le coût des autobus est fixe sur les voyages. Ainsi, cet objectif revient à minimiser le coût de l'ensemble des voyages à vide. Toutefois, comme chaque voyage à vide requiert un chauffeur et que la première partie de l'objectif cherche à minimiser les salaires des chauffeurs, le deuxième objectif est en quelque sorte pris en compte dans le premier objectif. Par conséquent, il est de coutume de viser un seul objectif, soit de minimiser le nombre de journée de travail.

Finalement, dans un but d'équité entre les chauffeurs, il serait bien de construire des journées de travail qui ont autant que possible la même durée. De ce fait, dans un objectif secondaire, l'idéal serait de respecter la moyenne du temps de travail par type de journées, donnée par la convention collective, en réduisant la variance entre chacune des journées de travail.

1.1.3 Les contraintes

Tout d'abord, les contraintes de couverture demeurent les principales. En effet, il est impératif de couvrir l'ensemble des voyages, des voyages à vide et des battements de manière unique. Par conséquent, pour chaque segment de voyage, il faut affecter un autobus et un chauffeur. Pour couvrir les segments de voyage avec les autobus, on construit des tournées qui incluent des voyages à vide et des battements. Or, pour chaque voyage à vide et pour chaque battement, il faut y associer nécessairement un chauffeur et un autobus de manière unique. Ces exigences de couverture sont à placer au premier plan. En effet, elles doivent être en aucun cas non respectées afin de réaliser la fréquence planifiée.

Les autres contraintes à considérer dans le PSTAJT sont principalement issues de la convention collective. Celle-ci est en fait un accord relatif aux conditions de travail et aux garanties sociales signé entre les représentants des salariés et les organisations patronales. Elle complète les dispositions du code du travail. Par conséquent, les règles qui la régissent sont généralement précises et doivent être respectées.

Les chauffeurs, comme le précise la convention collective, doivent se soumettre à certaines contraintes de temps au niveau de leurs journées de travail. Comme nous l'avons précisé auparavant, il existe plusieurs types de ces journées de travail. Ainsi chacune de ces contraintes temporelles va dépendre du type de journées de travail que les chauffeurs réalisent. On y retrouve notamment :

- les temps de travail minimum et maximum par jour ;
- le temps de travail moyen pour l'ensemble des journées ;
- l'amplitude maximale par journée ;
- les durées minimale et maximale de chaque pièce dans chaque journée ;
- les durées minimale et maximale d'un bris ;
- les heures de début et de fin d'un bris ; et
- la durée maximale d'un battement à un terminus.

Nous considérons ici le *temps de travail* comme la somme de tous les types de durées dans une journée de travail d'un chauffeur, sauf les temps de bris et de déplacements à pied. Il est à noter que le temps de travail moyen est un objectif secondaire à atteindre plutôt qu'une contrainte stricte.

En dehors de la convention collective, nous retrouvons des contraintes sur les déplacements admissibles. En effet, certains voyages à vide reliant certains points stratégiques éloignés les uns des autres sont interdits puisque trop coûteux. De plus, comme les autobus doivent se trouver au garage en début et en fin de journée, les stationnements ne servent qu'au cours de la journée et ne peuvent donc pas accueillir les autobus en fin de journée. Il est à noter qu'il n'y a pas de contrainte portant sur la quantité maximale d'autobus pour couvrir les diverses tournées. De ce fait, on suppose que le nombre d'autobus disponibles est suffisant pour assurer tous les déplacements proposés.

1.2 Revue de la littérature

La résolution du PSTAJT est connexe à trois types de problèmes rencontrés dans la littérature : la construction de tournées de véhicules, la construction de journées de travail pour les chauffeurs et la construction simultanée de tournées de véhicules

et de journées de travail. Les différents problèmes décrits dans la littérature peuvent être classés selon certaines caractéristiques, ces dernières influençant la complexité et la solution du problème résolu. Il est à noter que cette revue de la littérature se concentre sur les travaux relativement récents qui reposent sur des approches de programmation mathématique.

1.2.1 Construction de tournées de véhicules

Le problème de construction de tournées de véhicules demeure l'un des problèmes de base en recherche opérationnelle. Depuis plus de quinze ans, la recherche s'est intéressée plus particulièrement à un problème de grande complexité (problème NP-dur) : celui des tournées de véhicules avec plusieurs dépôts (PTVDM). Plusieurs études ont été faites sur ce type de problème, et nous allons en détailler les principales.

Bianco *et al.* (1994) ont formulé le PTVDM comme un problème de partitionnement d'ensemble avec contraintes supplémentaires. Toutefois, au lieu de résoudre directement cette formulation, ils réduisent d'abord la taille de l'ensemble des tournées admissibles en utilisant un critère basé sur le coût réduit de variables associés aux tournées. Ce critère fait appel à une borne supérieure qui se calcule à l'aide d'une « bonne » solution duale de la relaxation linéaire du problème. Pour trouver celle-ci, les auteurs proposent une recherche heuristique d'approximations successives, où chaque approximation est améliorée à l'aide d'une procédure heuristique. Les auteurs proposent aussi un critère semblable afin d'éliminer des arcs du réseau permettant de générer des tournées réalisables. Le modèle ainsi réduit peut être résolu par une méthode classique de séparation et évaluation progressive. De ce fait, ils ont pu résoudre avec ce procédé des problèmes allant jusqu'à 250 tâches et 6 dépôts. Il est à noter enfin que les mêmes auteurs ont développé dans Mingozi *et al.* (1995) cette méthode pour le PTVDM avec fenêtres de temps (PTVDMFT). En effet, ils ont réussi

à utiliser une notion de coût réduit pour chaque arc afin de pouvoir appliquer leur technique spécifique d'élimination d'arcs. Ils ont pu ainsi solutionner des problèmes allant jusqu'à 120 tâches et 5 dépôts.

Durant la même période, Ribeiro et Soumis (1994) ont proposé deux formulations du PTVDM : une comme un problème de partitionnement d'ensemble avec contraintes supplémentaires, et l'autre comme un problème de multi-flot avec contraintes additionnelles. Ils montrent qu'elles sont équivalentes (produisent la même borne) et suggèrent de résoudre la première par une méthode de génération de colonnes imbriquée dans une procédure de séparation et évaluation progressive. Le sous-problème consiste alors à résoudre un problème de plus court chemin. Les décisions de branchement sont prises sur les arcs des sous-problèmes. Avec cette technique, la taille des problèmes résolus atteint 600 tâches associées à 6 dépôts.

Basée sur la modélisation et la méthodologie précédente, Desaulniers *et al.* (1998) proposent une nouvelle approche pour le PTVDMFT. Ils généralisent le modèle de Ribeiro et Soumis au cas avec fenêtres de temps. Par conséquent, la résolution est identique, à l'exception des sous-problèmes où l'on retrouve des contraintes de ressources au niveau du problème de plus court chemin. Puis les auteurs vont plus loin en considérant des coûts d'attentes réelles. De ce fait, les sous-problèmes gardent une structure identique au modèle de Ribeiro et Soumis, mis à part leur fonction objective qui devient non linéaire. La méthodologie employée permet alors de solutionner de manière optimale des problèmes comportant jusqu'à 250 tâches et 5 dépôts.

Hadjar *et al.* (2002) combinent des idées provenant des approches de Ribeiro et Soumis (1994) et Bianco *et al.* (1994) pour développer un nouvel algorithme. En effet, ils emploient une procédure de séparation et évaluation progressive qui permet d'éliminer des variables selon le critère sur les coûts réduits et utilisent la génération de colonnes pour la résolution de la relaxation linéaire. De plus, les auteurs insèrent

dans cet algorithme une génération de nouvelles contraintes qui aboutissent à la production de facettes. Par conséquent, la recherche d'une solution entière est accélérée d'où la résolution de problèmes jusqu'à 800 tâches et 6 dépôts.

Mesquita et Paixão (1999) comparent plusieurs formulations dont deux sous forme de multi-flots. La première comporte deux types de variables : un type pour la connexion des tâches et l'autre pour l'affectation aux dépôts. La deuxième comprend un type de variable qui est la connexion des tâches associée à l'affectation de dépôt (comme pour le modèle de Ribeiro et Soumis). Ils démontrent que leurs relaxations linéaires donnent la même borne que celle obtenue par Ribeiro et Soumis. Ils résolvent par une procédure de séparation et évaluation progressive les deux modèles de multi-flots et obtiennent une résolution plus rapide avec leur formulation. Toutefois, afin d'être équitable, le modèle de Ribeiro et Soumis (1994) devrait être résolu par génération de colonnes tel que proposé dans leur article.

Enfin, Löbel (1998) a développé une technique pour résoudre des PTVDM de grande taille. Il formule le PTVDM comme un problème de multi-flots. Sa résolution se fait par génération de colonnes sur les variables (arcs) de cette formulation, où il insère une nouvelle méthode qu'il nomme "Lagrangean pricing". En plus de générer des variables de coûts réduits négatifs, cette technique permet de générer des variables de coûts réduits non négatifs complémentaires aux variables de coûts réduits négatifs (les arcs sélectionnés forment un chemin). Cela permet de solutionner de grands problèmes (25 000 tâches et 44 dépôts) de manière heuristique avec un gap assez petit.

Il est à noter que la taille des problèmes résolus par Löbel (1998) est difficilement comparable avec celle des problèmes résolus dans les autres articles. En effet, la très grande majorité des tâches ne peuvent être effectuées que par un ou deux dépôts. De plus, il semblerait que les tâches ne soient pas des tâches agrégées comme utilisées

par les autres auteurs. Pour chaque tâche non-agrégée, il existe la plupart du temps une autre tâche qui peut être effectuée à coût nul immédiatement après celle-ci. Il est donc évident que ces deux tâches seront fort possiblement faites de façon consécutive, d'où l'idée de les agréger. Finalement, notons que l'approche de Löbel (1998) ne peut être utilisée dans un contexte avec contraintes de temps ou de ressource puisque, dans ces cas les contraintes du problème maître seraient non linéaires.

1.2.2 Construction de journées de travail

Le problème de construction de journées de travail est relié à plusieurs domaines comme le transport aérien, le transport ferroviaire, le transport urbain, ... Nous allons ici nous intéresser plus particulièrement aux problèmes de construction des journées de travail (PCJT) des chauffeurs d'autobus.

L'approche proposée par Desrochers et Soumis (1989) fait partie des modèles de référence dans la résolution du PCJT. Les auteurs appliquent une méthode de génération de colonnes (une colonne représente une journée de travail réalisable d'un chauffeur) qui décompose le PCJT en deux parties : un problème maître qui est un problème de recouvrement (on permet aux chauffeurs de voyager en tant que passagers), et un sous-problème qui est un problème de plus court chemin avec contraintes de ressource. Cette formulation permet alors de résoudre des problèmes de 235 tâches donnant 50 journées de travail.

Mingozi *et al.* (1999) formulent le PCJT comme un problème de partitionnement d'ensembles avec contraintes supplémentaires. De la même façon que dans Bianco *et al.* (1994), les auteurs calculent une borne inférieure du PCJT à l'aide d'une solution heuristique du dual de la relaxation linéaire du problème. Ensuite la solution duale est utilisée pour réduire le nombre de variables dans le problème afin de résoudre

celui-ci par une procédure de séparation et évaluation progressive classique. Ils ont pu ainsi solutionner des problèmes allant jusqu'à 500 tâches (donnant 208 journées de travail).

DeSilva (2001) propose de résoudre le PCJT en combinant la programmation par contraintes et la programmation linéaire, afin d'utiliser les forces propres à chacune des deux méthodes. L'auteur formule le PCJT comme un problème de partitionnement d'ensemble, et énonce le sous-problème au sein de la génération de colonnes comme un problème de programmation par contraintes. Par conséquent, l'auteur a pu solutionner des problèmes comprenant jusqu'à 495 tâches pour donner 66 journées de travail.

Freling *et al.* (1999, 2003) proposent pour le PCJT avec un seul dépôt une résolution heuristique par génération de colonnes où le problème maître est résolu approximativement par relaxation lagrangienne. Le sous-problème qui est un problème de plus court chemin, est résolu en deux étapes : la première fournit un ensemble de pièces réalisables et la deuxième permet de générer des journées de travail à partir des pièces précédentes. Pour éviter de générer deux fois la même colonne, les multiplicateurs de Lagrange sont modifiés avant de produire de nouvelles colonnes dans le but de rendre le coût réduit des colonnes déjà générées non-négatifs. De cette manière, les auteurs ont réussi à résoudre des instances de 238 segments et 24 journées de travail.

Borndörfer *et al.* (2001) proposent à partir de la méthode de génération de colonnes, différentes techniques pour améliorer la résolution de PCJT de grande taille. Le sous-problème formulé comme un problème de plus court chemin avec contraintes de ressource est ici résolu par relaxation lagrangienne contrairement à Desrochers et Soumis (1989) qui le résolvent par programmation dynamique. Au niveau du problème maître, les auteurs résolvent le dual par une méthode heuristique d'ascension selon les coordonnées (*Coordinate Ascent*) en y insérant en plus une méthode de stabilisation. Pour l'intégrité de la solution heuristique, les auteurs utilisent une procédure

de séparation et évaluation progressive avec une stratégie de profondeur d'abord. Celle-ci se réalise en choisissant la branche présentant la moins grande détérioration de la manière suivante : 20 colonnes candidates sont sélectionnées puis chacun de ces candidats sont évalués sans générer de nouvelles colonnes. Si cette détérioration est petite, aucune colonne est générée et une autre colonne est fixée à nouveau selon le même procédé. Par contre, dans le cas d'une grande détérioration, des colonnes sont générées avant d'en fixer à nouveau. Par conséquent, les auteurs arrivent à solutionner des problèmes jusqu'à 1968 tâches (donnant 111 journées de travail) au moyen de cette méthode heuristique.

1.2.3 Construction simultanée de tournées de véhicules et de journées de travail

Depuis quelques années, la recherche s'est développée dans la résolution du problème de construction simultanée de tournées de véhicules et de journées de travail (PCSTVJT). En effet, plusieurs auteurs se sont rendus compte de l'efficacité de cette intégration pour certains types de problèmes, par rapport à la résolution séquentielle traditionnelle du PCSTVJT, soit la résolution du problème de tournées de véhicules puis du PCJT.

Ainsi Freling *et al.* (1999, 2003) suggèrent une modélisation du PCSTVJT avec un seul dépôt où les contraintes de disponibilité des autobus ne sont pas considérées. Ils proposent une approche par génération de colonnes où le problème maître est résolu par relaxation lagrangienne. Les auteurs relaxent alors toutes les contraintes impliquant des variables associées aux journées de travail pour obtenir des sous-problèmes lagrangiens sous forme de problème d'horaires de véhicules avec un seul dépôt. Les sous-problèmes de la génération de colonnes permettent de générer les variables associées aux journées de travail et se formulent comme des problèmes de

plus court chemin avec contraintes de ressources. Cette approche de génération de colonnes permet de résoudre la relaxation linéaire du problème. Une fois résolue, la dernière solution obtenue pour le problème d'horaires de véhicules avec un seul dépôt (le sous-problème lagrangien) est retenue et fixée. Il reste alors à résoudre un PCJT de la même manière (voir section précédente) que celle décrite dans Freling *et al.* (1999, 2003). Freling *et al.* (2003) ont rapporté des résultats présentant un gain substantiel sur le nombre de chauffeurs et de journées de travail par rapport à l'approche séquentielle (résolution du problème de tournées de véhicules puis du PCJT). Ils ont pu ainsi résoudre en un temps raisonnable des instances comportant 148 segments et 23 journées de travail. Dans Freling *et al.* (2001), les auteurs montrent les différents impacts sur le fait d'interdire aux chauffeurs le changement de véhicule à un bris : ils obtiennent un gain plus important sur le nombre de chauffeurs et de journées de travail, comparé à l'approche séquentielle.

