



Titre: Affectation des wagons aux trains de passagers dans un contexte
Title: opérationnel

Auteur: Norbert Lingaya
Author:

Date: 2003

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Lingaya, N. (2003). Affectation des wagons aux trains de passagers dans un
Citation: contexte opérationnel [Mémoire de maîtrise, École Polytechnique de Montréal].
PolyPublie. <https://publications.polymtl.ca/6983/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/6983/>
PolyPublie URL:

**Directeurs de
recherche:** Guy Desautniers, & François Soumis
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

AFFECTATION DES WAGONS AUX TRAINS DE PASSAGERS DANS UN
CONTEXTE OPÉRATIONNEL

NORBERT LINGAYA
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)

JANVIER 2003

© Norbert Lingaya, 2003.



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

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

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

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

0-612-81551-X

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

AFFECTATION DES WAGONS AUX TRAINS DE PASSAGERS DANS UN
CONTEXTE OPÉRATIONNEL

présenté par : LINGAYA Norbert

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

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

M. SAVARD Gilles, Ph.D., président

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

M. SOUMIS François, Ph.D., membre et codirecteur de recherche

M. GAMACHE Michel, Ph.D., membre

À Clive Herbert Lingaya, 1940 - 2000

Remerciements

Je remercie mesdames Francine Hébert, Natali Gagnon et monsieur Alain Vigeant de VIA Rail pour leur étroite collaboration, les gouvernements Canadien et Québécois pour leur soutien financier à travers les fonds FCAR et CNRSG, mes directeurs de mémoire pour leur indulgence et infinie patience ainsi que mes collègues du GERAD pour leur soutien technique. Je ne saurais oublier tous ceux qui m'ont soutenu moralement, sans leurs encouragements ce mémoire n'aurait pas vu le jour.

Résumé

Ce mémoire traite de l'affectation de wagons aux trains de passagers dans un contexte opérationnel. Ce problème apparaît typiquement lorsqu'un transporteur ferroviaire choisit d'optimiser l'utilisation de son matériel roulant en adaptant chaque convoi de wagons en fonction du taux de fréquentation du train desservi. Dans ce cas, les transporteurs opèrent généralement en trois phases : la première consiste à identifier une semaine typique afin de planifier des cycles de locomotives et wagons, obtenant ainsi un plan non daté qui peut être déroulé sur une saison. Fort de ce plan, une deuxième étape consiste à corriger localement et de façon datée ce plan lorsque l'approximation par une semaine type s'éloigne trop des prévisions, comme c'est le cas lors de journées de congé atypiques (Noël, Action de Grâce) ou à la frontière de deux saisons. Enfin, la dernière phase, utilisée dans un contexte opérationnel, vise à réaffecter les wagons disponibles en fonction de données plus précises sur la demande de chaque train. Cette troisième phase, dont l'horizon d'optimisation est plus court, prend en compte des contraintes opérationnelles qui sont généralement relaxées dans les phases de planification à long et moyen terme.

Nous avons implanté chez VIA Rail un logiciel traitant les trois phases susmentionnées et ce mémoire se concentre sur la troisième dont la principale difficulté est de tenir compte des positions des wagons dans les convois. Nous proposons deux modèles de base du type multiflots dans un réseau avec contraintes additionnelles. Le premier modèle est intuitif mais fait appel à des variables inhabituelles correspondant à des fonctions gérant le positionnement des wagons dans les convois. Le second modèle occulte ces variables grâce à une transformation équivalente appliquée au premier modèle qui fait toutefois apparaître des contraintes globales non linéaires. Une

fait toutefois apparaître des contraintes globales non linéaires. Une approche de résolution heuristique est développée pour résoudre ce second modèle. Celle-ci repose sur une méthode de génération de colonnes imbriquée dans une procédure de séparation et évaluation progressive. Les contraintes globales non linéaires sont initialement relaxées et traitées par la procédure de branchement lorsqu'elles sont violées. Des extensions du modèle de base essentielles pour l'application chez VIA Rail sont ensuite discutées avant de présenter des résultats numériques qui démontrent le potentiel des travaux réalisés. Finalement, une synthèse du projet est présentée.

Abstract

This master's thesis addresses the operational passenger car assignment problem. Such a problem typically arises when a railway carrier opts to maximize the use of its equipment by tailoring the set of cars assigned to a train according to its demand. Traditionally these carriers plan their operations in three phases: first they identify a typical week of operation in order to plan cycles of cars and locomotives, yielding an undated cyclic planning that can be unrolled over a season. With this plan, a second phase consists of, locally and in a dated manner, correcting it where the typical week approximation is too far from the forecasted demand, like it is the case on special holidays (Christmas, Thanksgiving) or at the transition between two seasons. Finally, the last phase addresses the car reallocation in an operational context according to more accurate data on car demand. This third phase, which planning horizon is shorter, takes into account some operational constraints that are relaxed during the long and medium term planning phases.

We have implemented at VIA Rail a software addressing the three above mentioned phases and this master's thesis focuses on the third, which challenge was to consider the car position within their respective train. We present two basic multi-commodity network flow models with additional constraints. The first one is intuitive but considers complex variables that correspond to function dealing with the positions of the cars within their train. The second model removes those variables using an equivalent transformation applied to the former model, however rising non-linear global constraints. A heuristic solution approach is developed to solve the latter: it is based on a column generation method embedded in a branch-and-bound algorithm.

The non-linear global constraints are at first relaxed and gradually enforced as they are violated. We then discuss extensions to the basic model that are required by VIA Rail's specific context and we present numerical results that demonstrate the potential of our approach. Finally we conclude with a synthesis of the project.

Table des matières

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xiv
LISTE DES FIGURES	xv
INTRODUCTION	1
CHAPITRE 1 : REVUE DE LA LITTÉRATURE	6
1.1 : Transport de marchandises	7
1.2 : Transport de passagers	8

1.3 : Génération de colonnes	10
--	----

1.4 : Positionnement du mémoire	11
---	----

CHAPITRE 2 : OPERATIONAL CAR ASSIGNMENT AT VIA RAIL

CANADA	12
-------------------------	-----------

2.1 : Sommaire	12
--------------------------	----

2.2 : Introduction	15
------------------------------	----

2.2.1 : Review of the literature	17
--	----

2.2.2 : Contribution	19
--------------------------------	----

2.2.3 : Paper Overview	19
----------------------------------	----

2.3 : Basic Problem Description	20
---	----

2.4 : Mathematical Formulations	23
---	----

2.4.1 : Time-Space Network	23
--------------------------------------	----

2.4.2 : Intuitive Formulation	27
---	----

2.4.3 : Tractable Formulation	31
---	----

2.5 : Solution Methodology	34
--------------------------------------	----

2.5.1 : Computing Car Cycles	35
--	----

2.5.2 : Computing Car Positions	37
---	----

2.6 : Extensions	39
2.6.1 : Multiple Service Classes	39
2.6.2 : Car Availability	40
2.6.3 : Maximum Mileage between Maintenance Checks	41
2.6.4 : Short Wye	42
2.6.5 : Car Orientation	44
2.6.6 : Trains with Multiple Locomotives	45
2.6.7 : Other Extensions	47
2.7 : Computational Experiments	48
2.7.1 : Description of Data	48
2.7.2 : Computational Results	50
2.8 : Conclusion	54
2.9 : Bibliography	55
2.10 :Appendix	57
CHAPITRE 3 : EXTENSIONS ET SYNTHÈSE	60
3.1 : Extensions	60
3.1.1 : Substitution de classe	60

3.1.2 : Entretien de jour	61
3.1.3 : Capacité d'entreposage des wagons dans les stations intermé- diaires	62
3.1.4 : Revenus anticipés différents	63
3.1.5 : Coût variable et restrictions sur la modification de convoi . .	63
3.1.6 : Contiguïté des wagons de même classe	64
3.2 : Synthèse	64
CONCLUSION	67
BIBLIOGRAPHIE	69

Liste des tableaux

Tableau 2.1 : Number of trains and average number of requests in test instances	51
Tableau 2.2 : Average computational results	53

Liste des figures

Figure 2.1 : Part of a network created by a two-atom cycle	25
Figure 2.2 : Run-thru and turn-around connections	43

Introduction

L'acquisition et l'amortissement de locomotives et de wagons représentent une grande partie des coûts d'exploitation des transporteurs ferroviaires. Une utilisation bien planifiée de ce matériel est donc une nécessité pour le transporteur qui veut rester compétitif et faire face à la concurrence émanant d'autres transporteurs, qu'ils soient ferroviaires, terrestres ou aériens. De là naît l'intérêt de l'étude de cette utilisation, c'est-à-dire de l'affectation des locomotives et des wagons aux services de train ; nous entendons par *service de train* un service offert par un transporteur ferroviaire, c'est-à-dire transporter des passagers d'un point A à point B à une heure donnée. Il s'agit d'un problème complexe qu'il convient de modéliser avec soin si l'on veut résoudre en des temps acceptables des cas rencontrés par des compagnies telles que VIA Rail.

Dans le monde des transporteurs ferroviaires, on trouve deux philosophies quant à la politique d'exploitation du matériel roulant : la première partitionne l'équipement en sous-ensembles atomiques, que l'on nomme convois, pour ensuite les affecter aux services de train, alors que la seconde s'autorise à modifier la constitution des convois en plusieurs endroits de son réseau en fonction de la demande des services de train. Cette composition de convoi, que l'on pourrait qualifier de dynamique, augmente les risques de bris et de retard du fait du nombre plus élevé de manipulations engendrées. Cependant, elle offre l'avantage de réduire le nombre de wagons tirés à vide et, avec une affectation adéquate, permet d'opérer avec moins de matériel, d'où une économie au niveau de l'acquisition et de l'amortissement. Nous ne discuterons pas ici des avantages ou inconvénients d'une politique par rapport à l'autre mais nous nous pencherons sur le cas des transporteurs ayant opté pour la deuxième politique. La

motivation de ce choix réside dans le fait que cette philosophie mène à un problème plus complexe (ne serait-ce que par le nombre d'entités à gérer) qui, de plus, doit être résolu avec attention si l'on veut en récolter les bénéfices promis, à savoir réduire le nombre de wagons tirés à vide et opérer avec un inventaire réduit. Nous invitons par contre les lecteurs intéressés par la première philosophie à consulter les travaux de DUBOIS (2001) : ce dernier s'est penché sur le problème de couverture des services de train par des rames indivisibles auto-tractées semblables aux rames de métro.

L'affectation de locomotives et wagons aux trains ne couvre qu'une partie du spectre de la planification en transport ferroviaire. Cependant, modéliser d'un bloc l'ensemble de ce processus donnerait lieu à un problème trop complexe pour être résolu. BUS-SIECK et al. (1997) proposent une division en 6 étapes de ce processus que l'on résout séquentiellement. Dans un premier temps, le transporteur doit se doter d'un réseau physique. La disposition géographique des agglomérations devant être desservies et des obstacles naturels sont les principaux facteurs qui en guident la constitution. Sur ce réseau, le transporteur peut alors planifier des lignes de services, c'est-à-dire un ensemble de points du réseau, nommés stations, reliés entre eux à une certaine fréquence par des services de train. Par la suite, une troisième étape est de définir les horaires de ces services de train pour lesquels il est de coutume de choisir une certaine régularité, caractéristique qu'on retrouve dans presque toute l'industrie du transport de passagers, quelqu'en soit le mode. Cette étape, complexe, fait souvent l'objet d'un modèle approximatif où certaines contraintes locales sont relaxées, générant ainsi un horaire de base qui doit être manuellement adapté par endroits, ce qui constitue la quatrième étape. À ce stade, nous avons les horaires des services des train auxquels il faut affecter locomotives et wagons, soit la cinquième étape sur laquelle nous nous attardons dans ce mémoire. Enfin, pour opérer ce matériel roulant, il faut du personnel. L'ultime étape de cette décomposition consiste donc à générer les horaires du personnel affecté aux trains en s'assurant que lois et conventions collectives sont respectées.

Nous citons plus haut VIA Rail (que nous nommerons par la suite VIA). Il s'agit d'un transporteur ayant opté pour la deuxième philosophie avec qui nous avons collaboré. Nous présentons dans ce memoire un des volets des travaux que nous avons effectués avec ce transporteur, dans le cadre de l'affectation de locomotives et de wagons aux trains. Il est à noter cependant que le fruit de ces travaux n'est pas limité au contexte de VIA. En effet, le modèle et les techniques de résolution développés peuvent être appliqués à tout transporteur ferroviaire ayant choisi la deuxième philosophie à l'instar de VIA. Bien entendu, des modifications mineures doivent être apportées à ce modèle et ces techniques afin de tenir en compte des particularités de chaque transporteur.

L'affectation des locomotives et wagons aux trains, bien que n'étant qu'une étape du processus global, n'en reste pas moins un problème ardu. Dans le cas de VIA, notre collaboration a fait apparaître une décomposition du problème en trois sous-étapes.

La première fût de modéliser et résoudre le problème de planification à long terme obtenu à partir d'un horaire d'une semaine type. Cette phase, baptisée Module de génération de Cycles d'Équipement (Equipment Cycling Module ou ECM) est brièvement décrite dans l'article que nous présentons au chapitre 2 et est étudiée plus en détails dans un article publié par CORDEAU et al. (2001a).

Par la suite, il arrive que certains jours donnés, les services de train ne suivent pas exactement l'horaire de la semaine type mentionnée plus tôt, comme c'est souvent le cas lors des longs congés où la demande est plus forte. Le problème d'adaptation des affectations cycliques générées lors de la première phase a donné naissance au deuxième module nommé Module de Planification Datée (Dated Planning Module, ou DPM).

Ces deux modules permettent de générer une planification à long terme, typiquement six mois, basée sur une estimation de vente de billets. Lors des opérations, la demande

effective peut différer de celle originalement prévue par un modèle stochastique. Il peut alors s'avérer judicieux de réorienter le matériel en réduisant les convois sur les services de train où la demande a été surestimée pour ajouter des voitures supplémentaires sur les trains où la demande dépasse la capacité courante. Ce problème a donné lieu au Module des Opérations Journalières (Daily Operations Module, ou DOM).

De la même façon que la décomposition proposée par BUSSIECK et al. (1997) relaxe certaines contraintes à l'étape 3 pour ne les introduire qu'à l'étape suivante, les modules de planification à long terme, ECM et DPM, négligent certaines contraintes opérationnelles que nous réintroduisons dans le module DOM. Il s'agit principalement du positionnement et de l'orientation des wagons dans leur convoi. Ces caractéristiques ont une grande importance sur la faisabilité d'une solution dans la mesure où le décrochage et l'accrochage d'un wagon en milieu de convoi est plus long que la même opération en queue de convoi et peut donc invalider un enchaînement d'affectations. En ce qui concerne l'orientation, des contraintes opérationnelles d'ordre légal veulent que toutes les voitures d'un même convoi accueillant des passagers partagent la même orientation pour des raisons de sécurité. La prise en compte de ces nouvelles contraintes de front conduit à un problème avec un très grand nombre de variables qui est difficile à résoudre. Nous proposons dans ce mémoire une approche heuristique à deux niveaux : un niveau microscopique constitué d'un modèle où ces contraintes compliquantes sont relaxées, résolu par une approche de génération de colonnes couplée à un algorithme de séparation et évaluation progressive ; un niveau macroscopique constitué d'un algorithme d'énumération dynamique qui, pour toute solution du premier niveau, est capable de générer des valeurs réalisables pour le positionnement s'il en existe ou, le cas échéant, d'introduire des contraintes au niveau inférieur afin de réduire ces infaisabilités.