Haase *et al.* (2001) proposent une nouvelle approche du PCSTVJT où la formulation comprend des variables associées aux journées de travail auxquelles s'ajoute une variable de compteur de véhicules qui sert notamment à appliquer un coût fixe pour chaque autobus utilisé. Cette formulation tient compte des contraintes de véhicules et permet de retrouver a posteriori les horaires des véhicules en un temps polynomial. La méthode de résolution utilise une approche par génération de colonnes imbriquée dans une procédure de séparation et évaluation progressive. Les auteurs ont ajouté à cela différentes stratégies d'accélération, comme par exemple : enlever des contraintes redondantes, agréger certains noeuds de bris, ajouter des contraintes de conservation de flot, et rendre dynamique les contraintes de véhicules. Ils ont implanté deux versions de cette approche de résolution : l'une exacte qui explore l'arbre de branchement complètement, et l'autre heuristique où plusieurs décisions consistant à fixer à 1 la valeur des variables associées aux journées de travail sont prises à chaque noeud de l'arbre d'énumération. Il est à noter que l'arbre est exploré suivant

une stratégie profondeur d'abord sans retour en arrière. Avec la version exacte, ils ont résolu à partir de données générées aléatoirement, des instances de 400 segments et 60 journées de travail en moins de trois heures. La version heuristique permet de résoudre des problèmes allant jusqu'à 700 segments et 121 journées de travail dans un temps sensiblement identique. Comme ce mémoire est basé sur cette approche, plus de détails sur celle-ci seront donnés dans le chapitre suivant.

Enfin, Huisman *et al.* (2003) proposent deux modèles qui sont des généralisations des modèles de Freeling *et al.* (1999, 2003) et Haase *et al.* (2001) pour le PCSTVJT avec plusieurs dépôts. La résolution de ces modèles comporte deux principales étapes : la première consiste à calculer une borne inférieure sur la valeur de la solution optimale, et la seconde consiste à trouver la meilleure solution réalisable. Dans la première étape, la borne est calculée approximativement en résolvant par génération de colonnes le modèle relaxé propre à chacune des formulations. Les auteurs rapportent obtenir des bornes de meilleures qualités avec la généralisation du modèle de Freeling *et al.* (1999, 2003). La borne obtenue sert à savoir si l'ensemble de colonnes générées lors de ce calcul est satisfaisant. Cet ensemble est par la suite employé pour obtenir des horaires réalisables pour les véhicules en utilisant une méthode de séparation et évaluation progressive sans générer d'autres variables associées aux journées de travail. De ces horaires réalisables de véhicules, des journées de travail sont calculées pour chaque dépôt en utilisant la même méthode que Freeling *et al.* (1999, 2003). De cette façon, les auteurs résolvent des problèmes (concrets ou générés aléatoirement) allant jusqu'à 200 segments associés à 4 dépôts.

L'objectif ultime de ce mémoire est de pouvoir résoudre en des temps raisonnables des PSTAJT réels de grandes tailles (près de 500 segments) incluant la possibilité de laisser les autobus à un stationnement en plein milieu de la journée. Après avoir passé en revue les principaux modèles portant sur le PSTAJT nous avons choisi de reprendre l'approche de génération de colonnes de Haase *et al.* (2001). Quelques

idées de Freling *et al.* (1999, 2003) seront aussi incorporées dans l'approche finale proposée. Afin de réduire les temps de résolution du problème, nous ferons appel à une technique de stabilisation des variables duales.

Chapitre 2

MODÈLE ET MÉTHODE DE RÉSOLUTION DE BASE

Dans ce chapitre, nous présentons une adaptation du modèle mathématique proposé par Haase *et al.* (2001) pour le PSTAJT. Ce modèle sera adapté au contexte décrit dans le chapitre précédent. Les principales différences par rapport à Haase *et al.* (2001) portent sur la possibilité que les autobus se rendent à un stationnement en milieu de journée et sur la fonction objectif : Haase *et al.* (2001) cherche à minimiser le nombre total de tournées d'autobus et de journées de travail, tandis que notre modèle minimise simplement le nombre de journées de travail. Nous décrivons ensuite les réseaux qui permettent de générer les journées de travail pour finalement présenter la méthode de résolution proposée par Haase *et al.* (2001). Ce modèle et cette méthode seront qualifiés par la suite de modèle et méthode de base.

2.1 Formulation mathématique

La formulation mathématique du PSTAJT fait appel à la notation suivante :

K : l'ensemble des types de journées de travail ;

Ω^k : l'ensemble des journées possibles de type $k \in K$ ($p \in \Omega^k$ est alors une journée possible) ;

S : l'ensemble des segments ($s \in S$ est un segment) ;

V : l'ensemble des voyages ($v \in V$ est un voyage) ;

H : l'ensemble des heures de départ possibles du stationnement ($h \in H$ est une heure possible de H). Une heure possible est l'heure à laquelle un chauffeur doit quitter le stationnement au volant d'un autobus afin de pouvoir se rendre au terminus de départ voulu à temps. Nous désignons par $h^* \in H$ la dernière heure possible de H .

La formulation s'appuie sur les paramètres suivants :

c_{ps} : un paramètre binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ couvre le segment de voyage $s \in S$, 0 sinon ;

i_{pv} : un paramètre binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ comporte un voyage à vide ou un battement se terminant au début du voyage $v \in V$, 0 sinon ;

o_{pv} : un paramètre binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ comporte un voyage à vide ou un battement débutant à la fin du voyage $v \in V$, 0 sinon ;

d_{ph} : un paramètre binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ comporte un voyage à vide à partir du stationnement avant ou à l'heure $h \in H$, 0 sinon ;

a_{ph} : un paramètre binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ comporte un voyage à vide arrivant du stationnement avant ou à l'heure $h \in H$, 0 sinon.

La formulation s'appuie sur les variables suivantes :

θ_p : est une variable binaire qui prend la valeur 1 si la journée $p \in \Omega^k$ est retenue, 0 sinon.

En utilisant cette notation, le PSTAJT peut se formuler comme suit :

$$\text{Minimiser } \sum_{k \in K} \sum_{p \in \Omega^k} \theta_p \quad (2.1)$$

sujet à :

$$\sum_{k \in K} \sum_{p \in \Omega^k} c_{ps} \theta_p = 1, \quad \forall s \in S \quad (2.2)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} i_{pv} \theta_p = 1, \quad \forall v \in V \quad (2.3)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} o_{pv} \theta_p = 1, \quad \forall v \in V \quad (2.4)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} d_{ph} \theta_p - \sum_{k \in K} \sum_{p \in \Omega^k} a_{ph} \theta_p \leq 0, \quad \forall h \in H \setminus \{h^*\} \quad (2.5)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} d_{ph^*} \theta_p - \sum_{k \in K} \sum_{p \in \Omega^k} a_{ph^*} \theta_p = 0, \quad (2.6)$$

$$\theta_p \in \{0, 1\}, \quad \forall p \in \Omega^k, k \in K. \quad (2.7)$$

La fonction objectif (2.1) cherche à minimiser le nombre total de journées de travail. Elle est soumise à plusieurs contraintes. Les contraintes (2.2) spécifient que chaque segment de voyage est parcouru une seule fois par un chauffeur et son autobus. Ensuite, les contraintes (2.3) assurent la présence d'un autobus et d'un chauffeur avant le départ de chaque voyage. Symétriquement, les contraintes (2.4) assurent ces présences après la fin du voyage. Il est à noter que ces contraintes permettent de façon indirecte d'amener les autobus au point de départ et de retirer les autobus des points d'arrivée. Les contraintes (2.5) assurent qu'à tout moment $h \in H$ le nombre de départs d'autobus effectués à partir du stationnement depuis le début de la journée soit toujours inférieur ou égal au nombre d'arrivées au stationnement depuis le début de la journée. La contrainte (2.6) oblige l'égalité entre ces deux

nombres au dernier moment de la journée, laissant alors le stationnement vide pour la nuit. Les contraintes (2.7) imposent que les valeurs des variables θ_p soit binaires.

Le modèle (2.1)-(2.7) permet de fournir directement les journées de travail des chauffeurs. En regardant l'ensemble des voyages à vide qui sont inclus dans les journées de travail, il est possible de retracer les différentes tournées d'autobus : du garage vers le stationnement, du garage vers le garage, du stationnement vers le stationnement, du stationnement vers le garage. De ce fait, les horaires des autobus sont fournis par ce modèle mathématiques de manière indirecte.

Le modèle présenté en est un de partitionnement d'ensemble avec contraintes supplémentaires. En pratique, il comporte un très grand nombre de variables, soit une pour chaque horaire de chauffeur valide. Haase *et al.* (2001) ont donc suggéré une approche de génération de colonnes imbriquée dans un algorithme de séparation et évaluation progressive pour résoudre ce modèle. Les sous-problèmes sont des problèmes de plus court chemin avec contraintes de ressource définis sur les réseaux décrits à la section suivante.

Il est à noter que le modèle réellement implanté par Haase *et al.* n'est pas tout à fait un modèle de partitionnement. En effet, dans le modèle présenté ci-dessus, on retrouve quatre contraintes de partitionnement pour chaque voyage (lorsqu'il y a un seul point de relève par voyage), alors que dans le modèle implanté seule une contrainte de partitionnement (2.3) est présente pour chaque voyage. Les trois autres contraintes sont remplacées par des contraintes de conservation de flot. Cela permet de résoudre plus rapidement les différentes instances du PSTAJT avec un modèle équivalent.

2.2 Réseaux

Dans cette section, nous décrivons un réseau espace-temps générique à partir duquel sera construit l'ensemble des réseaux associés aux sous-problèmes de l'approche de génération de colonnes. Ces réseaux permettront d'énumérer toutes les journées de travail possibles. En effet, les réseaux contiennent chacun un noeud « source » et un noeud « puits » : tout chemin du noeud « source » au noeud « puits » respectant les contraintes de ressource correspond à une journée de travail pour un chauffeur.

2.2.1 Les noeuds et les arcs

Le réseau générique est composé de noeuds et d'arcs : les noeuds représentent généralement des lieux physiques à des heures données et les arcs des actions d'un noeud à un autre. Une durée est associée à chaque arc. Dans un premier temps, tous les types de noeud et d'arc utiles aux réseaux sont énumérés. Leurs utilités sont ensuite expliquées par diverses modélisations.

Les différents types de noeud nécessaires sont les suivants :

- Noeud de début de journée : il y a un seul noeud de ce type pour représenter le début de journée d'un chauffeur. Ce noeud est en fait un noeud « source » du réseau générique.
- Noeud de fin de journée : il y a un seul noeud de ce type pour représenter la fin de journée d'un chauffeur. Ce noeud est en fait un noeud « puits » du réseau générique.
- Noeud terminus de départ : il y a un tel noeud pour chaque voyage afin de représenter le début d'un voyage au terminus de départ.
- Noeud terminus d'arrivée : il y a un tel noeud pour chaque voyage afin de représenter la fin d'un voyage au terminus d'arrivée.

- Noeud point de relève intermédiaire : il y a au minimum un tel noeud pour chaque voyage afin de permettre un changement éventuel de chauffeur pour chaque point de relève.
- Noeud début bris : il y a un tel noeud pour chaque début de bris possible.
- Noeud fin bris : il y a un tel noeud pour chaque fin de bris possible.

Les différents types d'arc sont les suivants :

- Arc de début de journée en conduisant un autobus à partir du garage.
- Arc de début de journée en conduisant un autobus à partir du stationnement.
- Arc de fin de journée en conduisant un autobus jusqu'au garage.
- Arc de fin de journée en conduisant un autobus jusqu'au stationnement.
- Arc qui représente un segment de voyage.
- Arc qui représente un battement.
- Arc qui représente un bris de durée minimale.
- Arc qui représente un prolongement de bris.
- Arc qui représente un déplacement en autobus jusqu'au garage suivi d'un bris de durée minimum.
- Arc qui représente un déplacement en autobus jusqu'au stationnement suivi d'un bris de durée minimum.
- Arc qui représente un déplacement en autobus (début d'une pièce) à partir du garage suite à un bris.
- Arc qui représente un déplacement en autobus (début d'une pièce) à partir du stationnement suite à un bris.
- Arc qui représente un déplacement à pied (début d'une pièce) à partir du garage ou du stationnement suite à un bris.
- Arc qui représente un déplacement à pied en début de journée.
- Arc qui représente un déplacement à pied en fin de journée.

A chaque arc est associée une durée qui correspond au temps requis par l'activité associée à l'arc. Ce temps peut comprendre un temps de déplacement, de préparation, d'attente et de bris.

Expliquons maintenant par plusieurs modélisations, la nécessité de tous ces types d'arc et de noeud et la relation logique entre chacun d'eux. La figure 2.1 décrit la partie du réseau relatif aux différentes possibilités de parcours d'un voyage ainsi qu'aux différentes possibilités d'arrivée ou de départ à un voyage. Ainsi, pour arriver au terminus de départ, cinq choix s'offrent à un chauffeur :

- Il est sur place dans l'autobus et effectue un battement après avoir réalisé un voyage. Il est à noter que dans le problème considéré, les voyages à vide qui permettent d'aller plus vite qu'un voyage avec des passagers d'un terminus à un autre sont ignorés.
- Il vient d'effectuer un bris et il revient au terminus à pied.
- Il commence sa journée de travail et il arrive au terminus à pied.
- Il effectue un voyage à vide en autobus du garage au terminus.
- Il effectue un voyage à vide en autobus du stationnement au terminus.

Au cours du voyage en autobus, il peut se produire un changement de chauffeur à un point de relèvement intermédiaire. Un chauffeur qui vient d'un bris ou qui commence sa journée de travail, arrive alors à pied à ce point et prend le relais du chauffeur présent dans l'autobus. Ce dernier va aussi repartir à pied pour effectuer un bris ou terminer sa journée. Enfin les possibilités qui s'offrent au chauffeur au terminus d'arrivée sont symétriquement les mêmes que celles du terminus de départ. Pour chaque voyage, on modélise chacun de ces choix par un arc de type correspondant, tel qu'illustré à la figure 2.1.

Il est à noter que pour les arcs de battement (lien entre deux voyages consécutifs), il faut que le temps de départ du second voyage soit supérieur ou égal au temps d'arrivée

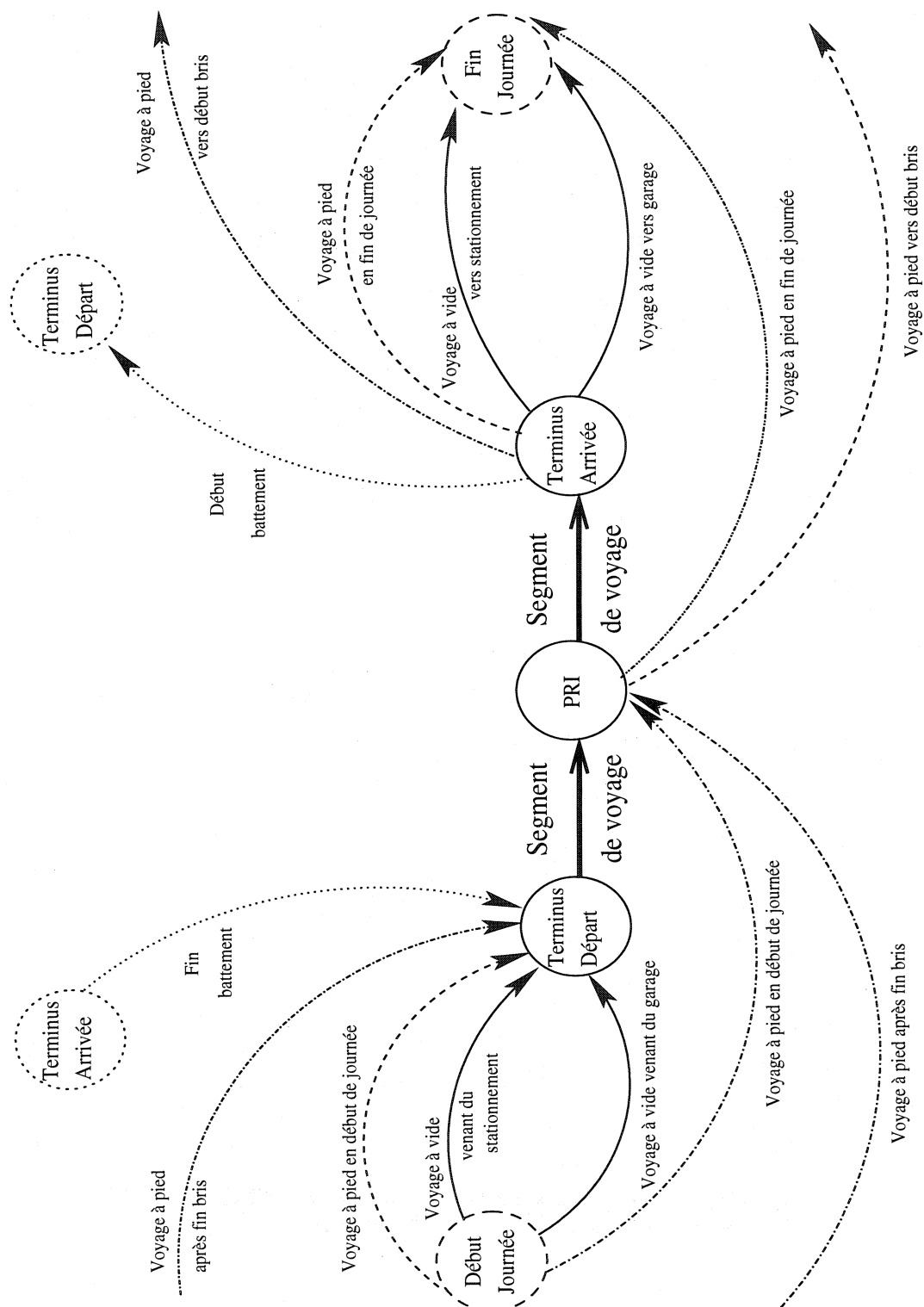


Figure 2.1 – Les différentes possibilités pour un voyage

du premier voyage et que le terminus d'arrivée du premier voyage soit le même que le terminus de départ du second voyage. De plus, le battement entre ces deux voyages ne doit pas excéder le temps de battement maximum (soit 20 minutes pour les tests). Enfin, au début ou à la fin d'un battement, un changement de chauffeur peut aussi s'effectuer de la même façon que le cas détaillé précédemment.

Le schéma 2.1 décrit les différentes alternatives de parcours au point d'origine (noeud « source ») qui symbolise le début de journée d'un chauffeur. Celui-ci peut alors rejoindre à pied un terminus ou un point de relève intermédiaire, où il opérera à un changement de chauffeur. Sinon il procédera à un voyage à vide du stationnement ou du garage pour rejoindre un des terminus. Les possibilités offertes au chauffeur, à la fin de sa journée (noeud « puits »), sont symétriquement les mêmes que celle du début de journée.

Nous terminons la description du réseau générique par la figure 2.2 qui explique la mise en place des bris. Tout d'abord, le chauffeur peut commencer son bris en effectuant un voyage à pied, d'un terminus ou d'un point de relève intermédiaire. Mais il peut aussi le débiter en réalisant un voyage à vide en autobus vers le garage ou le stationnement. Quand la durée minimum du bris du chauffeur est terminée, il va rejoindre un terminus de départ soit par un voyage à vide en autobus du garage ou du stationnement, ou en s'y rendant à pied. Enfin, il peut aussi à la fin de son bris rejoindre à pied un point de relève intermédiaire ou un terminus d'arrivée. À nouveau toutes ces possibilités sont modélisées par un arc de type adéquat au sein du réseau générique.

Le réseau précédemment décrit ne suffit pas à résoudre le problème. En effet, pour les chauffeurs il existe différents types de journées : des types de journée de travail comportant une seule pièce et d'autres en comportant plusieurs pièces. En fait, l'idée ici est d'associer un réseau par type de journée pour répondre à la spécificité de

chacun de ces types de journée. Ainsi par exemple, le réseau des journées AM aura ses propres caractéristiques en ce qui concerne l'heure de début et de fin de journée. De ce fait, les voyages non réalisables dans les heures admissibles de ce type de journée seront retirés. Les autres réseaux seront réalisés de la même façon, suivant leur spécificité au niveau des horaires.

Enfin, dans le but d'accélérer la résolution, nous définissons des sous-réseaux pour chacun de ces réseaux spécifiques à un type de journée de travail (voir figure 2.3). Ceux-ci sont construits avec un noeud source propre à une heure de départ possible et avec le noeud puits commun au type de journée. De cette manière, on obtient un sous-réseau pour chaque heure de départ possible. Chacun des arcs démarrant des noeuds sources appartient au sous-réseau associé au noeud source. De ce fait, on obtient trois groupes d'arcs :

- Les arcs de début de journée propres à chacun des sous-réseaux.
- Les arcs communs à tous les réseaux comme les arcs de segment de voyage ou encore les arcs de battement.
- Les arcs de fin de journée appartenant au réseau spécifique à un type de journée de travail.

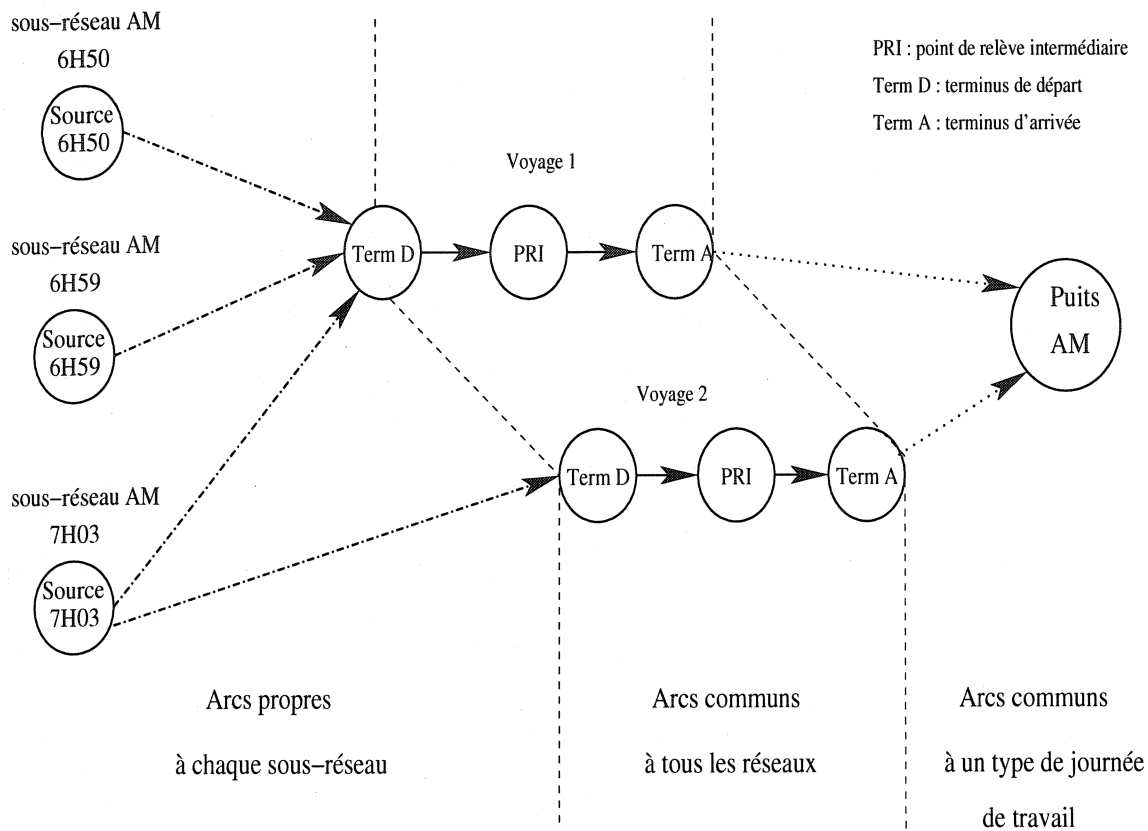


Figure 2.3 – Les sous-réseaux propres aux heures de départ

Dans le but de minimiser le nombre de chauffeurs utilisés pour couvrir l'ensemble des voyages demandés, un coût de un est associé à tous les arcs partant des noeuds « source ». Tous les autres arcs portent un coût nul.

2.2.2 Ressources

Les ressources permettent de gérer certaines contraintes qui s'appliquent à un horaire à la fois. Pour les tests effectués, sept ressources sont utilisées pour gérer les sept contraintes suivantes :

- Nombre minimum de pièces.
- Nombre maximum de pièces.
- Durée minimale d'une pièce.
- Durée maximale d'une pièce.
- Temps minimum de travail.
- Temps maximum de travail.
- Amplitude maximum.

Haase *et al.* (2001) expliquent en profondeur le comportement de ces ressources. À titre d'exemple, la ressource (définie comme une durée cumulée jusqu'à un noeud) pour l'amplitude se comporte de la façon suivante dans le réseau. Tout d'abord, les horaires s'obtiennent en construisant les chemins dans le réseau en partant du noeud du début de journée. Chaque noeud possède les bornes $[0, \text{amplitude maximale}]$ définissant une fenêtre de ressource, et à chaque arc est associé une durée. Aussi pour prolonger un chemin, on part d'un noeud i , pour se rendre à un autre noeud j en utilisant un arc (i, j) . La durée totale cumulée depuis le début de la journée jusqu'au noeud j s'obtient en ajoutant la durée cumulée jusqu'au noeud i plus la durée de l'arc (i, j) . La prolongation du dernier arc est acceptée si et seulement si la durée totale jusqu'au noeud est inférieure à la borne maximale de la fenêtre de ressource. Il est à noter que la borne minimale des fenêtres de ressource pour chacun des noeuds est une borne violable. Cela rend possible une attente avant le passage sur chacun des noeuds.

Il est à noter que la gestion des ressources permet des comportements plus complexes que celui décrit ci-haut. Par exemple, les bornes des fenêtres de ressource peuvent

varier d'un noeud à un autre pour la même ressource, des remises à zéro de la valeur d'une ressource peuvent être effectuées et des fonctions de prolongation plus complexes que l'addition de la durée peuvent être envisagées.

2.3 Méthode de résolution

Le PSTAJT (2.1)-(2.7) formulé comme un problème de partitionnement d'ensemble avec contraintes supplémentaires est résolu par une approche de génération de colonnes imbriquée dans un algorithme de séparation et évaluation progressive de la même manière que dans Haase *et al.* (2001). Cette section présente cette méthode de résolution.

2.3.1 Calcul d'une borne inférieure par génération de colonnes

Lors de la résolution du problème (2.1)-(2.7) l'algorithme de séparation et évaluation progressive va à chacun de ses noeuds de branchement calculer une borne inférieure qui s'obtient en relaxant les contraintes d'intégrité sur les variables θ_p . La relaxation linéaire obtenue s'appelle le problème maître et contient un très grand nombre de variables. Pour la résoudre, on utilise une technique de génération de colonnes (voir Dantzig et Wolfe, 1960). La figure 2.4 suivante présente brièvement la logique interne de cette technique.

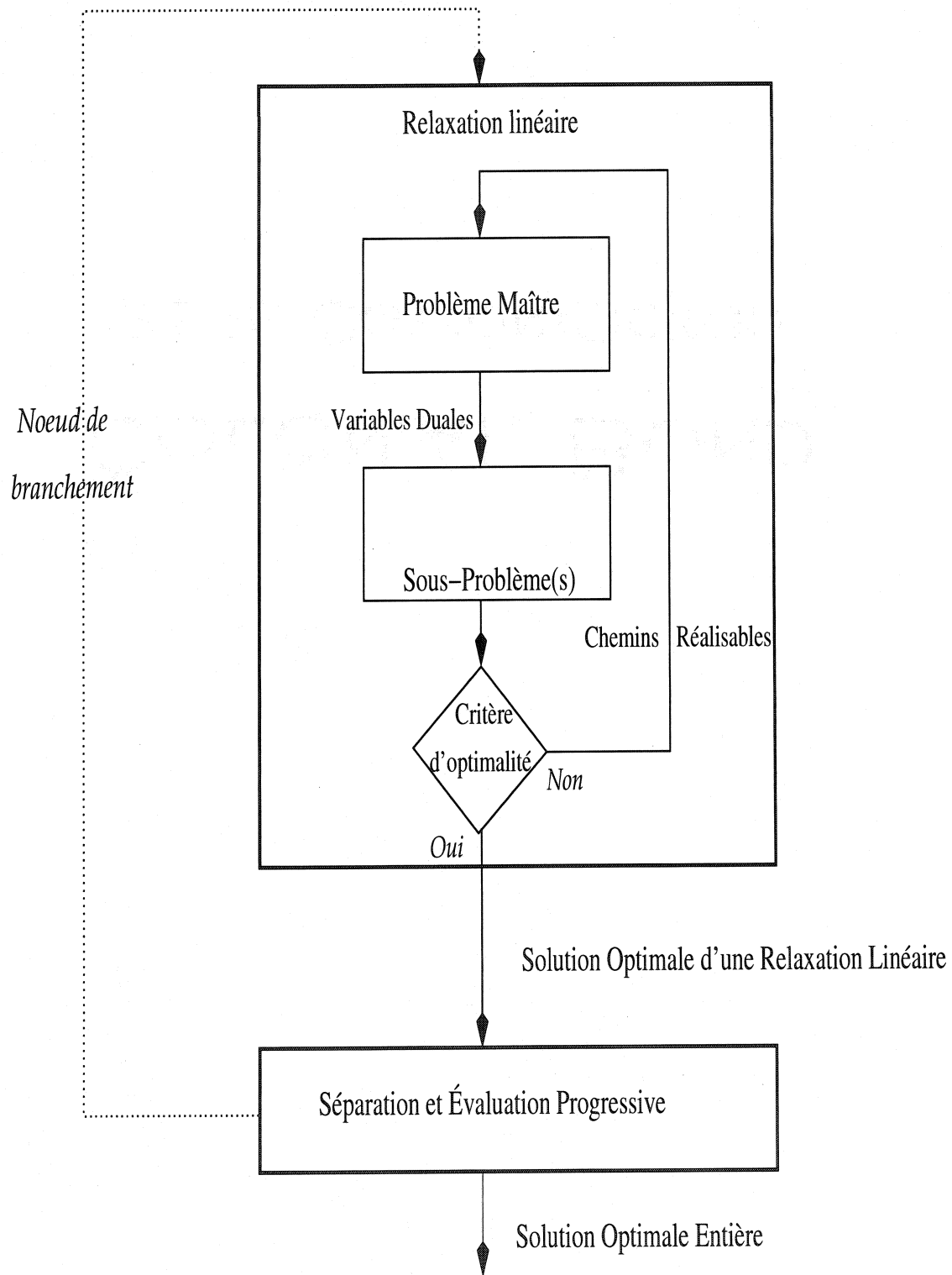


Figure 2.4 – Méthode de résolution

À chaque itération j de la méthode de génération de colonnes, une version restreinte du problème maître est résolue par la méthode du simplexe. Ce problème maître restreint ne considère qu'un petit sous-ensemble des variables de chemin : $\Omega_j^k \subseteq \Omega^k$. La solution courante du problème maître fournit l'information duale aux sous-problèmes. Cette information permet de déceler l'existence de variables de chemin de coût réduit négatif, parmi les variables de chemin non-considérées. Les sous-problèmes ont pour rôle alors de vérifier l'optimalité de la solution courante pour le problème maître et d'autre part, de générer de nouvelles variables à coût réduit négatif si l'optimalité n'est pas atteinte. La solution est déclarée optimale si aucune variable de coût réduit négatif ne peut être générée. Dans le cas contraire, le problème maître restreint est à nouveau résolu en ajoutant au moins une nouvelle variable de chemin et le processus se poursuit de cette façon jusqu'à ce que le critère d'optimalité soit atteint.