Il s'agit là de la majeure contribution de ce mémoire : modéliser une application réelle complexe en faisant appel à des contraintes non linéaires qui ont permis d'éviter l'uti-

lisation d'un très grand nombre de variables binaires de positionnement de wagons et proposer une méthode de résolution heuristique performante qui traite efficacement ces contraintes non linéaires à l'aide d'un branchement heuristique.

La structure de ce mémoire est comme suit. Le premier chapitre présente une revue de la littérature qui permet de faire le point sur l'état actuel des connaissances dans les domaines abordés par le présent mémoire. Par la suite, dans le chapitre 2, nous présentons un article publié dans la revue *Transportation Research, Part B*, qui décrit plus en profondeur le module DOM. Enfin, une brève synthèse du projet effectué en collaboration avec VIA sera présentée dans le troisième chapitre avant de passer à la conclusion.

Chapitre 1

Revue de la littérature

La littérature est relativement pauvre en articles traitant de l'affectation de matériel roulant aux trains de passagers et l'intérêt pour ce domaine de recherche n'est que récent. CORDEAU et al. (1998) se sont penchés sur la question et ont publié en 1998 une revue exhaustive des travaux traitant de l'optimisation en transport ferroviaire. Nous ne prétendons pas ici établir une revue aussi poussée, couvrant un spectre aussi large, mais nous voudrions plutôt nous en inspirer pour brièvement présenter les travaux connexes aux nôtres afin de mettre en relief et situer notre contribution. On constate que les problèmes de transport ferroviaire ont empiriquement été séparés en deux classes : d'une part les problèmes ayant trait au transport de marchandises, qui furent les premiers à être considérés, et le transport de passagers d'autre part. Nous respecterons ici cette séparation en nous attardant plus particulièrement sur la deuxième classe, à laquelle appartiennent nos travaux. Enfin, comme nous aurons l'occasion de le voir par la suite, notre approche est basée sur une modélisation de type tournées de véhicules avec contraintes de capacité, résolu par une technique de génération de colonnes imbriquée dans un algorithme de séparation et évaluation progressive. Il nous semble donc opportun de présenter un volet sur cette méthode de résolution.

1.1 Transport de marchandises

La première manifestation de travaux sur le transport de marchandises remonte à 1976 où FLORIAN et al. (1976) s'intéressent au problème d'affectation de locomotives à des trains de marchandise pour lesquels la demande s'exprime en puissance motrice. Leur objectif, semblable au nôtre, est de réduire les coûts d'exploitation et d'acquisition du matériel roulant. Pour ce faire, ils proposent une méthode de décomposition de Benders du problème modélisé par un programme linéaire avec variables mixtes, ce qui leur permet de résoudre des problèmes constitués d'environ 200 trains couverts par 2 types de locomotives.

Par la suite, on trouve des travaux où la demande s'exprime en terme de nombre de locomotives par train. BOOLER (1980) et WRIGHT (1989) proposent des approches heuristiques basées sur des problèmes d'affectation qui ont permis de résoudre des problèmes de taille moyenne. FORBES et al. (1991) proposent quant à eux une méthode de résolution exacte basée sur une suite de relaxations d'un problème d'affectation, encadrées par un algorithme de séparation et évaluation progressive. Cette méthode leur permet de résoudre des problèmes de taille semblable en termes du nombre de trains mais avec 5 types de locomotive, le tout cependant au prix d'un temps d'exécution plus conséquent.

Six ans plus tard, ZIARATI et al. (1997) introduisent des contraintes d'entretien des locomotives jusqu'ici relaxées et proposent pour le même type de problème une nouvelle heuristique où ils découpent l'horizon temporel en tranches de temps se chevauchant. La suite de problèmes définis sur ces tranches de temps est séquentiellement résolue selon l'ordre chronologique par une méthode de décomposition de Dantzig-Wolfe couplée à un algorithme de séparation et évaluation progressive. Le chevauchement temporel de quelques jours entre deux tranches contigües garantit une certaine continuité dans la solution ainsi générée par morceaux. Cette approche leur a permis de

résoudre, pour des données fournies par le transporteur CN North America, des problèmes de très grande taille : un peu moins de 2000 trains et plus de 25 types de locomotive. En 1999, ZIARATI et al. (1999) perfectionnent leur modélisation en introduisant des plans de coupe qui leur permettent d'améliorer la qualité des solutions générées.

1.2 Transport de passagers

Ce n'est que récemment que l'industrie du transport de passagers par voie de chemin de fer a suscité l'intérêt dans notre communauté. Ainsi, en 1992, RAMANI et MANDAL (1992) développent pour le compte de la compagnie Indian Railways une application gérant l'affectation des locomotives et wagons aux trains de passagers. Les contraintes dont ils doivent tenir compte sont semblables à celles que nous avons rencontrées dans le cadre de notre collaboration avec VIA, à savoir différentes classes de voiture, temps minimal de connexion en fonction de la modification de convoi, entretien périodique du matériel roulant. Cependant, leur approche diffère de la nôtre dans la mesure où elle repose sur une suite d'améliorations locales qui connectent les services de trains en se basant sur les heures d'arrivée et de départ aux stations.

En 1997, BUSSIECK et al. (1997), dont nous avons déjà cité la division en six étapes du processus de planification en transport ferroviaire, s'arrêtent, dans le cadre d'une synthèse sur la matière, sur le cas particulier où une seule pièce d'équipement doit être affectée à chaque service de train. Cette simplification leur autorise une modélisation en un problème de tournées de véhicule avec dépôts multiples (MDVSP). Si cette simplification trouve sa justification chez les transporteurs ayant opté pour la philosophie des convois indivisibles, elle perd malheureusement son intérêt dans les cas qui nous intéressent où l'on attend un important gain en terme de nombre de

pièces d'équipement en faisant justement varier le nombre de pièces affecté à chaque train en fonction de sa demande propre.

En 2001, CORDEAU et al. (2001a) détaillent la modélisation du module ECM dont nous avons déjà fait mention. Ils présentent un modèle en deux phases. La première visant à déterminer le routage des locomotives est un modèle multiflots avec énumération des enchaînements de trains dont le temps de connexion interdit toute modification de convoi. Il est résolu par génération de colonnes couplée à un algorithme heuristique de séparation et évaluation progressive où les contraintes d'intégrité sur les variables de flots des wagons sont relaxées. La deuxième phase, également un modèle multiflots résolu par génération de colonnes, permet de déterminer le routage des wagons connaissant celui des locomotives. Parallèlement, CORDEAU et al. (2000, 2001b) présentent un modèle produisant des solutions optimales pour une variante légèrement simplifiée du problème précédent. Ils se basent sur une méthode de décomposition de Benders où le problème maître détermine les enchaînements de trains sans modification de convoi et les sous-problèmes trouvent les routes des locomotives et des wagons. Dans ces deux cas, CORDEAU et al. (2000) résolvent des problèmes comportant 350 trains.

Enfin, en 2001, DUBOIS (2001) présente ses travaux où il s'intéresse à la planification stratégique en transport ferroviaire de passagers avec horaires fixes et variables. Ces travaux furent également entrepris avec la collaboration de VIA, mais il étudie le scénario où VIA choisirait d'opérer avec des convois indivisibles. Jusqu'à deux convois autonomes peuvent être jumelées pour desservir un même train ayant une demande élevée. Le modèle proposé est à nouveau un modèle multiflots avec contraintes supplémentaires qui est résolu par une technique de génération de colonnes couplée à un algorithme heuristique de séparation et évaluation progressive. DUBOIS (2001) introduit, dans le cas du problème avec horaires variables, des contraintes de synchronisation pour les trains portant le même numéro mais opérant des jours différents

et aussi pour les unités affectées au même train. Celles-ci sont d'abord relaxées et, lorsqu'elles sont violées, des décisions de branchement heuristiques sont prises pour les imposer. L'auteur a résolu en des temps acceptables des problèmes comportant 450 trains.

1.3 Génération de colonnes

La génération de colonnes, aussi appelée décomposition de Dantzig-Wolfe, a largement été utilisée pour résoudre diverses variantes du problème de tournées de véhicule. Avant de voir comment cette technique a été employée pour résoudre ces différents cas, on peut consulter trois articles présentant une synthèse de la génération de colonnes. Citons ainsi DESROSIERS et al. (1995) BARNHART et al. (1998) qui présentent la génération de colonnes comme un outil permettant de résoudre des programmes mathématiques en nombres entiers de très grande taille. En 1998, DESAULNIERS et al. (1998) complètent la synthèse de DESROSIERS et al. (1995) en présentant un modèle unifié permettant de modéliser une vaste gamme de problèmes d'affectation de ressources à des tâches avec contraintes de temps. Ces derniers décrivent l'utilisation d'une décomposition de Dantzig-Wolfe couplée à un algorithme de séparation et évaluation progressive permettant de résoudre la plupart des problèmes de ce type. La génération de colonnes telle que décrite dans cet article a été utilisée dans des contextes aussi variés que la confection d'horaires d'avion (DESAULNIERS et al. (1997)), d'horaires de véhicule avec dépôts multiples (RIBEIRO et SOUMIS (1994)), de tournées de véhicule avec fenêtres de temps (DESROCHERS et al. (1992), KOHL et MADSEN (1997)), et de tournées de véhicules auto-guidés (VILLENEUVE (2000)).

Récemment, la génération de colonnes a été appliquée à des problèmes hybrides de tournées de véhicules couplées à d'autres types de problèmes. Ainsi KLABJAN et al.

(2002) s'attaquent-ils au problème de construction des rotations d'équipage en se permettant de déplacer les heures de départ des vols tout en respectant le nombre d'avions disponibles. Plus tôt, CHRISTIANSEN (1999) se penchait sur la confection de tournées de bateau tout en considérant la gestion d'inventaire alors qu'en 2001, HAASE et al. (2001) présentaient le problème de génération simultanée de tournées d'autobus et d'horaires de chauffeurs.

1.4 Positionnement du mémoire

Si le nombre de publications sur les problèmes d'affectation de matériel roulant aux trains de passagers semble croître, nous pouvons noter que la plupart de celles-ci se situent plus au niveau de la planification à moyen ou long terme où il y a peu de place pour la prise en compte de détails opérationnels tels que l'orientation et la position des wagons. Nous présentons ici un outil d'aide à la décision plus proche du milieu opérationnel que de celui de la planification où nous avons accru le niveau de détail.

Dans les travaux que nous présentons, les décisions sont prises en considérant le nombre de billets réellement vendus par opposition aux estimations de ventes dont on dispose au niveau de la planification. De plus, ce modèle permet de déterminer quelles demandes accepter et quelles demandes refuser, un tel choix devant être fait dans le cas où le nombre de wagons disponibles est insuffisant en regard des demandes ou dans le cas où on s'autorise à refuser l'ajout de wagons supplémentaires pour cause de non-rentabilité. Finalement, mentionnons que la grande majorité des contraintes opérationnelles sont prises en compte dans notre modèle de sorte que les solutions produites peuvent être utilisées directement durant les opérations.

Chapitre 2

Operational Car Assignment at VIA Rail Canada

Ce chapitre présente un article, écrit en collaboration avec J.-F. Cordeau, G. Desautniers, J. Desrosiers et F. Soumis, qui a été publié en 2002 dans la revue *Transportation Research, Part B* (volume 36, numéro 9, pages 755-778). La première section procure un sommaire en français de cet article.

2.1 Sommaire

L'article que nous présentons dans le deuxième chapitre décrit l'implantation d'un logiciel permettant de résoudre le problème d'affectation de wagons aux trains de passagers tel que rencontré chez VIA. On y trouve en introduction des éléments qui situent les objectifs poursuivis et précisent qu'il s'agit-là du troisième et dernier volet de travaux dont le découpage est calqué sur la division de BUSSIECK et al. (1997) que nous avons déjà évoquée. Suivent une revue de la littérature, que nous avons étendue plus tôt, et une description formelle d'une version simplifiée du problème étudié : étant donné un ensemble de trains à couvrir pour lesquels on connaît la route des locomotives les desservant ainsi que le nombre minimum et désiré de wagons, étant

donné une répartition initiale des wagons en début d'horizon d'optimisation et une répartition finale imposée par la planification, le problème d'affectation consiste à trouver un ensemble de routes à meilleur coût pour les wagons disponibles de façon à respecter des contraintes opérationnelles telles que la couverture minimale des trains, le nombre maximal de wagons tirés par une locomotive, le temps maximal séparant deux entretiens d'un wagon et le temps minimal requis pour modifier un convoi. La structure de coût est déduite des coûts d'opération des wagons auxquels on soustrait les bénéfices anticipés par l'affectation de wagons supplémentaires à certains trains.

Deux modèles sont présentés pour résoudre ce problème et ils font tous deux appel à un réseau spacio-temporel où un noeud représente aussi bien la position spacio-temporelle d'un wagon que sa position stratégique dans le convoi, c'est-à-dire, si à un certain moment et à un certain endroit donné le wagon se situe en fin ou en milieu de convoi. Prendre en compte cette position stratégique permet de valider la contrainte selon laquelle le découplage d'un wagon en milieu de convoi est plus long que celui d'un wagon en queue. Le premier modèle, plus intuitif, présente une approche classique de routage de véhicule (VRP) où l'on restreint cependant le domaine réalisable à l'aide de variables inhabituelles et compliquantes : des permutations. Ces variables permettent en tout temps de garder trace de la position de chaque wagon dans le convoi auquel il appartient et assurent que, lorsque le temps disponible pour la modification de convoi ne permet que des couplages-découplages en queue, les wagons concernés sont effectivement en queue de convoi.

Le deuxième modèle tire profit du fait que ces variables compliquantes n'apparaissent pas dans l'objectif et ne sont là que pour limiter le domaine réalisable. Il suffit donc de trouver une restriction équivalente du domaine que l'on puisse exprimer en fonction de variables plus conventionnelles, plus simples à manipuler. Nous trouvons dans l'article une restriction équivalente basée sur la notion d'arcs (ou décisions) conflictuels. Grossièrement, on exploite le fait que si un wagon w_1 est accroché en queue

de convoi auquel appartenait déjà un wagon w_2 , w_2 ne peut être décroché dans un temps réservé au décrochage en queue tant et aussi longtemps que w_1 poursuivra sa route avec ce convoi.

Pour résoudre ce deuxième modèle, nous proposons une approche heuristique, basée sur une décomposition de Dantzig-Wolfe imbriquée dans un arbre de branchement, de même qu'un algorithme permettant de retrouver les positions de wagons à partir d'une solution réalisable. Ce dernier algorithme, qui est détaillé en annexe de l'article, est d'ailleurs la base d'une preuve par construction de l'équivalence de la restriction exprimée à l'aide de permutations et de celle exprimée à l'aide de variables de décision binaires.

Enfin, nous trouvons dans l'article un survol de six extensions faites au problème simplifié de façon à ce que le logiciel satisfasse à toutes les exigences opérationnelles de VIA. Chacune des extensions est discutée de même que les modifications apportées au modèle pour la prendre en compte. Finalement, l'article présente une synthèse des résultats obtenus sur 140 cas classés en 14 types.