2.3.2 Recherche d'une solution entière

Si la solution optimale de la relaxation linéaire du problème n'est pas entière, un algorithme de séparation et évaluation progressive est alors utilisé. Par conséquent, la méthode de génération de colonnes est appliquée à chaque noeud de branchement dans le but d'obtenir une borne inférieure à ce noeud. L'exploration de l'arbre se réalise en « profondeur d'abord » jusqu'à ce qu'une bonne solution soit trouvée, puis ensuite continue « au meilleur d'abord » afin de trouver la meilleure solution entière. L'exploration de l'arbre s'arrête lorsqu'une solution entière trouvée est prouvée optimale.

Des plans coupants sont aussi utilisés pour obtenir des solutions entières. Ainsi quand un nombre fractionnaire n_c de chauffeurs utilisés est obtenu à la fin d'un noeud de branchement, une nouvelle contrainte est imposée de la manière suivante :

$\sum_{k \in K} \sum_{p \in \Omega^k} \theta_k \geq \lceil n_c \rceil$. En effet, on sait qu'il n'y aura pas de solution entière qui utilisera

un nombre de chauffeurs inférieur à $\lceil n_c \rceil$. Par conséquent, on peut imposer que le nombre de chauffeurs soit supérieur ou égal à $\lceil n_c \rceil$.

Pour l'algorithme de séparation et évaluation progressive, les règles de séparation sont définies à l'aide des variables $X_{ij} = \sum_{k \in K} \sum_{p \in \Omega_{ij}^k} \theta_p^k$ où Ω_{ij}^k est l'ensemble des journées de travail de type k contenant la tâche i telle que la tâche j est effectuée immédiatement après i . Une tâche représente pour le PSTAJT soit le parcours d'un segment de voyage, soit l'action d'amener (retirer) un autobus avant (après) un terminus. Ces règles scindent l'ensemble des solutions en deux sous-ensembles disjoints en s'aidant de la règle de Ryan et Foster (1981). Les inter-tâches de la modélisation représentent un cas particulier de cette règle : deux tâches doivent être réalisées consécutivement par le même chauffeur ou non. Après la résolution de la relaxation linéaire à un noeud de branchement qui nécessite une séparation, la variable X_{ij} dont la valeur est la plus près de 1, est d'abord choisie. La règle de séparation impose ensuite que la valeur de la variable choisie soit 1 (premier ensemble) et impose 0 pour le deuxième ensemble.

Ces décisions se répercutent au niveau de la résolution des sous-problèmes. En effet, pour une décision imposée $X_{ij} = 1$ la prolongation d'une étiquette associée à un chemin couvrant i comme dernière tâche ne peut se faire vers un arc de tâche associé à une tâche autre que j . De même, il n'est pas possible de prolonger sur un arc de tâche correspondant à j une étiquette associée à un chemin couvrant comme dernière tâche $t \neq i$. De plus, cette décision permet de façon indirecte d'éliminer dans le problème maître une contrainte de couverture de tâche. En effet, par exemple, pour les contraintes de couverture des tâches i et j , la décision de branchement assure que si l'une des deux tâches est couverte alors l'autre le sera automatiquement. Par conséquent, l'une des deux contraintes de couverture se trouvant dans le problème maître peut être éliminée. Au contraire, pour une décision imposée $X_{ij} = 0$ la prolongation d'une étiquette associée à un chemin couvrant i comme dernière tâche ne peut pas

se faire vers un arc associé à la tâche j . Il est à noter qu'en présence de contraintes d'inter-tâches, la dominance des étiquettes est restreinte et nécessite l'utilisation de listes d'étiquettes telles que décrites dans Haase *et al.* (2001).

La méthode et la modélisation de base présentées dans ce chapitre nécessitent de grands temps pour la résolution des sous-problèmes. En effet, le fait que certaines ressources utilisées au sein du réseau s'opposent, ralentit de beaucoup la résolution des sous-problèmes. Ceci nous amène alors à proposer une stratégie d'accélération des sous-problèmes.

Chapitre 3

ACCÉLÉRATION DE LA RÉSOLUTION DES SOUS-PROBLÈMES

La stratégie d'accélération présentée dans ce chapitre repose sur des idées issues de Freling *et al.* (1999, 2003). Elle vise à accélérer la résolution des sous-problèmes en générant des pièces de travail, au lieu de journées de travail, pour ensuite les combiner pour former des journées de travail comportant plus d'une pièce. Ces types de journées de travail sont structurés de la manière suivante : pièce + bris + pièce + bris + pièce ... Avec cette stratégie, au niveau de la résolution d'un sous-problème, la recherche des plus courts chemins correspond en fait à trouver toutes les pièces de travail de plus petit coût réduit. Le reste de la méthodologie proposé au chapitre précédent demeure le même.

3.1 Réseau générique

Le réseau générique mis en place a pour but de générer des pièces de travail sans les déplacements à pied et les voyages à vide du début et de la fin de la pièce (possibilités concernant le garage ou le stationnement). Le fait d'éliminer les cas précédents aboutit à un calcul plus rapide des plus courts chemin au cours de la résolution des sous-problèmes, dû principalement à la diminution du nombre de possibilités. Ce réseau ne se préoccupe pas du type de journée et des bris. En effet ces aspects n'ont

pas lieu d'être pris en compte à ce niveau, le réseau ayant pour but simplement de générer des pièces de travail.

De la même manière que dans la modélisation précédente, des sous-réseaux sont mis en place pour chaque heure de départ possible. Le noeud « puits » est commun à tous les sous-réseaux, mais dans la nouvelle modélisation, une seule ressource est nécessaire afin de gérer la contrainte de la durée maximale d'une pièce. Les autres ressources n'ont plus lieu d'être car les contraintes reliées à ces ressources ne sont plus présentes. Il est à remarquer que pour éviter l'utilisation d'une ressource, les sous-réseaux auraient pu avoir leurs propres noeuds « source » et leurs propres noeuds « puits ». Le noeud « source » étant construit à une heure h , le noeud « puits » serait alors construit à une heure $h' = h + PcMax$ avec $PcMax$ la durée maximale d'une pièce de travail. Mais cette version prendrait beaucoup plus de mémoire dû au réseau bien plus grand (multiplication des arcs d'arrivée au noeud « puits ») pour un temps de résolution identique. En effet, la dominance ne pose aucun souci avec une seule ressource, car les consommations de ressources se font de manière identique sur l'ensemble du réseau.

Chaque arc qui débute une pièce de travail se voit attribuer un coût de un. Le coût réduit total d'une journée de travail est en fait la somme des coûts réduits des pièces qui la compose. Mais ce coût total sera ajusté par la suite lors de la combinaison des pièces de travail (voir algorithme 3.3).

Ainsi mis en place, ces réseaux permettent plus facilement de trouver les pièces de travail sans les voyages à vide du début et de la fin de la pièce. La section suivante présente la méthode de génération de pièces de travail réalisables et de journées de travail réalisables à partir des plus courts chemins des réseaux décrits.

3.2 Création des journées de travail réalisables

Ayant des chemins réalisables entre les différents terminaux des réseaux précédemment décrits, un algorithme doit être mis en place dans le but de créer des pièces de travail réalisables dans un premier temps et des journées de travail réalisables dans un dernier temps.

L'algorithme 3.1 est inséré au niveau d'une itération au sein de la technique de génération de colonnes. Cette itération produit la nouvelle valeur de l'objectif en résolvant une série de sous-problèmes (problèmes de plus court chemin). Pour chacune de ces résolutions, différents plus courts chemins sont calculés à partir d'une liste d'étiquettes propres à chacun de ces sous-réseaux. Cette liste d'étiquettes représente en réalité différentes possibilités de parcours de ce sous-réseau. Une série de chemins entre les noeuds « sources » et le noeud « puit » est obtenue, avec laquelle seront construites les véritables pièces et journées de travail dans la fonction « ConstruireChemin(*chemin_base*) » (ligne 7). Enfin pour chaque journée de travail réalisable trouvée au sein de cet algorithme sera créée une colonne correspondant à une journée de travail réalisable (ligne 8–9).

Algorithme 3.1 Intégration de l'algorithme au sein de la génération de colonnes

```

1: while critère d'arrêt n'est pas respecté do
2:   ListeUnePiece ← NULL
3:   for all Sous-réseau ∈ Ensemble des Sous-Réseaux do
4:     for all étiquette ∈ Ensemble des étiquettes du sous-réseau do
5:       // Calcul du plus court chemin
6:       chemin_base ← pcc(étiquette)
7:       ListeChemin ← ConstruireChemin(chemin_base)
8:       for all chemin ∈ ListeChemin do
9:         constructionColonnes(chemin)

```

La fonction « ConstruireChemin(*chemin_base*) » s'exécute à partir d'un chemin du

réseau générique (voir Algorithme 3.2). Ce chemin décrit une succession réalisable de segments de voyage par un chauffeur. Mais les possibilités précédant le terminus de départ ainsi que celles succédant le terminus d'arrivée ne sont pas encore prises en compte. En effet, à partir de ce chemin de base il est possible de construire en fait neuf véritables pièces de travail (ligne 7). En prenant comme notation « Terminus D $\rightarrow$ Terminus A » le chemin de base, les neuf possibilités sont les suivantes :

1. (À pied) Terminus D $\rightarrow$ Terminus A (À pied).
2. (À pied) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au garage).
3. (À pied) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au stationnement).
4. (Voyage à vide du garage) Terminus D $\rightarrow$ Terminus A (À pied).
5. (Voyage à vide du garage) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au garage).
6. (Voyage à vide du garage) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au stationnement).
7. (Voyage à vide du stationnement) Terminus D $\rightarrow$ Terminus A (À pied).
8. (Voyage à vide du stationnement) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au garage).
9. (Voyage à vide du stationnement) Terminus D $\rightarrow$ Terminus A (Voyage à vide jusqu'au stationnement).

Les possibilités précédentes ne sont pas nécessairement valables à chaque fois. En effet, les trajets jusqu'au stationnement ne sont pas toujours possibles ou encore les durées ou les heures de départ et d'arrivée des pièces de travail ne respectent pas forcément les contraintes de la convention collective (ligne 8). Après cette vérification, les paramètres concernant la pièce réalisable sont ajustés suivant la possibilité i en cours (lignes 10–13).

Ensuite avant de construire véritablement les pièces de travail, celles-ci doivent répondre aux exigences requises par le type de journée. Dans le cas où celles-ci ne comportent qu'une seule pièce de travail (lignes 20–24), les exigences seront au niveau du coût réduit (devant être strictement négatif) et au niveau des paramètres de temps : ils doivent correspondre aux contraintes de la journée de travail concernée. Puis quand le type de journée requiert plusieurs pièces de travail avec un bris minimum entre chaque pièce (lignes 17–19), la pièce est stockée dans une liste qui servira par la suite à générer les journées de travail comportant plusieurs pièces (ligne 29). Il est à noter que dans ce cas bien précis tous les coûts réduits des pièces sont acceptables. En effet pour la construction des journées de travail comportant plusieurs pièces, la contrainte porte sur la somme des coûts réduits de toutes les pièces qui composent la journée. De ce fait, les pièces à coût réduit positif sont admissibles à être insérées dans la liste. Après avoir passé en revue tous les types de journées possibles puis toutes les possibilités d'amener ou de retirer les autobus aux terminaux, l'algorithme 3.2 retourne une liste de journées de travail réalisables comportant soit une ou plusieurs pièces (ligne 34).

Algorithme 3.2 ConstruireChemin(*chemin*) : fonction générant des pièces de travail

```

1: coûtRéduit  $\leftarrow$  chemin.coûtRéduit
2: durée  $\leftarrow$  chemin.durée
3: h_départ  $\leftarrow$  chemin.h_départ, h_fin  $\leftarrow$  chemin.h_fin
4:
5: for all t_duty  $\in$  Duty_Types do
6:   for all i  $\in$  Possibilits do
7:     // Fonction regardant la validité d'un chemin suivant le cas n°i
8:     if ValideChemin(chemin, i) then
9:       //Réajustement des variables
10:      coûtRéduit  $\leftarrow$  coûtRéduit + coûtRéduit[i]
11:      durée  $\leftarrow$  durée + durée[i]
12:      h_départ  $\leftarrow$  min(h_départ + h_départ[i])
13:      h_fin  $\leftarrow$  min(h_fin + h_fin[i])
14:
15:      if durée, h_départ, h_fin conformes au type de duty
16:      ET chemin  $\neq$  NULL then
17:        if t_duty.NombrePiece  $\geq$  2 then
18:          élément.créer()
19:          SousListeUnePièce  $\leftarrow$  SousListeUnePièce  $\cup$  élément
20:        else
21:          if coûtRéduit  $<$  0
22:          ET durée respecte le temps de travail then
23:            chemin_valide.create()
24:            ListeChemin  $\leftarrow$  ListeChemin  $\cup$  chemin_valide
25:        else
26:          effacer(chemin)
27:
28: //Fonction pour combiner les pièces de la liste
29: SousListeChemin  $\leftarrow$  GenererPath2PC(SousListeUnePièce, ListeUnePièce)
30:
31: ListeChemin  $\leftarrow$  ListeChemin  $\cup$  SousListeChemin
32: ListeUnePièce  $\leftarrow$  ListeUnePièce  $\cup$  SousListeUnePièce
33:
34: retourne ListeChemin

```

L'algorithme 3.3 s'efforce de trouver toutes les combinaisons possibles de pièces de travail pour construire l'ensemble des journées de travail réalisables à plusieurs pièces.

Cet algorithme de complexité $O(n^2)$ est difficilement améliorable du fait de la double combinaison possible pour chaque paire de pièces. Par conséquent, effectuer un tri au préalable de la liste des pièces n'est pas d'une grande utilité et augmente même dans certains cas la complexité de l'algorithme (dû surtout à la complexité plus grande pour l'insertion d'un élément au sein de la liste).