2.2 Introduction

Faced with high competition from other means of transportation, North American passenger railways are constantly investigating new ways to reduce their costs and improve their service. One promising direction is to increase the utilization of the rolling stock by a better assignment of the available locomotives and cars to the scheduled trains. The units of equipment assigned to a train are said to form a *consist*. While some companies prefer to operate their trains with unbreakable consists for operational convenience, others believe that the costs and potential delays associated with *switching* activities are more than compensated by the savings that may be realized by being able to separate and recombine locomotives and cars at various locations in the rail network. This additional flexibility may indeed reduce the number of units of equipment needed to run a train schedule, lower the *deadhead* mileage (i.e., cars being moved without carrying passengers), and increase both profitability and the level of service provided by allowing cars to be assigned where they are the most useful.

This work focuses on the case where switching is allowed at various locations in the network. In this case, locomotives and cars must be assigned simultaneously to the scheduled trains because the minimum connection time between two consecutive trains covered by the same locomotives depends on whether cars need to be switched during the connection. In addition, when switching has to be performed, the connection time depends on the position of the switched cars within the consist since more time is required to switch a car that is in the body of the consist than one that is at its end.

This paper addresses the operational car assignment problem (OCAP) encountered at VIA Rail Canada (hereafter referred to as VIA) although the proposed model and

solution methodology may also be applied to other passenger railways that rely on a similar planning process. At VIA, the OCAP is the last part of a three-phase planning process.

Given the weekly train schedule valid for a whole season, the first phase seeks to determine a minimum-cost set of weekly locomotive and car cycles that satisfy the car demand on all trains as well as maintenance requirements and some level of operational constraints. A *cycle* is a sequence of consecutive trains that starts and ends at a maintenance center. Car demands, which are provided by the marketing department based on historical data, correspond to maximum demands over the season. This ensures that the solution to the OCAP will provide enough capacity for any week in the season even if the actual demand slightly varies from week to week. This first phase is usually performed a few months before the beginning of the season.

Because of holidays and other special events, the trains operated during a given week may differ slightly from those planned in the seasonal schedule: trains may be canceled, added or simply rescheduled to account for changes in the demand. The aim of the second phase is thus to adapt the set of cycles determined in the first phase so as to incorporate these changes. The problem tackled in this phase is very similar to that of the first phase with the exception that a specific, dated scheduling horizon is considered. To ensure continuity, the revised locomotive and car cycles must also be compatible, at the beginning and the end of the scheduling horizon, with those planned in the first phase. Updated cycles for a given week are normally computed one month ahead.

The last phase of the planning process handles short-term demand revisions formulated by the marketing department which is responsible for tracing ticket sales on a daily basis. Upward and downward revisions are usually forwarded every day to the scheduler and concern trains departing during the next few days. Revisions for the

current day are not accepted. Car requests arising from upward revisions require fast response times since additional tickets may be sold as soon as cars are assigned to fulfill them. Obviously, car requests should receive a positive response only when cars can be assigned to fulfill them with a positive expected profit and without canceling any previously accepted request. This phase is performed daily and considers a scheduling horizon of about a week starting on the next day. Because locomotive cycles are the basis of the crew schedules and cannot be modified, the update concerns only car cycles.

This paper addresses the OCAP arising in this last phase which can be briefly stated as follows. Given a weekly train schedule, planned car cycles and a set of requests for additional cars, the problem consists of finding maximum expected profit car cycles that are compatible with the planned cycles at the beginning and the end of a given scheduling horizon, fulfill all previously accepted car demands unless revised downwards, and satisfy all operational constraints. Two important sets of operational constraints that must be respected are the maintenance requirements and the minimal switching times. The first requires that each unit of equipment be inspected and serviced regularly. The second ensures that enough time is available to switch cars during connections with switching activities.

2.2.1 Review of the literature

Very few contributions concerning the management of passenger railway equipment can be found in the operations research literature. RAMANI and MANDAL (1992) developed an optimization-based decision support system that aims at minimizing fleet size. Based on previous works by RAMANI and RAGHURAM (1980) and RAMANI (1981), this system, which was implemented at Indian Railways, uses a local

improvement heuristic that examines train connections at individual stations of the network. Locomotives and cars are treated separately and equipment switching is not considered. More recently, a system was developed by BEN-KHEDER et al. (1997) to optimize the assignment of equipment modules to trains at the French Railways SNCF. Each module is composed of a certain number of locomotives and cars that cannot be separated. Because the problem consists of assigning full modules to the trains, the methodology developed by the authors does not apply to our context.

BUSSIECK, WINTER and ZIMMERMANN (1997) surveyed mathematical programming methods for public rail transport planning. The authors state that the problem of assigning rolling stock to a set of scheduled trains can be formulated as a multi-depot vehicle scheduling problem and present a review of several papers on that subject. Unfortunately, this statement is true only when the train consists are unbreakable.

Very recently, CORDEAU et al. (2001a) developed an optimization system to solve the equipment cycling problem encountered in the first phase of the planning process described above. This system is based on a multi-commodity network flow model with additional constraints that incorporate the level of flexibility required in an industrial application. In particular, the model handles different types of consists as well as maintenance requirements, substitution possibilities, equipment availability and penalties for car switching. This model is solved by a column generation approach embedded in a heuristic branch-and-bound search. On test instances from VIA involving several hundred trains per week, the system succeeded in finding good heuristic solutions with a smaller cost and a smaller number of car switchings than those produced by human planners. This approach is the core of a system that is now implemented at the company.

In parallel with the preceding work, CORDEAU, SOUMIS and DESROSIERS (2000, 2001b) presented an exact Benders decomposition approach for the simultaneous

assignment of locomotives and cars to passenger trains. The problem they address is very similar to that treated by CORDEAU et al. (2001a) but minor complicating considerations were left out of the model. Within reasonable CPU times, the authors succeeded in solving to optimality instances involving more than 350 trains per week.

Finally, the reader is referred to the work of CORDEAU, TOTH and VIGO (1998) for a recent survey of optimization models for train routing and scheduling.

2.2.2 Contribution

In this paper, we describe the model and solution methodology that was developed to solve the operational car assignment problem at VIA. This represents the first approach that explicitly considers the positions of the cars within the train consists. Keeping track of these positions increases the complexity of the problem but is necessary because minimum switching times depend on the positions of the switched cars. We thus present a particular solution approach to handle these complicating constraints. On VIA test problems, this approach produced solutions of good quality and is fast enough to be used in an operational planning context.

2.2.3 Paper Overview

The rest of the paper is organized as follows. The next section provides a detailed definition of a basic version of the OCAP. Section 2.4 introduces two mathematical formulations of the problem after describing the underlying network structure. The first formulation is very intuitive but turns out to be intractable. A second formulation that can be solved effectively is thus introduced. Next, Section 2.5 proposes

a solution approach for the second model. Then, Section 2.6 presents several extensions to the basic OCAP that were encountered at VIA and discusses their impact on the second model and the solution approach. Finally, computational results are reported in Section 2.7 and conclusions are drawn in the last section.

2.3 Basic Problem Description

In practice, the OCAP includes special features and complex constraints that may lead to a mathematical formulation difficult to understand. For ease of exposition, we thus begin by addressing a simplified version of the problem. Extensions to this basic problem will be discussed in Section 2.6.

The basic OCAP can be described as follows. Consider a set of trains T to be operated during a given time horizon that comprises the ordered set of days $D = \{d_0, \dots, d_{\max}\}$. Each train $t \in T$ is defined by an origin station $\text{from}(t)$, a destination station $\text{to}(t)$, a departure time $\text{dep}(t)$, an arrival time $\text{arr}(t)$, a car demand n_t , a number of additional requested cars q_t , and a maximum number of cars that can be assigned to it p_t . The car demands are usually provided by the marketing department while the upper bounds on the number of cars that can be assigned to a train are derived from the pulling capacity of the locomotives already assigned to it. The gap between $n_t + q_t$ and p_t allows for deadhead cars that can be moved empty to fulfill demand elsewhere in the network, or return to a maintenance center for a periodic inspection. Note that downward demand revisions can be taken into account directly by reducing n_t . Hence, requests for car removal do not need to be considered explicitly in the model. We assume that no trains run at night, where the night is defined as a period of time long enough to inspect and service a consist (typically, a few hours).

Although stops can be made at intermediate stations along a train to board and detrain passengers, we assume without loss of generality that equipment cannot be switched during these stops. Therefore, such intermediate stations do not need to be considered in the problem. Let S be the set of end-of-train stations. Each station $s \in S$ is characterized by the minimum time required at that station to switch a car located in the body of a consist $\mathbf{sw}_{\text{body}}(s)$, and to switch a car located at the tail end of a consist $\mathbf{sw}_{\text{tail}}(s)$. A car being switched out (or in) of a consist is considered in its body if at least one car behind it is not switched out (or in). Since in-body switching requires more maneuvers than tail switching, we assume that $\mathbf{sw}_{\text{body}}(s) \geq \mathbf{sw}_{\text{tail}}(s)$, $\forall s \in S$. Also, when both coupling and uncoupling of cars need to be performed on a consist, we assume that cars to be uncoupled are switched out of the consist before switching in the cars to be coupled.

Imposing the minimal switching time constraints defined above requires keeping track of the position of each car in the consist assigned to each train. This tracking process is performed through a set of positioning constraints which ensure that all cars switched at the tail end of a consist, either in or out, are really at its tail end and that all cars remaining in a consist from one train to another keep the same relative order.

To fulfill car demands, a sufficiently large set of passenger cars W is available. Each car must be maintained at least once every m days in order to comply with safety regulations. Such maintenance checks must be performed during night-time at a unique maintenance center which is located close to a station that will be referred to as the *base*. In the basic OCAP, we assume that all cars are equivalent with respect to car demands, i.e., any car can be used to fulfill any demand. Since car cycles originating and ending at the base are sought, cars available at the base are undifferentiated. However, cars that are outside the base at the start of the horizon are distinguishable and classified in two subsets that will yield initial conditions.

Subset $ICT \subset W$ contains all cars that are in a consist outside the base whereas subset $ICS \subset W$ gathers all other cars not in a consist (available) outside the base. For each car $w \in ICT$, t_w and i_w denote the last train it covered before the horizon and its position at the start of the horizon, respectively. For each car $w \in ICS$, s_w denotes the station where it is located at the start of the horizon. Finally, for each car $w \in ICT \cup ICS$, d_w denotes the number of days since it was last inspected ($d_w < m$). Cars are also subject to final conditions but not on an individual basis. Hence, two sets of constraints, FCT and FCS, are defined. A constraint $c \in FCT$ (or $c \in FCS$) indicates that a car with at most $m - d_c$ days since its last inspection is expected on train t_c (or is to be available at station s_c) at the end of the horizon. This car will be connected to the end of a planned cycle lasting another d_c days.

Given also as an input to the problem is the set R of all locomotive cycles that contain at least one train in the horizon. These cycles are essential to determine whether or not switching activities can occur between two consecutive trains as well as to identify cars that are switched. In the basic OCAP, we assume that exactly one locomotive is assigned to each train $t \in T$.

The cost/revenue structure of the basic problem is as follows. Each car assigned to a train $t \in T$ incurs an operational cost c_t . This cost is usually computed as the distance traveled along train t multiplied by a per mile cost that may take into account fuel consumption, inspection and servicing, as well as running repairs. Since the expected revenue yielded by minimal car demands corresponds to a fixed amount, only the revenues expected by fulfilling additional car requests need to be considered. Denote by q_t the number of additional cars requested on train $t \in T$, and by r_t the expected revenue obtained by providing one car. We assume that this revenue is the same for each car provided. Finally, let $T_q \subseteq T$ be the subset of trains for which $q_t > 0$.

Given all these inputs, the basic OCAP consists of determining a set of cycles for the

available cars that maximizes the total expected profits (i.e., revenues minus costs) and satisfies train car demands, maintenance requirements, car initial and final conditions, pulling capacities, and minimum switching times.

2.4 Mathematical Formulations

In this section, we describe the network structure underlying the OCAP. We then introduce a first formulation for the basic OCAP that includes compact constraints for car positioning. While intuitive, this formulation also appears to be intractable. To alleviate this, we propose a transformation that yields a less intuitive, nonlinear, integer mathematical program for which a solution algorithm can be designed.

2.4.1 Time-Space Network

In order to describe the network structure of the OCAP, we need to introduce the *atom* concept that relies on the following definition.

Definition 1. *Two consecutive trains $t_1 \in T$ and $t_2 \in T$ on a locomotive cycle $r \in R$ are said to be binding if and only if $\text{dep}(t_2) - \text{arr}(t_1) < \text{sw}_{\text{tail}}(\text{to}(t_1))$, which means that there is no time between t_1 and t_2 to perform switching. Hence if two trains are binding, we can deduce that any car will either cover both or neither of them.*

Each locomotive cycle $r \in R$ is partitioned into subsequences of trains (hereafter called *atoms*) by separating it between each pair of consecutive trains that are not binding. Let A be the set of *atoms*. For each atom $a \in A$, denote by $\text{first}(a)$

and $\text{last}(a)$ its first and last train, respectively. Also, when appropriate, $\text{prev}(a)$ represents the atom that precedes a along the cycle to which it belongs.

Now, let $G = (V, \mathcal{E})$ be a time-space network where V and \mathcal{E} represent its node and arc sets, respectively. Node set V contains the following six types of nodes (see Figure 2.1). For each day $d \in D$ and each station $s \in S$, there exists a pair of nodes SOD_d^s and EOD_d^s to represent, respectively, the start and the end of day d at station s . Then, for each atom $a \in A$, there exist up to four nodes (assuming $t_1 = \text{first}(a)$ and $t_2 = \text{last}(a)$) as follows: $D_{\text{body}}^{t_1}$ associated with station $\text{from}(t_1)$ and time $\text{dep}(t_1) - \text{sw}_{\text{body}}(\text{from}(t_1))$ on the day t_1 operates, to represent a departure following an in-body switching; $D_{\text{tail}}^{t_1}$ associated with station $\text{from}(t_1)$ and time $\text{dep}(t_1) - \text{sw}_{\text{tail}}(\text{from}(t_1))$ on the day t_1 operates, to represent a departure following a tail switching; $A_{\text{body}}^{t_2}$ associated with station $\text{to}(t_2)$ and time $\text{arr}(t_2) + \text{sw}_{\text{body}}(\text{to}(t_2))$ on the day t_2 operates, to represent an arrival followed by an in-body switching; and $A_{\text{tail}}^{t_2}$ associated with station $\text{to}(t_2)$ and time $\text{arr}(t_2) + \text{sw}_{\text{tail}}(\text{to}(t_2))$ on the day t_2 operates to represent an arrival followed by a tail switching. Note that $D_{\text{body}}^{t_1}$ is omitted when there is no time to perform in-body switching before train t_1 , that is, when $\text{prev}(a)$ exists and $\text{dep}(t_1) - \text{sw}_{\text{body}}(\text{from}(t_1)) < \text{arr}(\text{last}(\text{prev}(a)))$. Similarly, if in-body switching cannot be performed after train t_2 due to minimal switching time, node $A_{\text{body}}^{t_2}$ is omitted. Finally, all nodes that represent a departure or an arrival on a day outside the horizon are also discarded.