L'algorithme 3.3 va comparer toutes les pièces entre elles pour essayer de construire une journée de travail comportant plusieurs pièces. Cette procédure étudie juste le cas des journées de travail à deux pièces, car les types de journées rencontrées dans les jeux de données testés comportent au maximum deux pièces de travail. Il est à remarquer que cet algorithme peut facilement être généralisé pour des journées de travail comportant plus de deux pièces de travail (avec une complexité qui augmentera en conséquence). Pour pouvoir construire une journée de travail comportant plusieurs pièces, dans un premier temps il est nécessaire que la somme des durées des deux pièces (ligne 3) respectent les durées minimale et maximale de temps de travail spécifique au type de journée concerné. Dans un deuxième temps, l'algorithme considère deux cas : la pièce de travail A suivi d'un bris suivi de la pièce B (lignes 4-9), et la pièce de travail B suivi d'un bris suivi de la pièce A (ligne 12-16).

Pour que chacune de ces deux successions soit valide, les paramètres de bris doivent respecter certaines contraintes : les heures de bris doivent être incluses dans l'intervalle de temps prévu pour les bris et la durée du bris doit être supérieure à une certaine valeur minimale. La durée globale de la journée de travail doit nécessairement être inférieure à l'amplitude maximale de travail spécifiée dans la convention collective. La dernière exigence concerne le coût réduit total des deux pièces qui doit être strictement négatif. Il est à noter qu'il est inclus dans ce coût réduit total les coûts réduits des arcs de bris reliant les deux pièces. Enfin est soustrait à ce coût réduit total le nombre un, afin de compenser le coût de un (qui a pas lieu d'être dans les autres pièces de la journée de travail) du premier arc de la deuxième pièce. Après

avoir réalisé les différentes comparaisons entre toutes les pièces, l'algorithme va alors retourner un ensemble de journées de travail réalisables à deux pièces.

Algorithme 3.3 Fonction GenererPath2PC(SousListeUnePiece,ListeUnePiece)

```

1: for all élément  $A \in ListeUnePièce$  do
2:   for all élément  $B \in SousListeUnePièce$  do
3:     if  $A.durée + B.durée$  conforme à la durée de travail des journées à 2 pièces
       then
4:       // Cas 1 :  $A \rightarrow B$ 
5:       if paramètres bris respectés
6:       ET  $A.coûtRéduit + B.coûtRéduit - 1 < 0$ 
7:       ET  $A.durée + B.durée + durée\_bris \leq MaximumAmplitude$  then
8:          $chemin\_valide.créer(A,B)$ 
9:          $SousListeChemin \leftarrow SousListeChemin + chemin\_valide$ 
10:
11:      // Cas 2 :  $B \rightarrow A$ 
12:      if paramètres bris respectés
13:      ET  $A.coûtRéduit + B.coûtRéduit - 1 < 0$ 
14:      ET  $A.durée + B.durée + durée\_bris \leq MaximumAmplitude$  then
15:         $chemin\_valide.créer(B,A)$ 
16:         $SousListeChemin \leftarrow SousListeChemin + chemin\_valide$ 
17:
18: retourne SousListeChemin

```

Cet ensemble d'algorithmes qui repose sur des idées de Freling *et al.* (1999, 2003) permet de faire la jonction entre les solutions issues des réseaux mis en place et les méthodes de résolution présentées dans le chapitre précédent qui seront aussi utilisées de la même façon pour ce modèle.

3.3 Résultats

Afin de valider et comparer les deux méthodologies de résolution du PSTAJT présentées dans les chapitres précédents, des tests ont été effectués sur des données réelles

fournis par la compagnie Giro Inc. Cette section présente ces données et les résultats numériques obtenus par les deux méthodologies. Il est à noter que l'implantation des différents algorithmes s'est faite en prenant comme noyau central le logiciel Gencol et le logiciel Cplex 7.5. Les tests ont été effectués sur une machine 900 Mhz.

3.3.1 Description des jeux de données

Le jeu de données fourni par l'entreprise Giro Inc. comporte un ensemble de 463 voyages à couvrir sur la ligne 68 d'une ville méditerranéenne durant une journée. Une solution manuelle a été fournie ne satisfaisant pas toutes les règles. Nous avons construit différents jeux de données à partir de cet ensemble de voyages en choisissant les voyages couverts par un sous-ensemble des journées de travail. Dans ce qui suit chaque jeu de données est identifié à l'aide d'un identificateur du type Pv_j où v indique le nombre de voyages et j le nombre minimum de journées de travail trouvé par la société de transport en commun de la ville méditerranéenne pour couvrir les voyages du jeu de données. Par exemple, $P56_6$ correspond à un jeu de données comportant 56 voyages couverts par 6 journées de travail.

Les jeux de données comportent une seule ligne munis d'un terminus A et d'un terminus B. Cette ligne possède par ailleurs un seul point de relève en direction de A vers B et un seul point de relève en direction de B vers A. Il est à noter que pour les jeux de données de Giro Inc., le temps de battement maximal à un terminus est de vingt minutes. Concernant les voyages à vide, les paramètres fournis dans les jeux de données permettent de connaître la validité des différentes possibilités pour aller ou partir du garage ou du stationnement. Par exemple dans les jeux de données actuels, le déplacement d'un autobus du terminus B vers le stationnement n'est pas possible. Ensuite pour les déplacements valides leurs temps sont fournis suivant le moment de la journée.

Les différents problèmes sont résolus dans le but de fournir un ensemble de journées de travail. Mais ces journées de travail doivent respecter un certain nombre de contraintes portant notamment sur les types de journées auxquelles elles appartiennent. Pour les problèmes fournis par Giro Inc., trois types de journées sont mis en jeu : ceux de type “AM” (journées se déroulant le matin), ceux de type “PM”(journées se déroulant l’après midi) et ceux de type “2PCE”(journées comportant 2 pièces de travail avec un bris entre les 2 pièces). Le tableau 3.1 présente leurs spécificités.

Tableau 3.1 – Caractéristiques des différents types de journées de travail

Type	AM	PM	2PCE
H. début min.	5h00	12h30	00h00
H. fin max.	16h30	24h00	24h00
Amplitude max. (min)	–	–	720
Tps travail min. (min)	410	410	425
Tps travail max. (min)	448	448	448
Tps pièce min. (min)	–	–	118
Tps pièce max. (min)	–	–	370
Nombre de bris	0	0	1
H. début bris min.	–	–	11h00
H. fin bris max.	–	–	16h30

Chacun des problèmes à résoudre présente donc une liste de voyages à couvrir. Chaque voyage se définit par un lieu et une heure de départ ainsi qu’une heure d’arrivée. Le lieu d’arrivée se déduit facilement puis qu’il n’existe que deux terminus. Comme dans chaque jeu sont donnés les temps de parcours de chaque segment de voyage suivant l’heure de la journée, il est possible de déduire les heures de départ et d’arrivée de chacun des segments de voyage qui composent les voyages du problème. On retrouve ainsi annexé à chacun de ces jeux de données, tous les paramètres nécessaires pour résoudre correctement les problèmes. Les premiers tests furent réalisés avec le modèle de base.

Le tableau 3.2 présente la taille des différents jeux de données avec le modèle de base. Ce tableau présente pour chaque instance (Id Pb), le nombre de tâches à couvrir (Cont. Tâche), le nombre de contraintes de consommation de flot (Cont. cons. flot), le nombre de sous-réseaux (Réseaux), le nombre de noeuds (Noeuds) et le nombre d'arcs (Arcs). Il est à préciser que dans la construction des réseaux beaucoup d'arcs sont en commun notamment ceux qui modélisent les segments de voyages.

Tableau 3.2 – Tailles des problèmes avec le modèle de base

Id Pb	Cont. Tâche	Cont. cons. flot	Réseaux	Noeuds	Arcs
P56_6	56	254	315	522	940
P58_6	58	263	305	505	979
P63_7	63	285	354	595	1101
P85_9	85	384	400	723	1532
P114_12	114	515	550	949	1965
P119_13	119	537	582	1019	2110
P121_13	121	546	586	1019	2098
P141_15	141	636	640	1156	2528
P143_15	143	645	642	1153	2614
P148_16	148	667	648	1191	2759
P177_19	177	798	765	1395	3157
P199_21	199	897	822	1528	3653
P204_22	204	919	829	1577	3839
P206_22	206	928	835	1569	3861
P262_28	262	1180	973	1908	4980
P286_31	286	1288	953	2009	5813
P315_35	315	1419	1039	2180	6492
P344_37	344	1549	1090	2335	7210
P463_50	463	2084	1238	2887	9965

Ce tableau montre bien les proportions imposantes que peuvent prendre les plus gros problèmes. Cela permet déjà de justifier la simplification nécessaire du réseau mise en place dans le second modèle.

3.3.2 Résultats pour le modèle de base

Les jeux de données présentés dans la section précédente ont été résolus par le logiciel Gencol dont quelques fonctions ont été spécialisées pour répondre aux exigences du modèle de base. La résolution de tous les problèmes a été effectuée de la même manière et avec le même ajustement des paramètres de résolution de Gencol. Aussi il est possible au cours des itérations de la méthode de génération de colonnes de passer d'un modèle de résolution à un autre. Au niveau de ces modèles on peut y modifier par exemple le nombre de ressources pris en compte lors de la résolution et par conséquent modifier le nombre de contraintes reliées à ces ressources. Pour le modèle de base nous avons choisi de le résoudre séquentiellement avec cinq modèles. Les quatre premiers modèles comportent chacun quelques ressources et par conséquent les contraintes reliées à ces ressources. Le dernier modèle de résolution possède quant à lui toutes les contraintes de ressources. De ce fait, avec les quatre premiers modèles, la dominance s'effectue sur un sous-ensemble des ressources et les dernières itérations de la résolution tiennent compte de toutes les contraintes pour aboutir à une solution optimale.

Le tableau 3.3 présente les résultats obtenus sur les plus petits jeux de données : ceux des plus gros jeux de données n'ont pas été insérés dans ce tableau dû aux temps excessifs de résolution. Pour chaque instance (Id Pb), sont présentées la valeur de la solution obtenue en résolvant la relaxation linéaire, le nombre d'itérations de la génération de colonnes (Nb Iter), le nombre moyen de colonnes ajoutées au problème maître par itération (Nb Col/Iter), le temps de résolution du problème maître (CPU PM) en minutes, le temps de résolution des sous-problèmes (CPU SP) en minutes, et le temps de résolution totale (CPU total) en minutes.

On peut constater d'abord qu'il y a un gain de un chauffeur sur certains jeux de données indiqués en gras : notamment le jeu de données comportant 85 voyages.

Tableau 3.3 – Résultats de l'approche de base

Id Pb	Z_{LP}	Nb Iter	Nb Col/Iter	CPU PM (min)	CPU SP (min)	CPU total (min)
P56_6	6	39	199	<1	<1	<1
P58_6	6	39	162	<1	<1	<1
P63_7	7	43	150	<1	<1	<1
P85_9	8	85	192	2	1	3
P114_12	12	83	194	3	4	7
P119_13	13	121	181	3	41	44
P121_13	13	67	191	2	2	4
P141_15	14	98	233	6	46	52
P143_15	14	124	221	8	13	21
P148_16	15	97	218	5	45	50
P177_19	19	109	267	18	54	72
P199_21	20	172	244	23	341	364
P204_22	21	129	263	17	227	244
P206_22	21	130	227	25	166	191
P262_28	27	175	283	47	1516	1563

Les autres jeux de données possédant cet ensemble de voyages ont par conséquent un gain de un chauffeur. On peut remarquer qu'au niveau des plus gros problèmes, le temps de résolution des sous-problèmes est beaucoup plus grand que celui passé au niveau du problème maître. Cela est surtout dû aux problèmes de dominance sur les noeuds de bris, rencontrés au niveau du calcul du plus court chemin. Ce souci justifie la construction du second modèle sur lequel toute l'attention a été portée au niveau de la résolution des sous-problèmes.

3.3.3 Résultats avec accélération de la résolution des sous-problèmes

Pour résoudre les différents problèmes avec le second modèle, il a fallu insérer les algorithmes présentés dans la section précédente au sein de Gencol, au niveau de la résolution des sous-problèmes. Le reste de la résolution demeure semblable au modèle précédent mis à part quelques ajustements au niveau de certains paramètres de Gencol. Aussi comme nous le montre le tableau 3.4, le modèle hybride permet de résoudre dans un temps raisonnable de bien plus gros problèmes. De ce fait, la taille de nos jeux de données varie beaucoup. Les jeux de données ont donc été scindés en deux ensembles : les petits problèmes et les gros problèmes. Les paramètres de Gencol diffèrent d'un ensemble à un autre afin que les méthodes de résolution s'adaptent à la taille du problème rencontré.

Au vu des résultats qui apparaissent dans le tableau 3.4, il est clair que les temps de résolutions sont nettement améliorés : aussi bien que sur le temps passé dans le problème maître que celui passé à résoudre les sous-problèmes. Mais quand on compare les jeux de données résolus à l'aide des deux modèles, on s'aperçoit que le temps gagné se situe surtout au niveau de la résolution des sous-problèmes. Ce gain est encore plus visible au vu des résultats présentés dans le tableau 3.5. Celui-ci nous montre les statistiques obtenues sur la résolution heuristique des plus gros problèmes. Cette méthode de résolution heuristique permet d'enlever les dernières itérations de la résolution, très coûteuses au niveau du temps passé dans les sous-problèmes.

Après avoir bien amélioré les temps de résolution des sous-problèmes, une méthode de stabilisation des variables duales présentée dans la section suivante va être insérée au dernier modèle pour diminuer le temps passé dans le problème maître.

Tableau 3.4 – Résultats avec accélération de la résolution des sous-problèmes

Id Pb	Z_{LP}	Nb Iter	Nb Col /Iter	CPU PM (min)	CPU SP (min)	CPU total (min)	Gain CPU (%)
P56_6	6	31	90	<1	<1	<1	0
P58_6	6	58	28	<1	<1	<1	0
P63_7	7	42	65	<1	<1	<1	0
P85_9	8	154	133	<1	<1	1	66
P114_12	12	99	75	1	1	2	71
P119_13	13	75	133	1	1	2	95
P121_13	13	105	82	1	1	2	50
P141_15	14	95	136	3	2	5	90
P143_15	14	235	74	4	3	7	63
P148_16	15	101	141	4	3	7	86
P177_19	19	147	157	6	4	10	86
P199_21	20	167	166	13	13	26	93
P204_22	21	129	289	9	17	26	89
P206_22	21	245	219	14	16	30	84
P262_28	27	263	338	35	33	68	96
P286_31	29	210	297	34	65	99	–
P315_35	33	239	298	43	99	142	–
P344_37	35	303	351	90	148	238	–
P463_50	48	382	435	273	389	662	–

Tableau 3.5 – Résultats heuristique de l'approche avec accélération de la résolution des sous-problèmes pour les gros problèmes

Id Pb	Z_{LP}	Nb Iter gen	Nb Col/Iter	Tps PM min	Tps SP min	Tps total min
P177_19	19.1	138	158	6	2	8
P199_21	20	162	170	13	5	18
P204_22	21	128	292	10	10	20
P206_22	21.4	164	244	11	4	15
P262_28	27.1	239	333	32	10	42
P286_31	29	184	336	34	18	52
P315_35	33	209	333	41	12	53
P344_37	35	275	380	86	31	117
P463_50	48	337	476	268	80	348

Chapitre 4

STABILISATION DES VARIABLES DUALES

Il est bien connu que les méthodes de génération de colonnes présentent souvent une instabilité au niveau de l'ajustement des variables duales. En effet, les valeurs prises par ces variables peuvent changer considérablement d'une itération à l'autre comme le montre la figure 4.1. Chaque courbe de cette figure décrit l'évolution de la valeur d'une variable duale au cours des itérations de génération de colonnes. Ces grandes variations sont peu désirables car elles entraînent la génération de variables primales quelconques qui ont peu de chances de se retrouver dans une solution optimale. Il s'impose alors une mise en place d'un algorithme de stabilisation des variables duales qui est présenté dans ce chapitre. Dans un premier temps, nous justifions le choix de la méthode de stabilisation puis nous décrivons l'algorithme mis en place qui reposent sur des idées de Ben Amor (2002) et Ben Amor *et al.* (2003). Enfin les résultats qui découlent de cet algorithme sont présentés.