Arc set \mathcal{E} contains five types of arcs: *atom*, *init*, *final*, *wait* and *night*. For each locomotive cycle $r \in R$, consider the collection of D -type and A -type nodes created using the atoms of r , and chronologically order them. A series of *atom* arcs link each of those D -type nodes to each of its following A -type nodes (see Figure 2.1). Such an arc represents the assignment of a car to the sequence of trains found on r between the corresponding departure and arrival nodes, preceded and succeeded by a switching activity. *Init* and *final* arcs, which are built similar to the *atom* arcs,

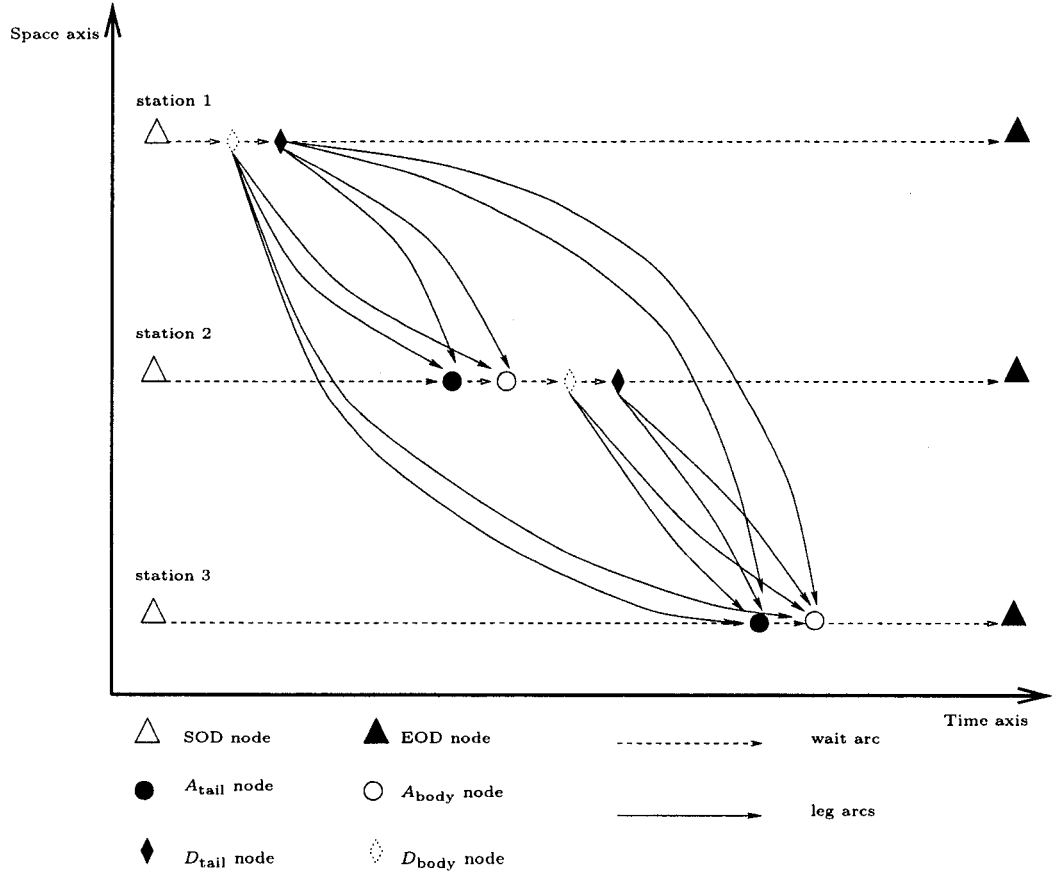


Figure 2.1: Part of a network created by a two-atom cycle

are used to model initial and final conditions. For each car $w \in ICT$, a series of *init* arcs links node $\text{SOD}_{d_0}^{\text{base}}$ to every A -type node created from the locomotive cycle containing train t_w , while for each car $w \in ICS$, a single *init* arc links node $\text{SOD}_{d_0}^{\text{base}}$ to node $\text{SOD}_{d_0}^{s_w}$. Similarly, for each final condition $c \in \text{FCT}$, a series of *final* arcs link every D -node created from the locomotive cycle containing train t_c to node $\text{EOD}_{d_{\max}}^{\text{base}}$, while for each final condition $c \in \text{FCS}$, a single *final* arc links node $\text{EOD}_{d_{\max}}^{s_c}$ to node $\text{EOD}_{d_{\max}}^{\text{base}}$. Note that all *atom*, *init* and *final* arcs spanning more than m days are omitted. For each day $d \in D$ and each station $s \in S$, *wait* arcs chronologically link all the nodes associated with the given day and station. These arcs represent waiting

during daytime between two assignments. Finally, for each station $s \in S$ other than the base and each day $d \in [d_0, d_{\max} - 1]$, node EOD_d^s is linked to node SOD_{d+1}^s by a *night* arc. Such an arc represents waiting during night-time between two assignments.

To illustrate, consider the network illustrated in Figure 2.1 and a path that starts from the SOD node in station 1, goes through the D_{body} node at station 1, the A_{body} node at station 2, the D_{tail} node at station 2, the A_{tail} node at station 3, and finally the EOD node at station 3. This path would imply that the consist to which the car belongs will undergo an in-body switching before the departure of train t_1 in station 1, an in-body switching following the arrival of train t_1 in station 2, a tail switching before the departure of train t_2 in station 2 and a tail switching after the arrival of train t_2 in station 3.

With each arc $e \in \mathcal{E}$ is associated a cost c_e which is equal to 0 if e is either a *wait* or a *night* arc, and to $\sum_{t \in T_e} c_t$ otherwise, where T_e denotes the set of trains covered by arc e .

Based on the graph $G = (V, \mathcal{E})$, we define a time-space network $G^k = (V^k, E^k)$ for each commodity $k \in K$, where a commodity is a couple (w, d) , $w \in W$, $d \in D$, that represents the car w beginning a cycle on day d . These networks allow one to implicitly enumerate every possible cycle lasting m days or less for every available car. K is partitioned in two subsets: K^{init} and K^{und} .

The commodities in K^{init} are those associated with a car $w \in \text{ICT} \cup \text{ICS}$ and day d_0 although the cycle of w has started before d_0 . Recall that cars in $\text{ICT} \cup \text{ICS}$ are those which are outside the base at the start of the horizon and which yield initial conditions. Because the cycle of such a car has started before day d_0 , the car must return to the base before the end of day $d_0 + m - 1 - d_w$. Hence, for each $k \in K^{\text{init}}$, V^k contains all nodes in V associated with a day in $P^k = [d_0, \min\{d_{\max}, d_0 + m - 1 - d_w\}]$,

and E^k contains all arcs whose end nodes belong to V^k except the non corresponding *init* arcs and all *final* arcs associated to a final condition constraint $c \in \text{FCT} \cup \text{FCS}$ for which $d_0 + m - 1 - d_w < d_{\max} + d_c$.

Commodities in K^{und} are associated with cars available at the base at the start of the horizon. Because they are not part of a planned cycle, these cars need not be differentiated. Hence, for each $k \in K^{\text{und}}$, V^k contains all nodes in V associated to a day in $P^k = [d, \min\{d_{\max}, d + m - 1\}]$ where d is the commodity's corresponding day, and E^k all arcs whose end nodes belong to V^k except all *init* arcs and all *final* arcs associated to a constraint $c \in \text{FCT} \cup \text{FCS}$ for which $d + m - 1 < d_{\max} + d_c$.

Finally, each network G^k contains a single source node $\text{src}^k = \text{SOD}_\delta^{\text{base}}$, where δ is equal to d , the day associated with commodity k , if $k \in K^{\text{und}}$, and to d_0 otherwise. Each network G^k also contains a set of sink nodes $\text{SNK}^k = \{\text{EOD}_\delta^{\text{base}} | \delta \in P^k\}$, where P^k is defined as above depending on whether k belongs to K^{init} or K^{und} .

2.4.2 Intuitive Formulation

Given these time-space networks, we present in this section an intuitive formulation for the basic OCAP that comprises compact car positioning constraints containing functions as variables. This unusual formulation will be transformed in the next section into a less intuitive, nonlinear, integer mathematical program.

Let $E = \bigcup_{k \in K} E^k$ be the set of all arcs in the networks G^k , $k \in K$. For each arc $e \in E$, define a binary variable X_e to indicate the flow on arc e and denote by $X = (X_e)_{e \in E}$ the vector of these flow variables. Also, for each train $t \in T_q$, define a non-negative integer variable Y_t to count the number of additional car requests satisfied for train t and denote by $Y = (Y_t)_{t \in T_q}$ the vector of these counter variables. With these two

types of variables, all constraints can be expressed as linear inequalities except the car positioning constraints.

Car positioning corresponds to selecting one of the possible car permutations for each train. However, given that cars can be switched in and out of a consist, either in its body or at its tail end, not all permutations are valid. They must be computed by following the train sequences along each locomotive cycle. In this context, car positioning constraints must ensure that every tail switching maneuver is applied to a car positioned at its tail end or belonging to a block of switched cars positioned at its tail end. They must also ensure that all cars not switched between two consecutive trains on a locomotive cycle do not change their relative positions. Note that it is not necessary to ensure that every in-body switching maneuver be performed on a car that is really in the body of a consist since in-body switching requires more time than tail switching. Finally, car positioning constraints must also be defined to ensure that initial car positions are respected.

All these constraints can be expressed using the following sets and mappings. Given a vector X and a train $t \in T$, let

$$\begin{aligned} W_t(X) &= \text{set of cars assigned to train } t; \\ W_t^{\text{tail}-}(X) &= \text{set of cars switched in at the tail end of train } t \text{ before its departure;} \\ W_t^{\text{tail}+}(X) &= \text{set of cars switched out at the tail end of train } t \text{ upon its arrival.} \end{aligned}$$

Any permutation of these cars can be given by a mapping of $W_t(X)$ onto $[1, |W_t(X)|]$.

To represent this, define

$$\begin{aligned}
\beta_t^-(X) &= \text{set of all mappings } \sigma_t : W_t(X) \mapsto [1, |W_t(X)|] \\
&\quad \text{such that } \sigma_t(w) > \sigma_t(w') \text{ for all pairs} \\
&\quad \text{of cars } (w, w') \in W_t^{\text{tail}-}(X) \times (W_t(X) \setminus W_t^{\text{tail}-}(X)); \\
\beta_t^+(X) &= \text{set of all mappings } \sigma_t : W_t(X) \mapsto [1, |W_t(X)|] \\
&\quad \text{such that } \sigma_t(w) > \sigma_t(w') \text{ for all pairs} \\
&\quad \text{of cars } (w, w') \in W_t^{\text{tail}+}(X) \times (W_t(X) \setminus W_t^{\text{tail}+}(X)).
\end{aligned}$$

Tail switching can then be enforced by restraining σ_t to lie within $\beta_t^-(X) \cap \beta_t^+(X)$, $\forall t \in T$.

Next, denote by TT the set of pairs of consecutive trains appearing along the locomotive cycles in R , that is, $(t, t') \in TT$ if and only if there exists a locomotive cycle $r \in R$ such that r covers t and t' consecutively in that order. Given a vector X and a pair of trains $(t, t') \in TT$, let

$$\begin{aligned}
W_{t,t'}(X) &= W_t(X) \cap W_{t'}(X) = \text{set of cars assigned to both trains;} \\
C_{t,t'}(X) &= \text{set of pairs of mappings } (\sigma_t, \sigma_{t'}) \text{ defining a permutation of } W_t(X) \text{ and} \\
&\quad W_{t'}(X), \text{ respectively, such that } (\sigma_t(w) - \sigma_t(w'))(\sigma_{t'}(w) - \sigma_{t'}(w')) > 0 \\
&\quad \text{for all pairs of cars } (w, w') \in W_{t,t'}(X).
\end{aligned}$$

This set of mapping pairs can be used to preserve the relative ordering of the cars not switched between two trains of a locomotive cycle.

Using this notation, the basic OCAP can be formulated as:

$$\begin{aligned}
&\text{Maximize} \quad - \sum_{k \in K} \sum_{e \in E^k} c_e X_e + \sum_{t \in T_q} r_t Y_t \\
&\text{subject to:}
\end{aligned} \tag{2.1}$$

$$\mathcal{A}X + \mathcal{B}Y \leq b \quad (2.2)$$

$$\sigma_t \in \beta_t^+(X), \quad \forall t \in T \quad (2.3)$$

$$\sigma_t \in \beta_t^-(X), \quad \forall t \in T \quad (2.4)$$

$$(\sigma_t, \sigma_{t'}) \in C_{t,t'}(X), \quad (t, t') \in TT \quad (2.5)$$

$$\sigma_{t_w}(w) = i_w, \quad \forall w \in \text{ICT} \quad (2.6)$$

$$X_e \text{ binary}, \quad \forall e \in E \quad (2.7)$$

$$Y_t \in [0, q_t], \text{ integer}, \quad \forall t \in T_q. \quad (2.8)$$

In this formulation, the objective function (2.1) aims at maximizing anticipated profits, that is, anticipated revenues minus operational costs. Equation (2.2) expresses in matrix form all linear operational constraints to be detailed in the next section, where \mathcal{A} , \mathcal{B} and b are coefficient matrices of appropriate dimensions. Constraint sets (2.3)–(2.6) represent the car positioning constraints which ensure that minimal switching times are respected, cars assigned to consecutive trains do not change their relative positions, and car initial positions are considered. Finally, constraints (2.7) and (2.8) provide binary requirements on the arc flow variables, and the lower and upper bounds on the counter variables as well as their integrality requirements, respectively.

While (2.1), (2.2), (2.7) and (2.8) define a classical mathematical program, the nature of the mapping variables σ_t , $t \in T$, leads to a program for which no solution method is known. The following section proposes a transformation that yields a tractable, though less intuitive, formulation.

2.4.3 Tractable Formulation

A closer look at formulation (2.1)–(2.8) shows that the complicating mapping variables σ_t do not appear in the objective function. Thus, if there exist constraints in terms of the arc flow variables X_e , $e \in E$, that are satisfied if and only if there exists at least one family of mappings $(\sigma_t)_{t \in T}$ satisfying (2.3)–(2.6), the latter constraints can be dropped in favor of the former. The values of the mapping variables can then be determined *a posteriori*. The following definition is required to construct the equivalent set of constraints.

Definition 2. Let $e \in E$ be an *atom* or *init* arc whose arrival node corresponds to a *tail switching*. An *atom*, *init* or *final* arc $e' \in E$ is said to *clash* with e if 1) both arcs are generated from the same locomotive cycle; 2) the time associated with the arrival node of e' is greater than that associated with the arrival node of e ; and 3) one of the following two conditions is satisfied: i) the departure node of e' corresponds to a *tail switching* and its associated time lies strictly within the interval defined by the times associated with the departure and arrival nodes of e ; or ii) e and e' are both *init* arcs and the initial position of the car associated with e' is greater than that of the car associated with e .

Intuitively, two clashing arcs cannot be chosen in a feasible solution as the car associated with arc e would be ahead of the car associated with arc e' when it is time to switch out the first of these cars at the tail end of the consist.

Denoting by $E_{\text{tail}} \subset E$ the subset of *atom* and *init* arcs corresponding to a tail switching at the arrival node, and by E_e^{clash} the subset of arcs clashing with an arc $e \in E_{\text{tail}}$, the following proposition indicates how to replace the car positioning constraints (2.3)–(2.6) by a set of nonlinear constraints expressed in terms of the arc flow variables X_e , $e \in E$.

Proposition 1. *There exists at least one family of mappings $(\sigma_t)_{t \in T}$ satisfying constraints (2.3)–(2.6) if and only if there exist arc flow variables X_e , $e \in E$, such that*

$$X_e - \sum_{e' \in E_e^{\text{clash}}} X_{e'} = 0, \quad \forall e \in E_{\text{tail}}. \quad (2.9)$$

The formal proof of this proposition is rather involved. Here, we provide only its main ideas.

Sketch of the Proof: The proof of the “only if” part (\Rightarrow) is simple. It consists of showing that, if two clashing arcs $e \in E_{\text{tail}}$ and $e' \in E_e^{\text{clash}}$ are used (i.e., $X_e > 0$ and $X_{e'} > 0$), then one of the car positioning constraints (2.3)–(2.6) is violated. On the one hand, the car w' associated with e' is positioned behind the car w associated with arc e once switched in at the tail end of the consist or due to the initial positions. On the other hand, switching out w at the tail end of the consist after switching in w' requires that w be positioned behind w' , leading to a contradiction.

The proof of the “if” part (\Leftarrow) is less straightforward and relies on an algorithm that finds, when constraints (2.9) are satisfied, a family of mappings $(\sigma_t)_{t \in T}$ satisfying constraints (2.3)–(2.6). Such an algorithm is briefly discussed in Section 2.5.2 and described in the Appendix. \square

Proposition 1 allows us to give a tractable formulation for the basic OCAP. This new formulation requires the following additional notation for describing the linear operational constraints (2.2).

Since the trains composing an atom are covered by the exact same equipment, train demand and pulling constraints can be aggregated along the atoms. Let $a(t)$ be the atom that contains train $t \in T$ and, for every $a \in A$, let $\rho_a = \min_{t \in a} p_t$, $\nu_a = \max_{t \in a} n_t$ and $E^{k,a}$ be the set of arcs of E^k corresponding to a . Furthermore, denote by E_w^{init} the

set of *init* arcs associated with $w \in \text{ICT} \cup \text{ICS}$; E_c^{final} the set of *final* arcs associated with $c \in \text{FCT} \cup \text{FCS}$; $V_{\text{base}} = \{\text{SOD}_{\text{base}}^d, \text{EOD}_{\text{base}}^d\}_{d \in D}$ the set of start and end of day nodes at the base; $I^{k,v}$ the set of arcs of E^k entering node v ; and $O^{k,v}$ the set of arcs of E^k exiting node v .

Using this notation, the basic OCAP can also be formulated as:

$$\text{Maximize} \quad - \sum_{k \in K} \sum_{e \in E^k} c_e X_e + \sum_{t \in T_q} r_t Y_t \quad (2.10)$$

subject to:

$$\sum_{k \in K} \sum_{e \in E^{k,a}} X_e \geq \nu_a, \quad \forall a \in A \quad (2.11)$$

$$\sum_{k \in K} \sum_{e \in E^{k,a}} X_e \leq \rho_a, \quad \forall a \in A \quad (2.12)$$

$$\sum_{k \in K} \sum_{e \in E^{k,a(t)}} X_e - Y_t \geq n_t, \quad \forall t \in T_q \quad (2.13)$$

$$\sum_{e \in E_w^{\text{init}}} X_e = 1, \quad \forall w \in \text{ICT} \cup \text{ICS} \quad (2.14)$$

$$\sum_{e \in E_c^{\text{final}}} X_e = 1, \quad \forall c \in \text{FCT} \cup \text{FCS} \quad (2.15)$$

$$\sum_{e \in O^{k,\text{src}^k}} X_e = 1, \quad \forall k \in K \quad (2.16)$$

$$\sum_{e \in I^{k,v}} X_e - \sum_{e \in O^{k,v}} X_e = 0, \quad \forall k \in K, \forall v \in V \setminus V_{\text{base}} \quad (2.17)$$

$$\sum_{v \in \text{SNK}^k} \sum_{e \in I^{k,v}} X_e = 1, \quad \forall k \in K \quad (2.18)$$

$$X_e - \sum_{e' \in E_e^{\text{clash}}} X_{e'} = 0, \quad \forall e \in E_{\text{tail}} \quad (2.19)$$

$$X_e \text{ binary}, \quad \forall e \in E \quad (2.20)$$

$$Y_t \in [0, q_t], \text{ integer}, \quad \forall t \in T_q. \quad (2.21)$$

Besides constraint set (2.19) that replaces the car positioning constraints (2.3)–(2.6), this new formulation comprises the same objective function (2.10) and constraint sets (2.20)–(2.21), while providing through constraints (2.11)–(2.18) an explicit expression for constraints (2.2) that were expressed in matrix form. Constraints (2.11) ensure that train car demands are satisfied via the assignment of cars to the atoms. Similarly, constraints (2.12) guarantee that train pulling capacities are not exceeded. Constraints (2.13) are used to count the number of fulfilled car requests for each train. Note that the computation of these requests can be included in (2.11) by aggregating car requests along the atoms and replacing each counter variable Y_t , $t \in T^q$, by a series of binary variables associated with the atoms, each variable corresponding to a number of fulfilled requests. Relations (2.14) and (2.15) define the initial and final conditions, while relations (2.16)–(2.18) provide for each commodity $k \in K$ the usual path structure from the source node src^k of network G^k to one of its sink node $v \in \text{SNK}^k$.

Note that the number of constraints (2.19) can be reduced considerably by aggregating the flow on all arcs $e \in E_{\text{tail}}$ that originate from the same generic arc in \mathcal{E} . Note also that the integrality requirements on the Y_t variables are not mandatory.

2.5 Solution Methodology

This section describes a heuristic solution approach for the basic OCAP. This approach, which is based on the unified framework proposed by DESAULNIERS et al. (1998) for deterministic time constrained vehicle routing and crew scheduling problems, allows the inclusion of all practical extensions encountered at VIA and discussed in the next section. The algorithm for positioning the cars on the trains once the car cycles are computed is presented at the end of the section.

2.5.1 Computing Car Cycles

To compute car cycles, model (2.10)–(2.21), which corresponds to a large-scale nonlinear integer program, is solved by a column generation method embedded in a heuristic branch-and-bound approach. First, an extension of the Dantzig-Wolfe decomposition principle (DANTZIG and WOLFE, 1960) is applied to this model after relaxing the nonlinear constraints (2.19) that will be dealt with during the branch-and-bound process. This decomposition process separates the problem into a subproblem and a master problem.

The subproblem is composed of a modified objective function, constraints (2.16)–(2.18) as well as the binary requirements on the arc flow variables (2.20). The objective function, which is not detailed here for reasons of conciseness (see DESAULNIERS et al. (1998) for more information), aims at finding a least reduced cost variable for the master problem. It consists of the first double sum of the original objective function (2.10) where the coefficients of the X_e variables are modified by the dual variable values of the master problem constraints. This subproblem can be divided into $|K|$ subproblems, one for each commodity $k \in K$. Each of these subproblems is a classical shortest path problem in G^k from the single source node src^k to one of the sink nodes $v \in \text{SNK}^k$. The set of feasible paths in G^k is denoted by Ω^k .

The master problem retains the objective function (2.10), the linking constraints (2.11)–(2.15) and the integer requirements on the counter variables (2.21). After applying the decomposition principle, these relations are expressed using a new type of variables called path variables. Such a variable θ_p represents the flow on path $p \in \Omega^k$ in G^k . Furthermore, for each path $p \in \Omega^k$, denote by c_p its cost and define the following binary parameters: g_{ap} is equal to 1 if p covers atom $a \in A$; h_{tp} is equal to 1 if p covers train $t \in T$; i_{wp} is equal to 1 if p contains an *init* arc associated with

car $w \in \text{ICT} \cup \text{ICS}$; and f_{cp} is equal to 1 if p contains a *final* arc associated with constraint $c \in \text{FCT} \cup \text{FCS}$. Using this notation, the master problem is written as:

$$\text{Maximize} \quad - \sum_{k \in K} \sum_{p \in \Omega^k} c_p \theta_p + \sum_{t \in T_q} r_t Y_t \quad (2.22)$$

subject to:

$$\sum_{k \in K} \sum_{p \in \Omega^k} g_{ap} \theta_p \geq \nu_a, \quad \forall a \in A \quad (2.23)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} g_{ap} \theta_p \leq \rho_a, \quad \forall a \in A \quad (2.24)$$

$$\sum_{k \in K} \sum_{p \in \Omega^k} h_{tp} \theta_p - Y_t \geq n_t, \quad \forall t \in T_q \quad (2.25)$$

$$\sum_{k \in K} i_{wp} \theta_p = 1, \quad \forall w \in \text{ICS} \cup \text{ICT} \quad (2.26)$$

$$\sum_{k \in K} f_{cp} \theta_p = 1, \quad \forall c \in \text{FCT} \cup \text{FCS} \quad (2.27)$$

$$\theta_p \text{ integer}, \quad \forall k \in K, \forall p \in \Omega^k \quad (2.28)$$

$$Y_t \in [0, q_t], \quad \forall t \in T_q. \quad (2.29)$$

This master problem is equivalent to (2.10)–(2.18), (2.20) and (2.21). It can be used to compute a lower bound for the current branch-and-bound node by relaxing the integrality constraints on the θ_p and Y_t variables. This linear relaxation is solved by a column generation procedure as follows. Starting with an empty subset of path variables in the relaxed master problem, this restricted master problem is solved by the primal simplex algorithm. The resulting dual variable values are then transferred to the objective function of the subproblems. In turn, these are solved by dynamic programming to identify the paths of least reduced cost. If paths with a negative reduced cost are found, the corresponding path variables are added to current subset

of path variables and the restricted master problem is solved again. If no such paths exist, the last primal solution of the restricted master problem is an optimal solution for the current linear relaxation and its value provides a lower bound for the current branch-and-bound node.

As mentioned above, integer solutions are found by exploring a branch-and-bound tree using a heuristic variant of the depth-first strategy that does not allow backtracking. At any given node of the tree, two types of branching decisions can be made. First, when the solution of a linear relaxation violates one or several relaxed constraints (2.19), a branching decision is imposed on one of these constraints. Let $e \in E_{\text{tail}}$ be the arc associated with this constraint. If $X_e < \sum_{e' \in E_e^{\text{clash}}} X_{e'}$, then arc e is removed from its network. Otherwise, all arcs in E_e^{clash} are removed. Second, when all relaxed constraints (2.19) are satisfied by a fractional solution of a linear relaxation, one or several path variable with fractional values are rounded up to the nearest integer. Such decisions are imposed by modifying the master problem right-hand side values and, possibly, removing some arcs and master problem constraints that have become useless (for instance, when the fixed path included an *init* arc).

2.5.2 Computing Car Positions

Solving formulation (2.10)–(2.21) produces car cycles for which a feasible position assignment exists. Such an assignment has yet to be computed. This section provides the main ideas of the proposed car positioning algorithm. The reader is referred to LINGAYA et al. (2000) for a more detailed description of it.

In the algorithm, the locomotive cycles are treated independently. For each cycle $r \in R$, the first train t_1 in cycle r operating within the optimization horizon is considered, and a list of ordered cars assigned to t_1 , but not switched in the consist

before covering t_1 , is built. Such a list of ordered cars is called a *label*. Note that either r starts before the horizon, in which case the list of cars is totally determined by the initial conditions, or r starts within the horizon, yielding an empty list. In both cases, the list is thus unique and easily constructed. This initial label is pushed onto a stack.

Trains in r are then processed sequentially starting with t_1 . For each train t in r , three steps are performed. The first step consists of building every possible label for t . This is achieved by looping on all cars c to be switched in the consist before t and on all labels l found on the aforementioned stack. Each pair (c, l) generates a series of new labels, each new label arising from a different position where c can be switched in l . All labels l used to generate the new labels are removed from the stack and set aside, while the new ones are placed on the stack and retain the label l from which they originate. The second step filters out of the stack the newly generated labels incompatible with the switching out schedule after t . A label l is deemed incompatible when a car c is scheduled to be switched out from the tail end of the consist while another car c' , that does not need to be switched out, is found behind c in l . The third step creates new labels by removing from the labels on the stack every car switched out of the consist after t . The resulting labels are then pushed on the reinitialized stack, omitting duplicate entries.

Once these three steps are accomplished for a train t in cycle r , the stack holds all feasible labels compatible with the scheduled switching activities along r up to t . Hence, when t corresponds to the last train in r operating within the horizon, feasible positions for the cars assigned to all trains in r can be found by choosing a label on the stack and recursively retrieving the labels from which it was derived. Finally, note that the final stack for each locomotive cycle $r \in R$ is never empty when constraints (2.19) are satisfied, or equivalently, feasible car positions can always be found in this case.

2.6 Extensions

This section presents several extensions to the basic OCAP that were addressed for VIA, and explains, for the most important ones, how the formulation (2.10)–(2.21) can be adapted to handle them. When necessary, we also describe how the solution methodology must be modified. For the sake of clarity, these extensions are discussed independently although their combination is relatively straightforward.

2.6.1 Multiple Service Classes

In general, railways offer different classes of service (for instance, business and economy) to their customers, each class yielding a different revenue per passenger. In order to avoid mixing passengers of different classes, cars are often dedicated to a single specific class. In this case, passenger demands are computed by service class, which are then converted into demands per car class. Requests are also formulated per car class.

In order to take multiple service classes into consideration, the following notation is required. Let J , indexed by j , be the set of service classes and partition W into $|J|$ disjoint subsets W^j of cars dedicated to service class $j \in J$. Next, denote by K^j the subset of commodities (w, d) such that $w \in W^j$, $j \in J$. Finally, a superscript j is added on the following parameters, sets and variables to distinguish the service class: r_t^j , q_t^j , n_t^j , ν_a^j , T_q^j , and Y_t^j . Using this notation, relations (2.10), (2.11), (2.13) and (2.21) are replaced in the tractable model (2.10)–(2.21) by

$$\text{Maximize} \quad - \sum_{k \in K} \sum_{e \in E^k} c_e X_e + \sum_{j \in J} \sum_{t \in T_q^j} r_t^j Y_t^j \quad (2.30)$$

$$\sum_{k \in K^j} \sum_{e \in E^{k,a}} X_e \geq \nu_a^j, \quad \forall a \in A, \forall j \in J \quad (2.31)$$

$$\sum_{k \in K^j} \sum_{e \in E^{k,a(t)}} X_e - Y_t^j \geq n_t^j, \quad \forall t \in T_q^j, \forall j \in J \quad (2.32)$$

$$Y_t^j \in [0, q_t^j], \text{ integer}, \quad \forall t \in T_q^j, \quad (2.33)$$

respectively. Final conditions imposed through constraints (2.15), also need to be adapted by specifying the service class of each required car. The solution methodology described in Section 2.5 is directly applicable to this modified formulation.