4.1 Choix du type de la stabilisation

Les approches de stabilisation peuvent être regroupées dans deux classes : les méthodes du type faisceaux et les méthodes du type proximal. Les méthodes de la première classe sont des méthodes de type faisceaux (Lemaréchal 1974) : elles visent

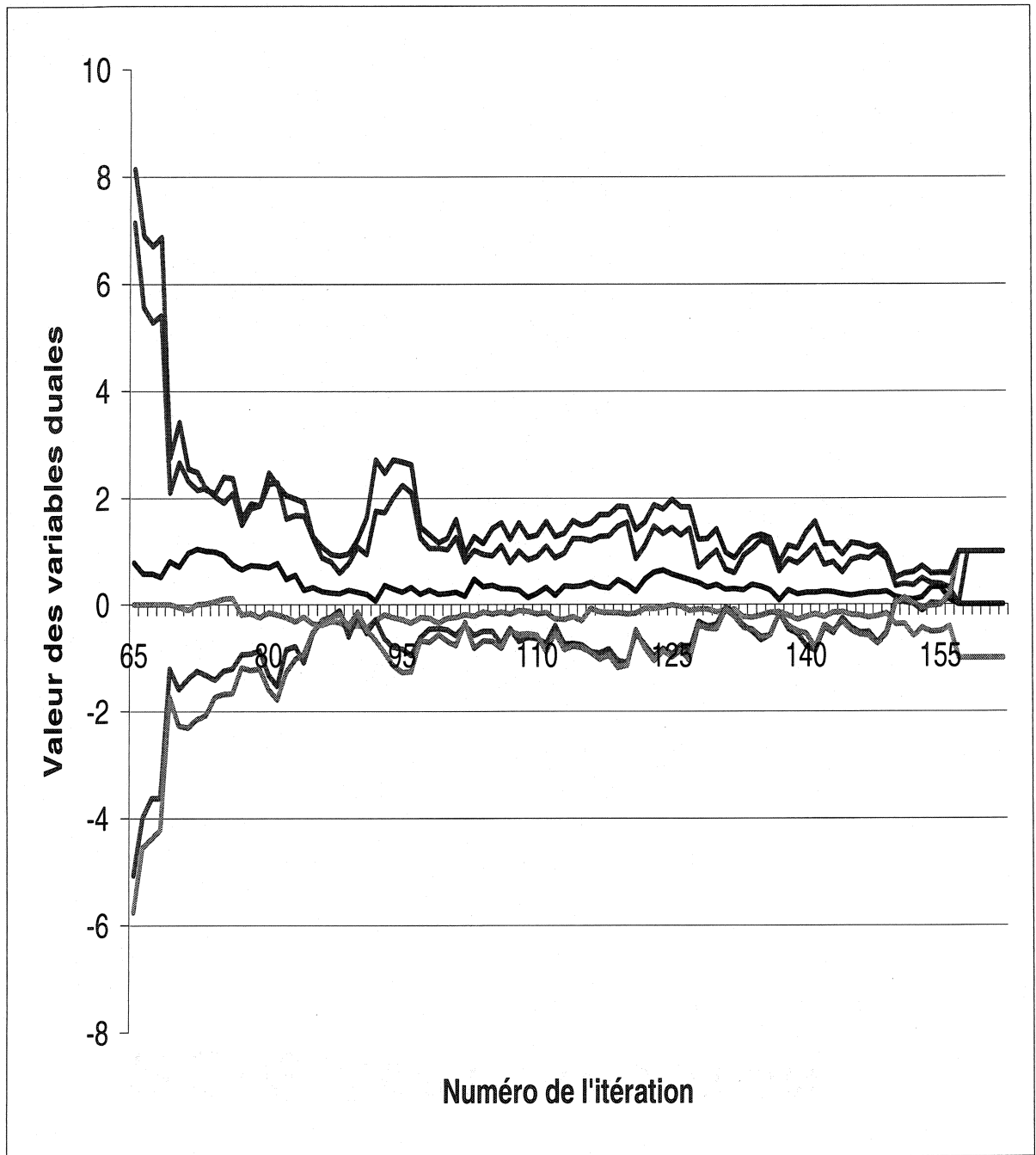


Figure 4.1 – Variation des valeurs de six variables duales

à stabiliser les méthodes de plans coupants en utilisant un centre de stabilité qui évolue au cours des itérations. À chaque itération, la fonction objectif duale contient une pénalité quadratique qui croît avec la distance entre la solution et le centre de stabilité. Lorsque la solution duale obtenue est meilleure que celle découlant du centre de stabilité, elle devient le nouveau centre de stabilité. Dans le cas contraire, le centre de stabilité ne change pas. Les méthodes de type proximal (par exemple Ben Amor *et al.* 2003) reposent quant à elles sur un algorithme du point proximal (Rockafellar 1976). Elles consistent aussi à ajouter un terme de pénalité dans la fonction duale. Toutefois, le point obtenu forme toujours le nouveau centre de stabilité, qu'il soit meilleur ou non.

La méthode que nous présentons se situe dans cette dernière classe. Celle-ci s'inspire de la méthode de stabilisation mise en place par Ben Amor (2002) et Ben Amor *et al.* (2003). La principale différence va porter sur le comportement de la valeur de l'objectif. En effet, Ben Amor *et al.* stabilisent les variables duales par plusieurs descentes de la valeur objectif du primal (en pointillés sur la figure 4.2). La méthode comporte alors des itérations majeures (mises à jour des paramètres de stabilité) et des itérations mineures (descente de la valeur objectif). À l'opposé, la méthode de stabilisation mise en place dans ce mémoire (en continu sur le graphe 4.2) s'efforce d'arriver à l'optimalité en une seule descente de la valeur de l'objectif. Pour ce faire, les mises à jour sont plus nombreuses et sont réalisées à l'aide de certains critères tout au long des itérations de génération de colonnes. À noter que la fonction objectif du primal n'est pas strictement décroissante au cours des itérations. Il arrive souvent que la valeur de l'objectif fluctue avant d'arriver à la valeur optimale. Par conséquent, à une itération donnée, il est possible que la valeur de l'objectif soit inférieure à celle de l'optimalité. Cela est rendu envisageable grâce à la présence de régions de pénalité.

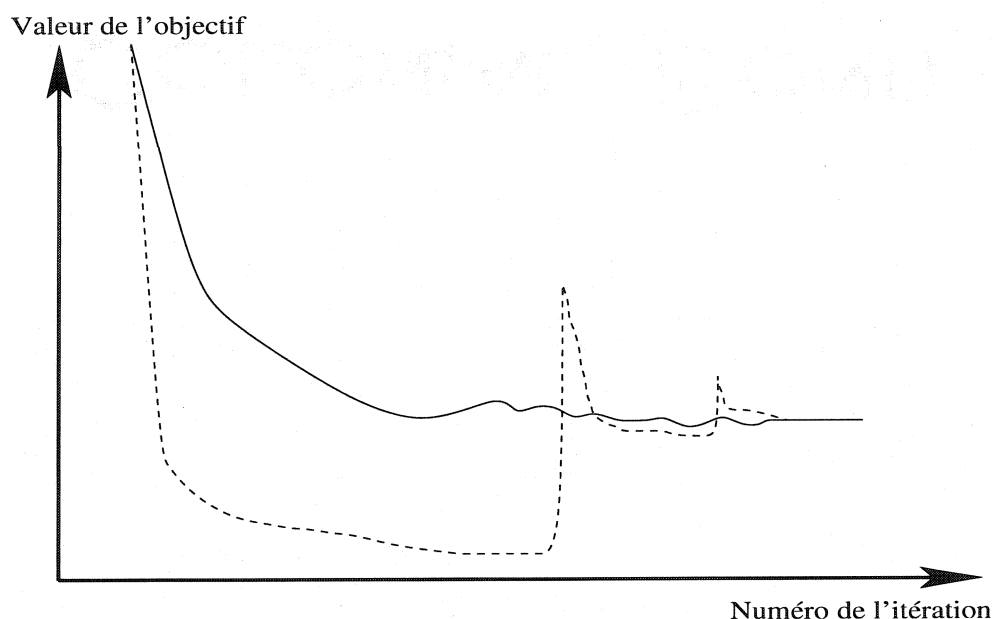


Figure 4.2 – Descente de l'objectif de deux méthodes proximales

4.2 Description de la méthode de stabilisation mise en place

Pour expliquer la mise en place de la stabilisation, il est préférable de repartir de la reformulation d'un problème primal :

$$\text{Minimiser } cx \tag{4.1}$$

$$\text{subject à : } Ax = b \tag{4.2}$$

$$x \geq 0. \tag{4.3}$$

En associant à la contrainte (4.2) le vecteur de variables duales π , le problème dual associé est alors :

$$\text{Maximiser } \pi b \quad (4.4)$$

$$\text{subject à : } \pi A \leq c. \quad (4.5)$$

Le but de la stabilisation est de borner les variables duales de la façon suivante : $d_1 - \omega_1 \leq \pi \leq d_2 + \omega_2$ (voir contraintes (4.8) et (4.9)). Les termes d_1 et d_2 sont les bornes tandis que ω_1 et ω_2 désignent la marge laissée aux variables duales par rapport aux bornes. Cela instaure alors une région de confiance (d_1, d_2) centrée en un point $\frac{d_1+d_2}{2}$ dans l'espace dual (figure 4.3). Dans le but de forcer les variables duales à rester dans cette région, un terme de pénalité $-\omega_1\epsilon_1 - \omega_2\epsilon_2$ est ajouté à la fonction objectif duale, qui pénalise dès que les variables duales quittent la région de confiance. Il en découle alors tout naturellement la formulation suivante du problème dual stabilisé.

$$\text{Maximiser } \pi b - \omega_1\epsilon_1 - \omega_2\epsilon_2 \quad (4.6)$$

subject à :

$$\pi A \leq c \quad (4.7)$$

$$-\omega_1 - \pi \leq -d_1 \quad (4.8)$$

$$-\omega_2 + \pi \leq -d_2 \quad (4.9)$$

$$\omega_1, \omega_2 \geq 0. \quad (4.10)$$

En associant aux contraintes (4.8)-(4.9) les variables primales s_1 et s_2 , le problème primal stabilisé correspondant est alors :

$$\text{Minimiser } cx - d_1s_1 + d_2s_2 \quad (4.11)$$

sujet à :

$$Ax - s_1 + s_2 = b \quad (4.12)$$

$$s_1 \leq \epsilon_1 \quad (4.13)$$

$$s_2 \leq \epsilon_2 \quad (4.14)$$

$$s_1, s_2, x \geq 0. \quad (4.15)$$

Il est à noter que pour construire cette région de confiance autour du centre, il est impératif que les δ_1 et δ_2 (avec $\delta_1 = \text{centre} - d_1$ et avec $\delta_2 = \text{centre} - d_2$) soient strictement positifs (voir figure 4.3). Dans le cas contraire, les variables duales resteraient identiques d'une itération à une autre au cours de l'algorithme de génération de colonnes. Par conséquent, l'algorithme arrêterait alors que l'optimalité ne serait pas obtenue. Enfin la fonction de pénalité aurait pu être construite à cinq morceaux ou plus, mais celle à trois morceaux mise en place est moins coûteuse en nombre de colonnes et suffit à borner correctement les variables duales.

La stabilisation des variables duales associées à chacune des contraintes de la modélisation du problème repose sur le principe d'acquérir au maximum de l'information duale au cours des itérations de génération de colonnes. Initialement, plusieurs informations sont présentes. La première information importante qui ressort de la figure 4.1, est le fait que les variables duales sont à l'optimalité toutes comprises entre 1 et -1. Cela s'explique facilement par le fait que le réseau qui modélise le PS-TAJT possède seulement des chemins réalisables de coût unitaire. Par conséquent, les centres de stabilisation seront nécessairement bornés entre 1 et -1.

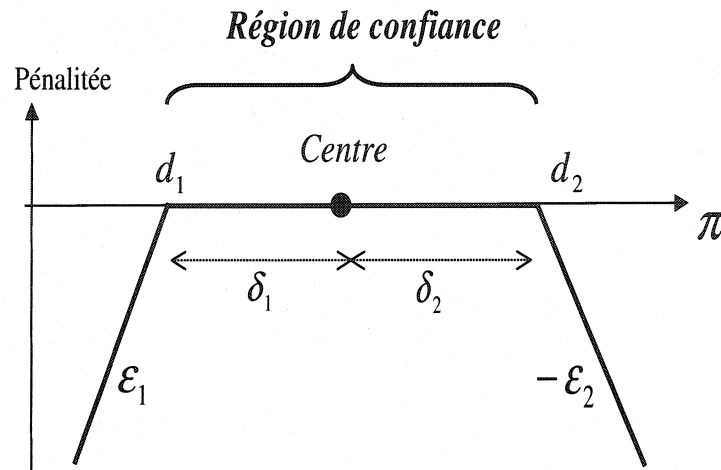


Figure 4.3 – Région de confiance dans l'espace dual

La seconde information concerne certaines tâches que l'on va pouvoir fixer dès le début de l'algorithme de génération de colonnes. L'idée ici est d'essayer de fixer (bornes très serrées) le plus possible des variables duales dès le début. Tout d'abord, il faut savoir que nécessairement dans un chemin faisant partie d'une solution optimale, l'une des variables duales associées aux tâches qui composent ce chemin prendra la valeur un. De ce fait, si la solution optimale nécessite par exemple vingt chauffeurs, alors vingt des valeurs duales optimales seront égales à un. Le but devient alors de trouver rapidement dès le départ, un ensemble maximum de tâches simultanées pour fixer leurs variables duales correspondantes à un. Pour ce faire, à partir de la répartition des voyages de la journée (figure 4.4), les tâches faisant partie de la coupe maximale (en pointillés) auront leurs variables duales fixées à un. Autrement dit, chacune de ses variables duales aura une région de confiance très restreinte autour de un et cette fixation aura lieu avant même la mise en route de l'algorithme de génération de colonnes.