2.6.2 Car Availability

In the basic OCAP, we considered an unlimited fleet of cars to fulfill car demands. Unless it is possible to rent a large number of cars from an outside source, this assumption does not hold in practice since every railway owns a limited number of cars. Therefore, the real-life OCAP also includes car availability constraints. Furthermore, given that the demand is larger over the weekends, railways often decide to increase their maintenance activities (other than the regular maintenance checks) at the beginning of the week to have as many cars as possible available for the weekends. This creates a car availability that varies on a daily basis.

To take daily car availability into account, the following constraints can be added to formulation (2.10)–(2.21):

$$\sum_{s \in S} \sum_{k \in K} \sum_{e \in O^{k, \text{stop}_s^d}} X_e \leq u_d, \quad \forall d \in [d_0, d_{\max}], \quad (2.34)$$

where u_d corresponds to the number of cars available on day $d \in [d_0, d_{\max}]$. These constraints simply count the number of cars operating outside the base every day. Again, the basic solution methodology is directly applicable to the modified model.

2.6.3 Maximum Mileage between Maintenance Checks

In the basic OCAP, regular maintenance checks must be performed at least once every m days on each unit of equipment. However, these inspections may also be scheduled according to the mileage traveled by the cars. Often, the rule is a combination of the number of days and the mileage traveled since the last check; for instance, each car must be maintained at least once every m days or at every μ miles, whichever limit is reached first.

The upper limit on mileage can be modeled using a mileage resource that accumulates the number of miles traveled along a cycle. Let m_e be the number of miles traveled along edge $e \in E$, and let M_v^k be the total number of miles accumulated by the car associated with commodity $k \in K$ from the source node src^k to node $v \in V^k$. Obviously, the value of such a variable must fall within the interval $[0, \mu]$; otherwise, the corresponding cycle violates the constraint on the number of miles between two maintenance checks.

This extension to the basic OCAP can be dealt with by adding the following constraints to formulation (2.10)–(2.21):

$$X_e(M_{o(e)}^k + m_e - M_{d(e)}^k) \leq 0, \quad \forall k \in K, \forall e \in E^k \quad (2.35)$$

$$M_v^k \in [0, \mu], \quad \forall k \in K, \forall v \in V, \quad (2.36)$$

where $o(e)$ and $d(e)$ denote the origin and the destination node of edge $e \in E$, respectively. Constraints (2.35)–(2.36), together with constraints (2.16)–(2.18) and (2.20), define the domain of a shortest path problem with resource constraints (see, e.g., DESROSIERS et al., 1995). This new formulation is solved using the decomposition approach proposed in Section 2.5. In this case, constraints (2.35)–(2.36) are considered in the column generation subproblem which becomes a shortest path problem with resource constraints. Such a subproblem can be solved by the dynamic programming algorithm of DESROCHERS and SOUMIS (1988).

2.6.4 Short Wye

There exist two types of connections between consecutive trains on which a unit of equipment is assigned: turn-arounds and run-thrus. These are illustrated in Figure 2.2. A consist performs a *turn-around* if its orientation needs to be reversed during the connection in order to leave in the opposite direction for the outbound train. Otherwise, the connection is said to be a *run-thru*. Obviously, a turn-around connection requires more time than a run-thru connection. Moreover, it requires a special Y-shaped track, called a *wye*, where the consist, starting from the stem, will move forward on a branch of the wye, then backward from this branch to the other, and, finally, forward toward the stem. In general, the whole consist is turned around at once and the cars remain in the same relative positions behind the locomotive for the outbound train. However, in some stations, the wye is not long enough (or a turn-table is used instead of a wye) and only the locomotive can be turned around. In this situation, the order of the cars behind the locomotive is reversed for the outbound train. Hence, constraints (2.5) do not hold anymore when the connection between t and t' is a turn-around on a short wye. Even modifying the definition of $C_{t,t'}(X)$ would not be sufficient to derive a tractable formulation since Proposition 1 would not be valid anymore. In fact, it does not seem possible to provide a complete mathematical program that does not directly use car positioning variables since, depending on its position, a car switched in the body of a consist can be found at its tail end after a turn-around in a station with a short wye.

The solution methodology proposed for this case and described below is based on the formulation (2.10)–(2.21) where the definition of the clashing arcs, used to define the sets E_e^{clash} in constraints (2.19), is changed. Indeed, an arc $e' \in E$ is now said to be clashing with an arc $e \in E_{\text{tail}}$ if it verifies all conditions stated in Definition 2 and if an even number of turn-arounds on a short wye (or none) is performed by the

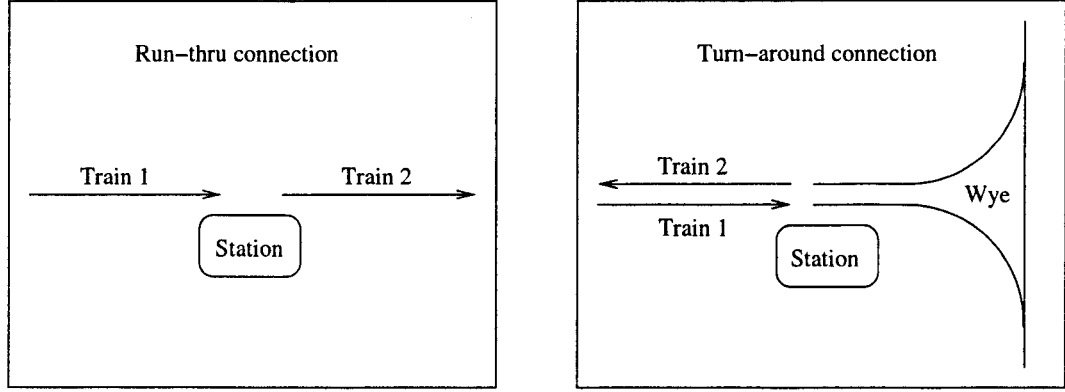


Figure 2.2: Run-thru and turn-around connections

locomotive between the first train of the atom associated with e' and the last train of the atom associated with e . When the number of turn-arounds on a short wye is odd, the car associated with arc e' is switched in at the tail end, but is positioned in the body when it is time to switch out the car associated with arc e , thus creating no conflict. However, in this case, if the first car was switched in the body of the consist, it could be positioned at its tail end when the second car is supposed to be switched from the tail end. Obviously, this creates a conflict that cannot be detected without considering precise car positions. Therefore, the proposed model, (2.10)–(2.21) with a modified definition for the sets E_e^{clash} , is a relaxation of the problem with short wye tracks and its solution does not necessarily ensure that all cars can be positioned on the trains without switching conflicts.

In order to solve this extension of the problem, we propose the following two-step iterative procedure. In the first step, the relaxed formulation (2.10)–(2.21) is solved using the method described in Section 2.5.1 to yield car cycles. In the second step, a slightly modified version of the positioning algorithm described in Section 2.5.2 is used to find car positions that do not yield switching conflicts. If such positions exist, the positioning algorithm finds them and the whole solution process stops. Otherwise,

it identifies a switching conflict derived from the proposed car cycles and suggests a conflicting arc to eliminate. In this case, the solution process returns to the first step to find alternate car cycles. The modifications required for the positioning algorithm consist of reversing the positions of the cars when a turn-around on a short wye is performed by a locomotive, and identifying conflicts which can only occur when switching out cars. Such a conflict is found when all labels preceding the move of switching out a car from the tail assign an in-body position for that car. In this case, the arc associated with this tail switching is the one suggested for removal.

2.6.5 Car Orientation

Another constraint that often needs to be addressed in practice concerns the orientation of the passenger cars. A car is oriented forward if its vestibule (entrance) is on the locomotive side; otherwise, it has a backward orientation. To ensure that all vestibules are equally spaced along the train, all cars assigned to the same train should have the same orientation. Since the cars assigned to the first train of a locomotive cycle exit the maintenance center with the same orientation (usually, forward) after an inspection, each of these cars keeps the same orientation all along this cycle unless it is switched out of the consist. In this case, the orientation of the switched cars might be backward with respect to trains departing from the connecting station in a direction opposite to the direction of the train from which the cars were switched out. Therefore, these cars cannot connect to these outbound trains unless they are turned around by a local locomotive assigned to that station. Note that not all stations have local locomotives at their disposal.

This extension can be handled simply by modifying network G as follows. Two series of D -type and A -type nodes are created for each station $s \in S$. Assuming that

all cars leave the maintenance center with a forward orientation and, without loss of generality, that the tracks in station s are oriented east-west, a series of nodes contains all A -type and D -type nodes of station s corresponding to arrivals and departures of westbound trains, while the other series contains the remaining A -type and D -type nodes of station s . These series of nodes are such that all cars switched out of the last train of the atom associated with an A -type node of a series can connect without turning around to the first train of the atom associated with any subsequent D -type node of the same series. If the cars assigned to the first train of a locomotive cycle leave the maintenance center with a backward orientation, then all D -type and A -type nodes created from the atoms of this cycle must be placed in the opposite series of nodes. Finally, at stations where a local locomotive is available, an arc from each A -type node to the first appropriate D -type node of the opposite series is added to represent the possibility of turning around a car using a local locomotive. The appropriate D -type node is the one associated with the earliest outbound train to which a car switched out of the inbound train associated with the A -type node can connect given the additional time required to turn it around. Note that such an arc can bear a positive (penalty) cost. The basic solution methodology does not require any modification.

2.6.6 Trains with Multiple Locomotives

One important assumption made for the basic OCAP is that exactly one locomotive must be assigned to each train. For most railways, this assumption is too restrictive and the definition of the OCAP should allow for more than one locomotive per train. When two locomotives are assigned to a train, the following two consist configurations can be considered: i) a forward locomotive followed by a set of cars assigned to it, a second forward locomotive, and another set of cars assigned to it; and ii) a forward

locomotive followed by a set of cars assigned to it, another set of cars assigned to the second locomotive, and the second locomotive in a backward orientation. In both cases, all sets of cars might be empty and the cars in a set are either all forward or all backward.

Given that each block formed by a locomotive and the cars assigned to it can move independently, we consider that, among the cars assigned to a locomotive, the furthest from the locomotive is considered at the tail end. For instance, the first car of the second set in configuration ii) is at the tail end of the second part of the consist. Obviously, from the whole consist viewpoint, this car is in its body and requires the minimal in-body switching time when switched during a connection between two trains to which the same two locomotives are assigned. The configurations for the case with more than two locomotives assigned to a train, which can be deduced by extending the above configurations, lead to similar difficulties. Note that locomotive orientations and their relative positions when more than one locomotive are assigned to a train are given as input.

This extension is treated using a heuristic procedure as follows. On the one hand, model (2.10)–(2.21) is not modified. As before, all locomotive cycles are treated independently in the construction of network G . Therefore, all tail end cars that need to be switched require the minimal tail switching time even if they are positioned in the middle of a multiple-locomotive consist. On the other hand, the solution methodology is slightly modified to account for this imprecision. As proposed for the short wye extension, the cycle computing algorithm of Section 2.5.1 and the car positioning algorithm of Section 2.5.2 are embedded in an iterative process. First, the cycle computing algorithm is run to determine car cycles. Then, the car positioning algorithm looks for valid car positions without any switching conflict (that may arise only for a tail end car positioned in the middle of a multiple-locomotive consist). If such positions exist, it finds them. Otherwise, it identifies the switching conflict and

proposes a tail switching arc to remove in order to avoid this conflict. The process then returns to the cycle computing algorithm.

2.6.7 Other Extensions

Several other extensions to the basic OCAP can be considered in practice and handled by adapting model (2.10)–(2.21) and the proposed solution methodology. Here, we briefly list some extensions that are integrated in the software system implemented at VIA.

- (a) When multiple car classes are considered, it is possible to perform substitutions between classes. For instance, if there is a shortage of economic class cars, it is always possible to use business class cars instead. The opposite is also conceivable, but with a penalty.
- (b) Regular maintenance checks can be performed during daytime if the maintained cars remain at the maintenance center for a sufficient period time.
- (c) When cars are switched out of a consist in a station, they are left on side tracks. These tracks have a limited capacity that needs to be taken into account.
- (d) When more than one car of the same service class are requested for the same train, the anticipated revenues of these cars might be different. Indeed, the demand forecast for the last of these cars might be less than the passenger capacity of a car.
- (e) Switching often incurs additional costs that should be taken into account. Moreover, the switching activities can be restricted during certain periods of time in certain stations due to a lack of personnel.

- (f) When multiple car classes are considered, cars of a same class should have consecutive positions on all trains in order to avoid, as much as possible, mixing passengers of different service classes. Furthermore, all non-deadhead cars should be consecutive.

2.7 Computational Experiments

The methodology described in this paper was developed with the goal of providing VIA with a computer system for the evaluation of car requests and the update of planned cycles. We now describe the data used in the simulation experiments and give a summary of the results obtained.

2.7.1 Description of Data

The data used in the simulation experiments concern the trains operated by VIA in the Québec-Windsor corridor. These data are described in detail by CORDEAU et al. (2001a) and we provide only a brief description here. VIA operates approximately 350 trains every week on a physical network composed of nine end-of-train stations. The company uses two types of locomotives, two types of first-class (business) cars and two types of second-class (economy) cars that can be combined in three different ways to form train consists with different operating speeds. The complete fleet is composed of over 130 units of equipment. For each train, the demand for first-class cars is normally 0 or 1 whereas that for second-class cars varies between 2 and 8. The pulling capacity of a locomotive varies between 5 and 8 cars. Finally, a single maintenance center, located in Montreal, is used for the weekly inspection of all locomotives and cars.

To evaluate the performance of our solution methodology, test instances were generated from the weekly schedule used in a particular season. For this weekly schedule, we first determined locomotive and car cycles using the solution approach introduced by CORDEAU et al. (2001a) for the first phase of the planning process. Then, we created a large number of instances by randomly generating demand revisions for first-class and second-class cars.

For each train $t \in T$ and each service class $j \in J$, let n_t^j denote the demand for cars of class j on train t according to the season schedule. A new demand \tilde{n}_t^j is then associated to the train and service class by generating a random number from a normal distribution of mean $n_t^j(1 + \delta_t)$ and standard deviation $n_t^j\sigma$, and rounding the result to the nearest integer.

Two scenarios were considered to generate demand revisions. In the first scenario (A), we set $\delta_t = 0$ and thus generated revised demands centered around the original demand for each train and each service class. The values of σ were then calibrated so as to represent actual operating conditions as much as possible. In this scenario, the value of σ was set to 0.0973 for second-class cars so as to obtain a 20% probability of having a different demand when the original demand is four cars. For first-class cars, the value of σ was set to 0.1942 so as to obtain a 1% probability of having a different demand when the original demand is one car. In the second scenario (B), we simulated the effect of having a major week-end event in one of the most important stations of the network, Toronto. To do this, the value of δ_t was set to 0.1 for all trains going to Toronto on Friday or Saturday and all trains leaving from Toronto on the next Sunday or Monday (with $\delta_t = 0$ for all other trains). The values of σ were the same as in the first scenario, thus yielding higher probabilities of upward revisions.

For each scenario, we then considered scheduling horizons ranging from 1 to 7 days. Finally, for each scenario and each scheduling horizon, we obtained ten test instances

by generating different sets of demand revisions with the procedure described above. A total of 140 instances were thus generated.

For each train t and each service class j such that $\tilde{n}_j^t > n_j^t$, a number of requests equal to $\tilde{n}_j^t - n_j^t$ must be generated. In the case of an upward demand revision, non-increasing expected profits are randomly generated for each request using a uniform distribution. To reflect the planning process used at VIA, a priority is also assigned to each request for the addition of a car. This priority is chosen randomly in the set $\{1, 2\}$ and reflects the importance of satisfying a particular request. Scaling factors multiplying the expected profit are then used to translate these priorities in the objective function (2.1). As mentioned previously, requests for the removal of cars are treated directly by reducing the value of n_j^t . These decisions are made externally by the marketing department and need not be considered explicitly in the model.

The characteristics of the test instances are summarized in Table 2.1. In this table, column “Trains” indicates the number of trains in the scheduling horizon. Then, columns “Add” and “Remove” report the average number of requests for the addition and removal of cars, respectively. Each line in the table corresponds to a different combination of scenario and scheduling horizon. For example, in the 6-day instances for scenario A, there were 296 trains. On average, these instances had 1.2 and 0.9 requests for the addition and removal of first-class cars, and 7.4 and 8.3 requests for the addition and removal of second-class cars, respectively.

2.7.2 Computational Results

Each of the 140 instances was solved using the solution methodology explained in Section 2.5. To reduce the size of the Dantzig-Wolfe master problem, locomotive

Table 2.1: Number of trains and average number of requests in test instances

Instance	Trains	First class		Second class	
		Add	Remove	Add	Remove
A-1	52	0.2	0.2	0.5	0.8
A-2	105	0.4	0.3	2.1	1.3
A-3	147	0.5	1.0	3.3	2.7
A-4	186	0.7	0.7	5.0	4.4
A-5	241	1.1	1.7	5.9	6.7
A-6	296	1.2	0.9	7.4	8.3
A-7	348	0.8	1.8	9.0	8.4
B-1	52	0.4	0.2	0.6	0.8
B-2	105	0.7	0.3	4.5	1.6
B-3	147	0.5	0.5	9.5	2.8
B-4	186	0.6	1.0	13.2	2.3
B-5	241	1.3	1.3	19.6	3.2
B-6	296	1.7	1.1	21.7	4.5
B-7	348	2.2	0.9	23.1	5.1

pulling capacity constraints (2.12) as well as track capacity constraints (see Section 2.6.7) are first relaxed and added dynamically to the problem when violated. As explained in Section 2.5.1, positioning constraints (2.19) are also introduced dynamically when they are violated after solving the linear relaxation at a given node of the branch-and-bound tree.

The computational results obtained are summarized in Table 2.2. Here, column “Dynamic constraints” indicates the total number of locomotive pulling capacity, track capacity and positioning constraints that were added during the solution process. Because of short wyes, several iterations sometimes had to be performed before the car positioning algorithm could find a feasible solution (see Section 2.6.4). This information is reported in column “Major iterations”. Then, column “Requests filled” indicates the percentage of requests for the addition of cars that were successfully filled. Finally, column “CPU time” reports the total running time of the algorithm on a Sun Ultra-10 (440 MHz) computer. All results represent the average over the ten instances of the same scenario-horizon combination.

The results show that the approach yields feasible solution in very short computing times. For the instances involving a single day, the CPU time rarely exceeded 1 minute and was below 30 seconds in most cases. Computing times also appear to grow linearly with the length of the scheduling horizon. For the largest instances involving 7 days and an average of more than 30 requests (instance set B-7), the CPU time rarely exceeded 15 minutes. Because scaling factors are used to take request priorities into account, evaluating the quality of these solutions precisely is difficult. Indeed, the objective function (2.1), which includes both positive and negative terms, may become very small or even negative on some instances. Computing the relative integrality gap is therefore meaningless. However, it is worth mentioning that imposing integrality constraints was in general very easy and the absolute integrality gaps were extremely small. The percentage of requests for car additions that were

Table 2.2: Average computational results

Instance	Dynamic constraints	Major iterations	Requests filled (%)	CPU time (minutes)
A-1	0.0	4.8	100	0.39
A-2	2.3	7.3	88	1.09
A-3	7.3	6.9	71	1.77
A-4	17.0	6.0	74	6.77
A-5	29.8	7.5	79	4.11
A-6	31.9	9.1	88	8.86
A-7	28.3	4.9	92	4.37
B-1	0.0	4.8	70	0.39
B-2	11.5	8.4	94	1.44
B-3	5.0	7.1	54	2.85
B-4	29.2	7.0	67	3.82
B-5	33.8	7.1	75	4.75
B-6	34.9	9.2	84	11.02
B-7	32.3	5.8	75	11.19

successfully filled varies between 54% and 100%. This percentage varies with the number of requests but also with the importance of the original demand. When the original demand can only be satisfied by using all available cars, very few addition requests can be satisfied unless removal requests are also formulated. A request for the addition of a car can also fail to be fulfilled if the revenue associated with it is lower than the cost of moving a car to the required location before and after fulfilling the request. Finally, the number of major iterations required seems to be independent of problem size while the number of dynamic constraints generated increases linearly with the length of the scheduling horizon.

It is worth mentioning that requests that cannot be fulfilled when solving the OCAP will normally be rejected by the operations personnel unless additional cars can be borrowed from another part of the network or another railroad. When a request is rejected, marketing personnel will be notified that additional capacity could not be added on the specified train. Finally, if requests to add cars on a given train are regularly rejected, the company will adjust its demand forecast on that train for the next season.

2.8 Conclusion

This paper introduces a formulation and a solution approach for a complex operational problem that arises in the planning process of most passenger railways that rely on breakable consists and car switching. In practice, the problem must often be solved repeatedly to take schedule and demand revisions into account. To obtain a powerful decision analysis tool, fast computing times are of utmost importance. The computational results obtained with our approach show that it is well suited to be used in an interactive fashion. A software system based on this approach is now implemented at VIA Rail Canada and is being evaluated by the company.

Acknowledgments

We wish to thank Mrs. Francine Hébert, Mr. Alain Vigeant and Mrs. Natali Gagnon of VIA Rail for their valuable collaboration. This work was supported by the Québec Government (Fonds pour la Formation de Chercheurs et l'Aide à la Recherche) and by a joint research project of the Natural Sciences and Engineering Research Council of Canada and AD OPT Technologies Inc.

2.9 Bibliography

N. BEN-KHEDER, J. KINTANAR, C. QUEILLE AND W. K. STRIPLING. "Decision support scheduling systems for SNCF". Presented at the INFORMS Fall Meeting, Dallas, 1997.

M. R. BUSSIECK, T. WINTER AND U. T. ZIMMERMANN. "Discrete optimization in public rail transport". *Mathematical Programming* 79, 415–444, 1997.

J.-F. CORDEAU, G. DESAULNIERS, N. LINGAYA, F. SOUMIS AND J. DESROSIERS. "Simultaneous locomotive and car assignment at VIA Rail Canada". *Transportation Research Part B: Methodological*, 35, 767–787, 2001a.

J.-F. CORDEAU, F. SOUMIS AND J. DESROSIERS. "A Benders decomposition approach for the locomotive and car assignment problem". *Transportation Science* 34, 133–149, 2000.

J.-F. CORDEAU, F. SOUMIS AND J. DESROSIERS. "Simultaneous assignment of locomotives and cars to passenger trains". *Operations Research*, 49, 531–548, 2001b.

J.-F. CORDEAU, P. TOTH AND D. VIGO. "A survey of optimization models for train routing and scheduling". *Transportation Science* 32, 380–404, 1998.

G. B. DANTZIG AND P. WOLFE. "Decomposition principle for linear programming". *Operations Research* 8, 101–111, 1960.

G. DESAULNIERS, J. DESROSIERS, I. IOACHIM, M.M. SOLOMON, F. SOUMIS AND D. VILLENEUVE. "A unified framework for deterministic time constrained vehicle routing and crew scheduling problems". In T.G. Crainic and G. Laporte, editors, "Fleet Management and Logistics", pages 57–93. Kluwer, Norwell, MA, 1998.

M. DESROCHERS AND F. SOUMIS. "A generalized permanent labelling algorithm for the shortest path problem with time windows". *INFOR* 26, 191–212, 1988.

J. DESROSIERS, Y. DUMAS, M.M. SOLOMON AND F. SOUMIS. "Time constrained routing and scheduling". In M. O. Ball, T. L. Magnanti, C. L. Monma and G.L. Nemhauser, editors, "Network Routing, Handbooks in Operations Research and Management Science, 8", pages 35–139. Elsevier Science, Amsterdam, 1995.

N. LINGAYA, J.-F. CORDEAU, G. DESAULNIERS, J. DESROSIERS AND F. SOUMIS. "Operational car assignment at VIA Rail Canada". Technical Report 2000-55, GERAD, École des Hautes Études Commerciales, Montréal, Canada, 2000.

K. V. RAMANI. "An information system for allocating coach stock on Indian Railways". *Interfaces* 11(3), 44–51, 1981.

K. V. RAMANI AND B. K. MANDAL. "Operational planning of passenger trains in Indian Railways". *Interfaces* 22(5), 39–51, 1992.

K. V. RAMANI AND G. RAGHURAM. "Utilisation of transportation units, an optimal allocation study". *Opsearch* 17(1), 33–40, 1980.

2.10 Appendix

This appendix presents the details of the car positioning algorithm discussed in Section 2.5.2. The description of this algorithm requires the following definitions.

As before, define a *label* l as an ordered list of cars indicating the positioning of the cars in a consist. This consist may be the one assigned to a train or a temporary one after switching out some cars and before switching in other cars. An empty list of cars, referred to as `(nil)`, constitutes a valid label. Usually, a label also contains a pointer to the label from which it was created, but, for the sake of conciseness, this pointer is omitted here. Labels are grouped into the sets L and L' . Finally, for a train t , let $W_t^-(X)$ and $W_t^+(X)$ be the sets of cars switched in before the departure, and switched out upon the arrival, respectively.

For a locomotive cycle $r \in R$, the operator `FirstTrain(r)` returns the first train of r that operates in the horizon. For a locomotive cycle $r \in R$, the boolean operator `Init(r)` returns `true` if and only if r begins before the horizon, thus yielding initial conditions. For a train $t \in T$ and a car $w \in W_t^-(X)$, the boolean operator `Tail-(t, w)` returns `true` if and only if w is switched in tail before the departure of t . For a train $t \in T$ and a label $l \in L$, the operator `Last(t, l)` returns the last car in l that is not switched in at the departure of t ; `nil` is returned if no such car exists. For a label $l \in L$ and a pair of cars $w_1 \in W$ and $w_2 \in W$, the operator `Insert(l, w_1, w_2)` returns the label l' obtained by copying l and inserting w_2 after w_1 in the new list; if w_1 is `nil`, then w_2 is inserted at the beginning of the new list. For a label $l \in L$ and a car $w \in W$, the operator `NextCar(l, w)` returns the first car in l if $w = \text{nil}$; the car following w in l if w is not the last car in l ; and `nil` if w is the last car in l . For a train $t \in T$, a label $l \in L$ and a car $w \in W_t^+(X)$, the boolean operator `Legal(t, l, w)` returns `false` if and only if w is switched out from the tail at the

arrival of t and there exists another car in l , positioned after w , that remains in the consist. For a label $l \in L$ and a car $w \in W$, the operator **Remove**(l, w) returns the label l' obtained by copying l and removing w from the new list. For a label set L , the function **Reduce**(L) eliminates from L all redundant labels, keeping one copy of each different list. For a locomotive cycle $r \in R$ and a train $t \in T$ in that cycle, the operator **NextTrain**(r, t) returns the train following t in r if t is not the last train in r operating in the horizon; otherwise, **nil** is returned. Finally, for a locomotive cycle $r \in R$ and the last label set L computed for r , the function **Position**(r, L) takes any label $l \in L$ to position the cars on the trains in r that operate in the horizon. Using the (omitted) pointer to the label from which l was created, it retrieves the label associated to each train in that cycle.

The pseudo-code of the algorithm is given on the next page. In short, the main parts of the algorithm are as follows. For each locomotive cycle $r \in R$, the label set L is initialized in steps 3–6 as containing the ordered list of cars behind the locomotive at the beginning of the horizon if r begins before it, or as containing the **nil** label otherwise. Then, for each train $t \in T$ along r that runs during the horizon, the following operations are performed. First, in steps 8–19, the cars $w \in W_t^-(X)$ switched in before the departure of t are added in every feasible position according to whether or not the car is switched at the consist tail end. Each possibility creates a new label from which new labels will be created when treating the next train along the cycle r . Second, in steps 20–26, the cars $w \in W_t^+(X)$ switched out at the arrival of t are removed from the labels to create new labels containing only the remaining cars. When a car need to be switched out from the consist tail end, it is verified that it is located there before creating the new label. Otherwise, the new label is not created.

Algorithm 1 Car Positioning Algorithm

```

1: for all  $r \in R$  do
2:    $t = \text{FirstTrain}(r)$ 
3:   if  $\text{Init}(r)$  then
4:      $L = \{(w_1, \dots, w_n)\}$ , where  $w_i$  is the car in position  $i$  behind the locomotive
       associated with  $r$  at the beginning of the horizon.
5:   else
6:      $L = \{(\text{nil})\}$ 
7:   repeat
8:     for all  $w \in W_t^-(X)$  do
9:        $L' = \emptyset$ 
10:      for all  $l \in L$  do
11:        if  $\text{Tail}^-(t, w)$  then
12:           $w_{\text{last}} = \text{Last}(t, l)$ 
13:        else
14:           $w_{\text{last}} = \text{nil}$ 
15:        repeat
16:           $L' = L' \cup \{\text{Insert}(l, w_{\text{last}}, w)\}$ 
17:           $w_{\text{last}} = \text{NextCar}(l, w_{\text{last}})$ 
18:        until  $w_{\text{last}} = \text{nil}$ 
19:       $L = L'$ 
20:      for all  $w \in W_t^+(X)$  do
21:         $L' = \emptyset$ 
22:        for all  $l \in L$  do
23:          if  $\text{Legal}(t, l, w)$  then
24:             $L' = L' \cup \{\text{Remove}(l, w)\}$ 
25:           $\text{Reduce}(L')$ 
26:         $L = L'$ 
27:       $t = \text{NextTrain}(r, t)$ 
28:    until  $t = \text{nil}$ 
29:     $\text{Position}(r, L)$ 

```

Chapitre 3

Extensions et synthèse

L'article que nous avons présenté dans le chapitre précédent décrit les extensions apportées au modèle de base dans le but de répondre aux exigences opérationnelles de VIA. Cependant, afin de garder une longueur raisonnable, six d'entre elles n'ont pu qu'être mentionnées sans que leur modélisation n'ait pu être discutée. Nous nous proposons ici de les développer brièvement avant de poursuivre avec une description de l'implantation réalisée chez VIA et d'une brève synthèse du projet.