Une fois ces informations duales bien intégrées initialement, il est nécessaire de mettre en place un algorithme dont le but est de contrôler les déplacements dans l'espace dual

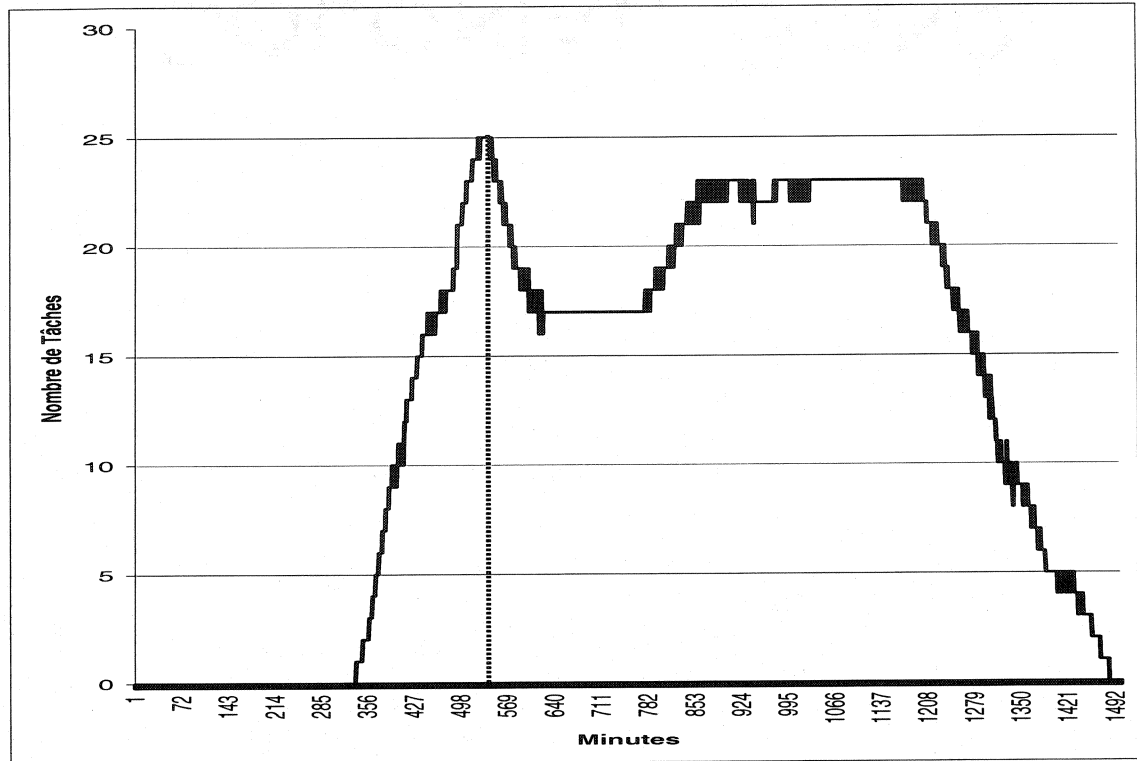


Figure 4.4 – Répartition des voyages sur une journée

au cours des itérations de génération de colonnes. Ainsi cet algorithme va construire et modifier des régions de confiance autour de chaque variable duale. Il est à noter que les régions de confiance ont été construites seulement sur les variables duales qui en valaient la peine : en effet chaque fonction de pénalité construite pour une variable duale requiert deux colonnes de plus dans le problème maître. De ce fait, l'algorithme de stabilisation (voir Algorithme 4.1) ne se préoccupe pas des variables duales associées aux contraintes relatives au stationnement (ces variables varient très peu).

L'algorithme 4.1 est utilisé pour chaque itération i de la génération de colonnes. Cet algorithme fait appel aux notations suivantes :

R : désigne l'ensemble des contraintes du PSTAJT pour lesquelles on veut utiliser

- un mécanisme de stabilisation de la variable duale associée ;
- r : est l'indice d'une contrainte qui identifie les composantes des vecteurs b , π_i , ϵ_i , δ_i et $centre_i$ à l'itération i ;
- $centre_i$: est le centre de stabilisation à l'itération i . Il est à noter que la valeur de $centre_i^r$ est initialisée à la valeur de la variable duale π^r ;
- δ_i^r et ϵ_i^r : désignent respectivement l'écart par rapport au centre de stabilisation $centre_i^r$ et la pente de la fonction de pénalité. Étant donné que la fonction de pénalité est construite symétriquement, on a $\delta_{1i}^r = \delta_{2i}^r = \delta_i^r$ et $\epsilon_{1i}^r = \epsilon_{2i}^r = \epsilon_i^r$. Il est à noter que la valeur de δ_i^r est initialisée à 0.9 et que la valeur de ϵ_i^r est initialisée à 0.01 ;
- Z_i : est une borne inférieure sur la valeur optimale recherchée et Z est la meilleure borne Z_i trouvée avant l'itération i . Il est à noter que Z a pour valeur initiale l'infini ;
- $Cmin_i$: désigne le plus petit coût réduit de l'ensemble des colonnes ajoutées dans le problème maître à l'itération i . Cette valeur donne une approximation du plus petit coût réduit sur l'ensemble de toutes les variables du problème puisque tous les sous-problèmes ne sont pas résolus à une itération i . L'algorithme demeure tout de même exact, car cette borne est seulement utilisée pour décider à quel moment réduire les valeurs des pentes et la taille de la région de confiance. Ces pentes et cette taille pourraient ne jamais être ajustées et l'algorithme serait quand même exact (mais plus long à converger) ;
- K : désigne une borne maximale sur le nombre de chauffeurs requis pour le problème. Cette valeur peut être facilement obtenue en divisant par exemple la somme des temps de tous les voyages sur le temps minimal requis pour effectuer une journée de travail ;
- α et β : sont des constantes déterminées qui ont été fixées à 1.1 et 1.5 respectivement. Ces constantes sont utilisées pour réduire la région de confiance et augmenter les pentes.

Algorithme 4.1 Algorithme de la stabilisation dynamique intégré pour chaque itération i

```

1: for all  $r \in R$  do
2:   if  $\pi_i^r \geq 0$  then
3:      $centre_i^r \leftarrow \min\{\pi_i^r, 1\}$ 
4:   else
5:      $centre_i^r \leftarrow \max\{\pi_i^r, -1\}$ 
6:    $Z_i \leftarrow \sum_{r \in R} b^{rT} \pi_i^r + C \min_i K$ 
7:   if  $Z_i > Z$  then
8:      $Z \leftarrow Z_i$ 
9:      $\delta_i^r \leftarrow \max\{\delta_i^r / \alpha, 0.1\}$ 
10:     $\epsilon_i^r \leftarrow \min\{\epsilon_i^r * \beta, 1.1\}$ 

```

Ainsi dans l'algorithme 4.1 quand la borne inférieure Z_i est supérieure à la précédente (ligne 2) une mise à jour complète des régions de confiance et des pentes est effectuée (lignes 9 à 10). La région de confiance autour du centre de pénalité va alors se resserrer (distance divisée par α) et les pentes de la fonction de pénalité vont augmenter (multipliées par β). Cela va aider les valeurs des variables duales à rester dans la région de confiance. Par contre, quand la borne inférieure n'est pas améliorée, les régions de confiance restent identiques de même que les pentes. Dans les deux cas, les centres de la région de confiance sont mis à jour (lignes 2 à 5) selon les valeurs duales de l'itération i . Cela évite une trop grande pénalité due à un fort décalage de la région de confiance par rapport à la valeur duale.

Il est à remarquer que les paramètres de la région de confiance (δ_i^r) et les pentes (ϵ_i^r) sont bornés au niveau de l'algorithme 4.1 de stabilisation. Cela évite d'avoir une région de confiance trop étroite autour du centre de stabilisation et d'avoir alors des variables ayant trop peu de marge de manoeuvre. Ces variables pourraient autrement se loger dans la région fortement pénalisée trop proche du centre, et c'est ce que l'on veut empêcher.

4.3 Résultats de la stabilisation

4.3.1 Relaxation linéaire

La figure 4.5 présente les effets de l'algorithme de stabilisation mis en place pour résoudre la relaxation linéaire d'une instance typique du PSTAJT. La courbe en pointillés décrit les variations d'une variable duale au cours des itérations de l'algorithme de génération de colonnes sans la méthode de stabilisation. Par contre, la courbe en continu représente les variations de cette même variable duale quand la méthode de stabilisation est intégrée à la résolution. Au premier coup d'oeil, on constate que la technique de stabilisation réduit la variabilité de la valeur de la variable duale au cours des itérations. On remarque aussi qu'avec la méthode de stabilisation, une solution réalisable est trouvée plus vite (gain d'une dizaine d'itérations). De manière générale, les différentes instances du PSTAJT sont résolues de façon exacte en moins d'itérations.

Le tableau 4.1 présente les résultats obtenus sur les huit plus grands jeux de données utilisés au chapitre 3. En effet, c'est bien sur les temps de résolution des plus grands jeux de données que se situe l'enjeu de ce mémoire et c'est sur ces jeux que la stabilisation des variables duales peut avoir le plus grand impact. On constate alors que l'utilisation de cette technique produit des gains sur les temps de résolution assez appréciables pour tous les jeux de données sauf pour l'instance P286_31. Ces gains sont attribuables en majeure partie à une réduction des temps de résolution du problème maître. En effet, si on les compare avec ceux du tableau 3.4, les temps de résolution du problème maître sont nettement inférieurs (en moyenne divisé par 3). Enfin on peut observer une réduction du nombre d'itérations pour chacun des jeux de données sauf pour l'instance P286_31. Cela peut s'expliquer par le fait qu'en fixant dès le départ certaines variables duales (associées aux tâches de la coupe maximale),

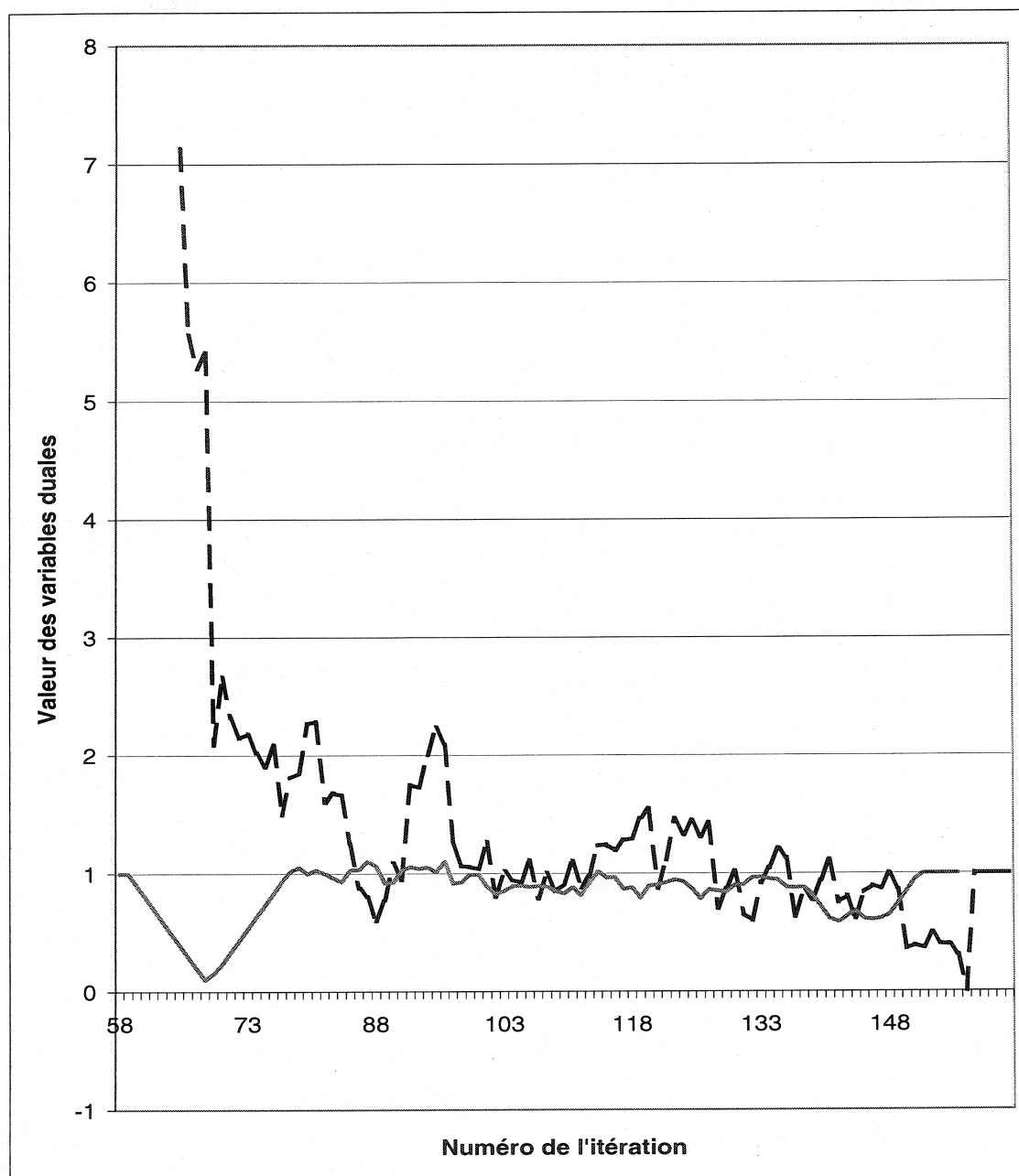


Figure 4.5 – Variation des valeurs d'une même variable duale pour une résolution avec stabilisation (trait plein) et sans stabilisation (trait pointillé).

l'algorithme de génération de colonnes réussit à trouver plus rapidement des solutions réalisables des différentes instances du PSTAJT.

Tableau 4.1 – Résultats avec stabilisation dynamique

Id Pb	Z_{LP}	Nb Iter	Nb Col /Iter	CPU PM (min)	CPU SP (min)	CPU total (min)	Gain CPU (%)
P199_21	20	156	116	3	9	12	54
P204_22	21	119	214	3	10	13	50
P206_22	21	173	161	4	10	14	53
P262_28	27	160	262	7	33	40	41
P286_31	29	231	287	21	94	115	-16
P315_35	33	213	262	19	54	73	49
P344_37	35	201	273	20	143	163	32
P463_50	48	333	324	93	418	511	23

Les gains sur les temps de résolution sont bien plus perceptibles lorsqu'une approche heuristique est utilisée (voir tableau 4.2). En effet, les dernières itérations très coûteuses en temps de résolution des sous-problèmes sont absentes de ce type de résolution. On obtient alors un gain pour toutes les instances par rapport aux résultats heuristiques de la méthode sans stabilisation (voir tableau 3.5). En moyenne, les temps de résolution sont divisés par deux.

Au vu des résultats obtenus en stabilisant les variables duales lors de la génération de colonnes, il est judicieux d'intégrer cette méthode de stabilisation lors de la résolution des différents jeux de données. De plus, lorsqu'on ajoute cette méthode au sein d'une résolution heuristique, les gains sur les temps de résolution sont encore plus importants. Par conséquent, nous retiendrons au chapitre suivant l'approche heuristique stabilisée pour résoudre les relaxations linéaires rencontrées dans une méthode de séparation et évaluation progressive qui nous permettra d'obtenir des solutions entières.

Tableau 4.2 – Résultats heuristiques de l’approche avec stabilisation

Id Pb	Z_{LP}	Nb Iter	Nb Col /Iter	Tps PM min	Tps SP min	Tps total min	Gain CPU (%)
P199_21	20	144	123	3	4	7	61
P204_22	21	118	216	3	4	7	65
P206_22	21	150	168	3	4	7	53
P262_28	27	158	265	7	11	18	57
P286_31	29	209	311	21	18	39	25
P315_35	33	193	269	16	14	30	43
P344_37	35	196	280	19	47	66	44
P463_50	48	298	354	88	102	190	45

4.3.2 Résolution en nombres entiers

Comme la plupart des solutions optimales des relaxations linéaires ne sont pas entières, il est nécessaire d’inclure l’approche heuristique stabilisée au sein d’une méthode de séparation et évaluation progressive. Dans le chapitre 2, on a parlé des contraintes d’inter-tâches dans une méthode exacte. Or afin de résoudre les plus grosses instances, nous avons utilisé une résolution heuristique en nombres entiers. De ce fait, l’exploration de l’arbre de branchement ne se fait qu’en profondeur i.e., sans retour en arrière. Ensuite, il faut savoir qu’à chaque noeud de branchement, la résolution est heuristique afin d’oter du temps en évitant le phénomène de queue engendré par les méthodes de génération de colonnes. Cette queue est coupée lorsque la valeur de l’objectif n’a pas varié de plus de 10^{-3} au cours de 3 dernières itérations. Puis au niveau des décisions de branchement sur les inter-tâches, on ne peut fixer celles-ci à 1, que si les valeurs des variables X_{ij} sont initialement supérieur à 0.7. Aussi il est possible de fixer 10 inter-tâches ou moins à un même noeud de branchement.

Une autre méthode de branchement est utilisé au cours de la résolution en nombres entiers. À un noeud de branchement, il s’agit de fixer à 1 une variable θ_p correspondant

à une colonne i.e., un chemin réalisable . En effet, la variable θ_p dont la valeur est la plus près de 1, est fixée à 1. Pour imposer $\theta_p = 1$, on retire du problème maître la colonne associée à θ_p , on modifie les membres de droite concernés et on modifie les sous-réseaux des sous-problèmes en éliminant tout les arcs associés à une tâche couverte par la colonne associé à θ_p . Il est à noter que l'on favorise la fixation des colonnes au détriment de la fixation des inter-tâches.

Le tableau 4.3 présente les résultats sur les 8 plus grands jeux de données. On peut constater dans un premier temps que, pour toutes les instances du PSTAJT, des solutions entières ont été trouvées. On peut remarquer que les valeurs des solutions optimales trouvées lors de la relaxation linéaire (voir tableau 4.1) sont les mêmes que pour celles de la résolution en nombres entiers, ce qui signifie que pour chaque résolution en nombres entiers, la solution trouvée ne peut être améliorée (gap nul pour chaque instance). Il est à noter aussi que pour les plus grands jeux de données, les solutions entières trouvées comportent un gain sur le nombre de chauffeurs utilisés par rapport aux solutions trouvées par la société de transport en commun de la ville méditerranéenne (caractère gras dans le tableau 4.3). Enfin, les temps totaux de résolution des différentes instances indiquent que l'un des objectifs fixés par ce mémoire est atteint, à savoir résoudre les plus grandes instances dans des temps raisonnables. En effet, la plus grande instance (463 voyages) est résolue en un temps d'environ huit heures, ce qui est un temps acceptable pour ce type de problème.

Il faut noter que les solutions fournies par la société de transport de la ville méditerranéenne violent certaines contraintes de temps sur les journées de travail : temps minimum et maximum d'une pièce de travail (min pc et max pc) et temps minimum et maximum de travail (min wk et max wk). Par conséquent, afin d'établir une comparaison équitable entre ces solutions et celles produites par l'approche que nous avons développée, nous avons jusqu'à maintenant utilisé des contraintes relaxées telles que les solutions fournies par la société de transport soient tout juste réalisables. Pour

Tableau 4.3 – Résultats heuristiques en nombres entiers

Id Pb	Z_{IP}	Nb noeuds branch.	CPU PM min	CPU SP min	CPU total min
P199_21	20	17	6	5	12
P204_22	21	24	6	6	13
P206_22	21	22	6	7	13
P262_28	27	19	18	16	34
P286_31	29	48	55	30	86
P315_35	33	47	48	29	78
P344_37	35	38	62	56	120
P463_50	48	75	297	174	474

compléter notre étude, nous avons testé notre approche sur trois des plus grands jeux de données (P315_35, P344_37 et P463_50) en considérant des contraintes resserrées i.e., se rapprochant plus des contraintes réelles (celles de la convention collective).

Les résultats sont présentés dans le tableau 4.4 qui comporte deux parties. La première partie donne les résultats obtenus précédemment avec les contraintes relaxées et la deuxième partie ceux obtenus avec les contraintes resserrées. Pour chacune de ces parties, on indique sous l'intitulé "Violation permise", la violation maximale permise (en minutes) pour chacune des contraintes relaxées, les valeurs des paramètres étant données au tableau 3.1. Par exemple, la violation maximale permise de 40 minutes pour le temps minimum de travail signifie que ce temps minimum devrait être en réalité de 6h50 pour des journées de deux pièces plutôt que de 6h10 comme nous l'avons considéré jusqu'à présent. On constate ainsi que les valeurs des violations permises prises par la société de transport sont assez grandes. Sous l'intitulé "Résolution", on donne la valeur de la solution entière (Z_{IP}), le nombre de noeuds de branchement (Nb B&B) et le temps de résolution requis (CPU) en minutes. Finalement, sous l'intitulé "Nb de jour. trav.", on rapporte les nombres de journées de travail par type de journée (AM, PM, 2PCE) découlant de la solution obtenue.

Tableau 4.4 – Résultats heuristiques en nombres entiers avec différentes valeurs des violations

Id Pb	Violation permise (min)				Résolution			Nb de jour. trav.		
	min pc	max pc	min wk	max wk	Z_{IP}	Nb B&B	CPU (min)	AM	PM	2PCE
P315_35	15	15	40	90	33	47	78	5	12	16
P344_37	15	15	40	90	35	38	120	3	10	22
P463_50	15	15	40	90	48	75	474	4	15	29
P315_35	15	15	0	30	33	77	59	5	11	17
P344_37	15	15	0	30	35	71	153	3	11	21
P463_50	15	15	0	30	48	96	389	10	19	19

De ces résultats, on tire les observations suivantes. Premièrement, de part et d'autre, les solutions entières requièrent le même nombre de chauffeurs (Z_{IP}) malgré la possibilité de moins violer certaines contraintes pour la seconde résolution. On constate aussi que ces diminutions de possibilités de violations, n'affectent pas les temps de résolution qui sont globalement semblables aux temps de résolution avec les violations de base. Deuxièmement, les valeurs des violations permises n'ont pas vraiment d'impact sur la répartition des journées entre les différents types. En effet, de manière générale, les journées ayant deux pièces de travail sont beaucoup plus nombreuses dans les solutions obtenues que les journées ne comportant qu'une seule pièce. Cela s'explique facilement par le fait qu'il est possible de travailler plus longtemps dans une journée de travail avec deux pièces. Ainsi, ces types de journée sont privilégiés dans des solutions requérant un minimum de chauffeurs.

Finalement, notons que des tests ont été effectués sans permettre aucune violation. Dans ce cas, l'approche heuristique proposée n'a pas réussi à obtenir une solution entière réalisable pour certains problèmes. Étant donné la taille des problèmes, il n'a pas été possible de déterminer s'ils étaient non-réalisables ou si le branchement était

trop heuristique pour générer une solution réalisable lorsque le domaine réalisable devient très étroit.

Conclusion

Ce projet de maîtrise portait sur le problème de construction simultanée d'horaires de chauffeurs et d'itinéraires d'autobus. L'objectif de ce projet était de pouvoir résoudre les plus grandes instances de ce type de problème en des temps raisonnables tout en ayant des solutions de bonne qualité. Un modèle et une méthode semblables à ceux proposés par Haase *et al.* (2001) ont été testés dans un premier temps. Il s'est avéré que le temps passé dans les sous-problèmes de la génération de colonnes occasionnait des temps de résolution beaucoup trop grands.

De ce fait, une méthode d'accélération de résolution des sous-problèmes a été proposée. Celle-ci a été insérée au sein même de la génération de colonnes afin de simplifier les sous-réseaux et de découper la résolution des sous-problèmes. Cette méthode d'accélération a été très bénéfique car elle engendre un gain moyen de 85 % au niveau du temps total de résolution des différentes instances. Le plus grand jeu de données (463 voyages) pouvait alors être résolu dans un temps qui commençait à être raisonnable.

Observant des temps de résolutions encore trop élevés au niveau du problème maître, une méthode de stabilisation des variables duales a été ajoutée à la méthode d'accélération. Les résultats obtenus ont démontré l'apport de cette méthode, puisque les gains moyen au niveau du temps de résolution ont été de 40%.

Les deux méthodes présentées ont ensuite été insérées dans une approche heuristique de séparation et évaluation progressive, et les résultats qui s'ensuivirent furent très satisfaisants. En effet, d'une part, les temps de résolution sont fort acceptables pour les plus grandes instances et, d'autre part, les solutions entières sont de bonne qualité. Il est à noter finalement que les modèles mis en place ont permis de réduire le nombre de chauffeurs requis de l'ordre de 5%.

Pour terminer ce mémoire, nous proposons les directions futures de recherche suivantes. Premièrement, d'autres considérations du PSTAJT pourraient être prises en compte. À titre d'exemple, dans certaines conventions collectives, on exige que les journées de travail aient, en moyenne, une durée visée. Cette considération nécessite des modifications au niveau du modèle et de la méthode de résolution proposées. Deuxièmement, lors de la construction simultanée, des itinéraires d'autobus et des horaires de chauffeurs, on pourrait permettre la possibilité de modifier légèrement l'horaire (construit manuellement par la société de transport en commun de la ville méditerranéenne) des voyages afin de réaliser des économies sur les coûts d'autobus et de chauffeurs. Finalement, on pourrait même penser à considérer ce problème dans le contexte où il faut plutôt définir cet horaire en respectant au mieux les fréquences désirées de passage sur les routes.

Bibliographie

BANIHASHEMI M., L., et HAGHANI, A. (2000). A New Model for the Mass Transit Crew Scheduling Problem. *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*, 505, Springer, Berlin, Germany, 1–16.

BEN AMOR H., (2002). Stabilisation de l'algorithme de génération de colonnes. *PhD Dissertation 2002*. Département de mathématiques et de génie Industriel, École Polytechnique de Montréal, Canada.

BEN AMOR H., et DESROSIERS, J., (2003). A Proximal Trust-Region-Algorithm for Column Generation Stabilization. *Cahiers du GERAD* G-2003-43.

BIANCO, L., MINGOZZI, A., et RICCIARDELLI, S. (1994). A Set Partitioning Approach to the Multiple Depot Vehicle Scheduling Problem. *Optimization Methods and Software*, 3, 163–194.

BORNDÖRFER, R., GRÖTSCHER, M., et LÖBEL, A. (2001). Scheduling Duties by Adaptive Column Generation. *ZIB-Report*, 01-02, Berlin, Allemagne.

CEDER A. (1986). Methods for Creating Bus Timetables. *Transportation Research A*, 21(1), 59–83.

CEDER A. (1984). Bus Frequency Determination Using Passenger Count Data. *Transportation Research A*, 18(5/6), 439–453.

DANTZIG, G.B. et WOLFE, P. (1960). Decomposition Principle for Linear Programs. *Operational Research*, 8, 101–111.

DESAULNIERS, G., LAVIGNE, J., et SOUMIS, F. (1998). Multi-Depot Vehicle Scheduling Problems with Time Windows and Waiting Cost. *European Journal of Operational Research*, 111, 479–494.

DE SILVA, A. (2001). Combining Constraint Programming and Linear Programming on an Example of Bus Driver Scheduling. *Annals of Operations Research*, 108, 277–291.

DESROCHERS, M., et SOUMIS, F. (1989). A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, 23, 1–13.

FRELING, R., HUISMAN, D., et WAGELMANS, A.P.M. (2003). Models and Algorithms for Integration of Vehicle and Crew Scheduling. *Journal of Scheduling* 6, 63–85.

FRELING, R., HUISMAN, D., et WAGELMANS, A.P.M. (2001). Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice. *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems* 505, Springer, Berlin, Allemagne, 73–90.

FRELING, R., WAGELMANS, A.P.M., et PAIXÃO, J.M.P. (1999). An Overview of Models and Techniques for Integrating Vehicle and Crew Scheduling. *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems* 471, Springer-Verlag, Heidelberg, 441–460.

FRELING, R. (1997). Models and Techniques for Integrating Vehicle and Crew Scheduling. *Tinbergen Institute Research Service*, 157, Thesis Publishers, Amsterdam.

GALLO, G. et DI MIELE, F. (2001). Dispatching Buses in Parking Depots. *Transportation Science*, 35(3), 322–330.

- HAASE, K., DESAULNIERS, G., et DESROSIERS, J., (2001). Simultaneous Vehicle and Crew Scheduling in Urban Mass Transit Systems. *Transportation Science*, 35(3), 286–303.
- HAASE, K., et FRIBERG, C., (1999). An Exact Branch and Cut Algorithm for the Vehicle and Crew Scheduling problem. *Computer-Aided Scheduling of Public Transport, Lecture Notes in Economics and Mathematical Systems*, 471, Springer, Berlin, Allemagne, 63–80.
- HADJAR, A., MARCOTTE, O. et SOUMIS, F. (2001). A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. *Les Cahiers du Gerad*, G-2001-25.
- HICKMAN, M.D. (2001). An Analytic Stochastic Model for the Transit Vehicle Holding Problem. *Transportation Science*, 35(3), 215–237.
- HUISMAN, D., FRELING, R., et WAGELMANS, A.P.M. (2003). Multiple-Depot Integrated Vehicle and Crew Scheduling. *Technical Report EI2003-02, Econometric Institute, Erasmus University, Rotterdam, Pays-Bas*.
- LEMARÉCHAL, C. (1974). An Algorithm for Minimizing Convex Functions. *Proceedings IFIP 74 Congress, Rosenfeld J.L. (ed.)*, North-Holland, Amsterdam, 552–556.
- LÖBEL, A. (2000). Solving Large-Scale Multiple-Depot Vehicle Scheduling Problems. In : N.H.M Wilson (ed.), *Computer-aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, 471, Springer-Verlag, Heidelberg, 193–220.
- LÖBEL, A. (1998). Vehicle Scheduling in Public Transit and Lagrangean Pricing. *Management Science*, 44(12), 1637–1649.

MESQUITA, M., et PAIXAO, J. (1999). Exact Algorithms for the Multiple-Depot Vehicle Scheduling Problem Based on Multicommodity Network Flow Type Formulations. In : N.H.M Wilson (ed.), *Computer-aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems, 471, Springer-Verlag, Heidelberg, 221–243.

MINGOZZI, A., BOSCHETTI, M.A., RICCIARDELLI, S., et BIANCO, L. (1999). A Set Partitioning Approach to the Crew Scheduling Problem. *Operations Research*, 47(6) 873–888.

MINGOZZI, A., BIANCO, L., et RICCIARDELLI, S. (1995). An Exact Algorithm for Combining Vehicle Trips. *Computer-Aided Transit Scheduling*, Springer-Verlag, New York.

ODONI, A.R., WILSON, N., et ROUSSEAU, J.M. (1994). Models in Urban and Air Transportation. Handbooks in Operations Research and Management Science 6, *Operations Research and the Public Sector*, S.M. Pollock, M.H. Rothkopf et A. Barnett (éditeurs), Elsevier Science, Amsterdam, 129–150.

NGUYEN, S., PALLOTINO, S., et MALUCELLI, F. (2001). A Modeling Framework for Passenger Assignment on a Transport Network with Timetables. *Transportation Science*, 35(3), 238–249.

RIBEIRO, C., et SOUMIS, F. (1994). A Column Generation Approach to the Multiple-Depot Vehicle Scheduling Problem. *Operations Research*, 42(1), 41–52.

ROCKAFELLAR, R.T. (1976). Monotone operators and the proximal point algorithm. *SIAM J. Control. Optim.*, 14, 877–898.

RYAN, D. M., et FOSTER, B. A. (1981). An Integer Programming Approach to Scheduling. *Computer Scheduling of Public Transport : Urban Passenger Vehicle and Crew Scheduling*, A. Wren (ed.), Pays-Bas, Amsterdam, 269–280.