3.1 Extensions

3.1.1 Substitution de classe

Il arrive, lorsqu'on considère plus d'une classe de wagons, qu'une pénurie de wagons d'une classe soit accompagnée d'un surplus de wagons d'une autre classe en un point du réseau. Dans ce cas, il s'avère intéressant de procéder à une substitution de classe plutôt que d'avoir recours à de coûteuses mises en place pour rapatrier un wagon adéquat ou même de refuser une requête. S'il est clair qu'offrir un wagon de classe affaire à un voyageur de classe économique ne lui crée pas de désagrément, l'inverse

n'est pas toujours souhaitable. On introduit donc une notion de coût de substitution pour pondérer cette notion d'indésirabilité. Cette extension est facilement réalisable en ajoutant des arcs supplémentaires au graphe G : il suffit, pour chaque utilisation potentielle d'un wagon de classe $j_1 \in J$ en classe $j_2 \in J$, de dupliquer l'arc représentant la couverture d'une séquence de trains par ce wagon de classe j_1 , de modifier la contribution aux contraintes de couverture en visant plutôt les contraintes concernant la couverture par la classe j_2 , et d'ajouter au coût de cet arc le coût lié à la substitution.

3.1.2 Entretien de jour

Il est parfois possible d'effectuer l'entretien des wagons durant le jour pourvu que l'on dispose du temps nécessaire pour le faire. Étant donné que les routes de locomotive sont connues, on peut identifier les opportunités d'entretien de jour qu'offrent ces routes. Une telle opportunité est reconnue lorsque

1. la route d'une locomotive contient un train $t \in T$ qui arrive au centre d'entretien à l'heure $\text{arr}(t)$;
2. le temps de connexion entre ce train et le train suivant le long de cette route est suffisant pour décrocher un wagon ;
3. il existe un train quittant le centre d'entretien le même jour après l'heure $\text{arr}(t) + \Delta$ où Δ représente le temps nécessaire pour un entretien de jour.

La dernière condition est imposée afin de ne pas confondre les entretiens de jour et ceux de nuit.

Pour chacune des opportunités d'entretien de jour identifiées, une contrainte de conservation de flot et deux arcs sont ajoutés. Le premier arc relie le noeud d'arrivée

du train concerné au noeud puits de la journée d'arrivée et contribue positivement d'une unité à la contrainte de conservation de flot. Le second arc relie le noeud source du même jour au noeud de départ du premier train quittant après l'heure $\text{arr}(t) + \Delta$ et contribue négativement d'une unité à la même contrainte. Par conséquent, si le premier arc est utilisé par un wagon, la contrainte de conservation de flot imposera que le second arc soit aussi utilisé. Dans ce cas, le wagon utilisant ce second arc retourne dans le réseau après un entretien et peut être utilisé pour un maximum de m jours avant le prochain entretien.

3.1.3 Capacité d'entreposage des wagons dans les stations intermédiaires

Si certaines stations intermédiaires permettent le couplage et découplage de wagons, il reste qu'elles peuvent être limitées dans leur possibilité d'entreposer des wagons durant des journées entières. Pour traiter cette limitation, il faut apporter les modifications suivantes au réseau G afin de pouvoir répertorier les wagons qui passent des journées entières dans les stations. D'abord, les arcs de type *wait* reliant un noeud SOD_d^s d'une station $s \in S$ au jour $d \in D$ au premier noeud de la station s au jour d sont enlevés empêchant ainsi les wagons inactifs pour la journée d d'emprunter la suite des arcs *wait*. Deuxièmement, chaque arc $(i, j) \in \mathcal{E}$ de type *atom* est dupliqué. La copie d'un tel arc dont le noeud i représente le départ d'un train d'une station $s \in S$ au jour $d \in D$ débute au noeud SOD_d^s et se termine au noeud j . Ces nouveaux arcs sont les seuls qui permettent d'affecter les wagons à leur premier train de la journée. Finalement, pour chaque station $s \in S$ et chaque jour $d \in D$, un arc d'un nouveau type, appelé *idle* et reliant directement les noeuds SOD_d^s et EOD_d^s , est ajouté. Cet arc permet l'entreposage des wagons à la station s au jour d . Par conséquent, un wagon inactif durant toute une journée utilise soit un arc de type *idle* ou un arc de

type *atom* comprenant plus d'un train et une période d'inactivité durant une journée complète.

Pour s'assurer que le nombre de wagons entreposés dans chaque station à chaque jour ne dépasse pas la capacité de la station, il suffit d'ajouter une contrainte par station concernée et par jour en fixant le membre de droite à la capacité maximale de la station, puis d'ajouter une contribution unitaire aux arcs représentant le fait qu'une pièce d'équipement est entreposée à ladite station (i.e., les arcs *idle* et *atom* cités plus haut).

3.1.4 Revenus anticipés différents

Les revenus anticipés lors de l'affectation d'un wagon supplémentaire à un train sont déterminés par un modèle de prévision probabiliste. Il s'ensuit que lorsqu'on anticipe ouvrir plus d'un wagon supplémentaire sur un même train, la probabilité de remplir un wagon diminue avec le nombre de wagons déjà alloués à ce train. Ainsi, le revenu anticipé pour le premier wagon devrait être supérieur à celui du deuxième. Pour modéliser cela, il suffit de remplacer chaque variable Y_t , $t \in T$, par une somme de variables binaires $Y_{i,t}$, $i \in \{1, 2, \dots, q_t\}$, ayant des revenus anticipés respectifs $r_{i,t}$ tels que $i < j \Rightarrow r_{i,t} \geq r_{j,t}$. La maximisation de la fonction objectif assure que, pour $i < j$, $Y_{j,t}$ ne peut prendre la valeur 1 que si $Y_{i,t} = 1$.

3.1.5 Coût variable et restrictions sur la modification de convoi

Le couplage/découplage d'un wagon requiert une main-d'oeuvre dont le coût et la disponibilité peuvent varier d'une station à l'autre, voire d'un moment de la journée

à l'autre. On peut tenir compte de ces variations en modulant le coût des arcs modélisant un couplage ou découplage en fonction du lieu et du moment où ces opérations sont effectuées. On peut même enlever ces arcs du graphe G lorsque ces opérations sont prohibées aux moments et aux stations considérés.

3.1.6 Contiguïté des wagons de même classe

Lorsque l'on considère des convois constitués de wagons de différentes classes, il est souhaitable que les wagons d'une même classe soient tous regroupés en un seul bloc contigu de façon à pouvoir limiter les déplacements des passagers entre les différentes classes de wagons. De plus, dans la mesure du possible, les wagons tractés à vide à des fins de repositionnement ne doivent pas séparer des wagons en service. Ce sont là des contraintes de positionnement que l'on reporte donc au niveau de l'algorithme de positionnement des wagons. À chaque situation indésirable est associée une forte pénalité. La solution ayant le moins de violation est alors retenue et une décision modifiant le graphe G de façon à éliminer une des violations est alors appliquée dans l'espoir de trouver une meilleure solution. La sévérité de ces pénalités permet de pondérer l'indésirabilité de ces violations face aux coûts d'exploitation.

3.2 Synthèse

Comme nous l'avons vu plus tôt, le projet décrit ici est le troisième volet d'une implantation en trois phases visant à traiter l'affectation des locomotives et wagons aux trains exploités par VIA dans l'Est du Canada. Les trois modules baptisés ECM, DPM et DOM suivent toutes la même architecture : une interface graphique (GUI)

tournant sur un PC permet aux usagers d'importer des données en format texte d'un système externe vers une base de données interne, de les visualiser et manipuler. Cette interface donne accès à des paramètres et permet, à travers la notion de scénario, de procéder à des études de cas. Elle pilote un optimiseur résidant sur une station UNIX avec qui elle échange des données par le biais de fichiers textes et demande des services par le biais de la fonctionnalité REXEC (remote execution ou exécution à distance). Plusieurs interfaces graphiques peuvent de façon concurrente accéder à la base de données ce qui permet à plusieurs usagers de préparer en parallèle des scénarios. Cependant, une seule optimisation par module peut être exécutée en même temps. Une fois trouvées, les solutions peuvent aussi être éditées manuellement par un planificateur à partir de l'interface. Avant d'être enregistrées dans le système, les solutions éditées manuellement sont validées et évaluées afin de permettre au planificateur de mieux mesurer les impacts des changements qu'il propose. Lorsque le planificateur obtient une solution qu'il lui convient, l'interface graphique permet de l'exporter en format texte vers un système externe et de produire des rapports qui peuvent être imprimés.

Le système complet comprenant l'interface graphique et les trois modules ECM, DPM et DOM ont été livrés et installés chez VIA au cours des années 1997 à 1999. Toutefois, afin d'être totalement intégré aux opérations de VIA, ce système demande un certain développement informatique de la part du transporteur ferroviaire. En effet, des programmes d'échange de données sont nécessaires pour transférer l'information de et vers leur serveur central à travers les fichiers textes mentionnés plus haut. Or, un gel du développement des programmes d'échange de données avec le système central a été décrété par VIA au milieu du projet afin de mobiliser tout le personnel informatique de VIA sur le bug de l'an 2000. Le développement n'a toujours pas repris par la suite suite à des coupures de personnel et des restrictions budgétaires. Seule l'intégration du module ECM a pu être complétée, l'intégration des deux autres modules étant

toujours en suspens. Par conséquent, bien que les tests réalisés dans ce mémoire démontre un potentiel pour la méthodologie développée, l'utilisation quotidienne de cette méthodologie se fait encore attendre.

Finalement, mentionnons aussi que VIA a récemment fait l'acquisition d'un nouveau type d'équipement (locomotives et wagons) qui s'opère par rame composée d'une locomotive et d'un certain nombre de wagons qui peut varier d'une rame à l'autre. Dans ces rames, le couplage et le découplage de wagons nécessitent des opérations de soudure, ce qui rend ces rames difficilement divisibles en un temps raisonnable. Avec ce nouvel équipement, les couplages et découplages de wagons deviennent inexistantes en mode opérationnel et n'ont pas besoin d'être planifiés ce qui amène VIA à remettre en question la pertinence du système proposé. C'est une autre raison qui explique la mise en suspens du projet.

Conclusion

Lorsqu'un transporteur ferroviaire opte pour la philosophie où il s'autorise à modifier les convois de wagons affectés aux services de train en certains points de son réseau afin de mieux satisfaire la demande en passagers et ainsi augmenter ses revenus, il prend là le pari que le gain ainsi obtenu dépasse les surcoûts liés, d'une part, à l'infrastructure et à la main d'oeuvre nécessaires pour procéder à cette modification et, d'autre part, aux bris plus fréquents dus aux manipulations. Il est donc important de disposer d'outils performants optimisant l'utilisation des wagons et locomotives, et ce tant au niveau stratégique qu'opérationnel. En effet, même si l'on dispose d'une planification bien optimisée, il est essentiel de pouvoir prendre en compte les changements de dernière minute, survenant à la suite de variations de la demande, de bris d'équipement ou de retards. Il est donc souhaitable de disposer d'un outil dont la performance garantit des temps de réponse permettant une utilisation interactive, type d'utilisation qui est souvent une nécessité dans un contexte opérationnel.

Les travaux de ce mémoire se sont concentrés sur les outils pouvant répondre aux changements dûs aux variations de la demande. Ces variations pouvant être anticipées au moins une journée à l'avance, elles ne requièrent pas l'utilisation d'un logiciel d'optimisation qui produit des solutions en quelques dizaines de secondes, on dispose plutôt, dans ce cas, de quelques dizaines de minutes pour résoudre le problème. Nous avons donc proposé pour ce type de problème un logiciel d'optimisation basé sur un modèle multiflots, non linéaire, en nombres entiers et avec contraintes additionnelles qui se résout par une méthode de génération de colonnes imbriquée dans une procédure heuristique de séparation et évaluation progressive. Avec cette approche, nous

avons montré qu'il est possible de prendre en compte des contraintes aussi complexes que le positionnement relatif de chaque wagon à l'intérieur de son convoi dans un outil répondant aux exigences du monde opérationnel, notamment en ce qui concerne les temps d'exécution. De plus, il est bien connu que les méthodes de génération de colonnes sont très souvent utilisées dans des outils de planification. Or, nos travaux montrent également que, couplées à de bonnes heuristiques, il est tout à fait envisageable d'utiliser ces méthodes dans le monde des opérations.

Dans des travaux futurs, on pourrait très bien se pencher sur les problèmes liés aux bris d'équipement et aux retards qui nécessitent cette fois des temps de résolution très courts. Il apparaît donc logique que l'on viserait des réoptimisations locales, d'abord pour toujours assurer un temps d'exécution préservant une utilisation interactive, mais surtout pour ne pas trop remettre en question le plan initial, plan qui est lui-même fruit de processus d'optimisation, mais qui a surtout servi à établir d'autres planifications en aval, telles que les horaires du personnel navigant. De là découle un certain intérêt à préserver dans la mesure du possible ce plan original. Ce type de travaux ne se bornerait pas aux transporteurs ferroviaires, mais soulèverait sans nul doute un grand intérêt auprès des transporteurs aériens qui doivent composer avec des problèmes similaires où un outil adéquat permettrait de réaliser de substantielles économies.

Bibliographie

- BARNHART, C., JOHNSON, E.L., NEMHAUSER, G. L., SAVELSBERGH, M. W. P. ET VANCE, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research* 46, 316–329.
- BEN-KHEDER, N., KINTANAR, J., QUEILLE, C. ET STRIPLING, W. K. (1997). Decision support scheduling systems for SNCF. Presented at the INFORMS Fall Meeting, Dallas.
- BOOLER, J. M. P. (1980). The solution of a railway locomotive scheduling problem. *Journal of the Operational Research Society* 31, 943–948.
- BUSSIECK, M. R., WINTER, T. ET ZIMMERMANN, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming* 79, 415–444.
- CHRISTIANSEN, M. (1999). Decomposition of a combined inventory routing and time constrained ship routing problem. *Transportation Science* 33(1), 3–16.
- CORDEAU, J.-F., DESAULNIERS, G., LINGAYA, N., SOUMIS, F. ET DESROSIERS, J. (2001a). Simultaneous locomotive and car assignment at VIA Rail Canada. *Transportation Research Part B: Methodological* 35, 767–787.
- CORDEAU, J.-F., SOUMIS, F. ET DESROSIERS, J. (2000). A Benders decomposition approach for the locomotive and car assignment problem. *Transportation Science* 34, 133–149.
- CORDEAU, J.-F., SOUMIS, F. ET DESROSIERS, J. (2001b). Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research* 49, 531–548.

- CORDEAU, J.-F., TOTH, P. ET VIGO, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science* 32, 380–404.
- DANTZIG, G. B. ET WOLFE, P. (1960). Decomposition principle for linear programming. *Operations Research* 8, 101–111.
- DESAULNIERS, G., DESROSIERS, J., IOACHIM, I., SOLOMON, M.M., SOUMIS, F. ET VILLENEUVE, D. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. Dans Crainic, T.G. et Laporte, G., éditeurs, “Fleet Management and Logistics”, pages 57–93. Kluwer, Norwell, MA.
- DESAULNIERS, G., DESROSIERS, J., SOLOMON, M.M. ET SOUMIS, F. (1997). Daily aircraft routing and scheduling. *Management Science* 43, 841–855.
- DESROCHERS, M., DESROSIERS, J. ET SOLOMON, M.M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research* 40, 342–354.
- DESROCHERS, M. ET SOUMIS, F. (1988). A generalized permanent labelling algorithm for the shortest path problem with time windows. *INFOR* 26, 191–212.
- DESROSIERS, J., DUMAS, Y., SOLOMON, M.M. ET SOUMIS, F. (1995). Time constrained routing and scheduling. Dans Ball, M. O., Magnanti, T. L., Monma, C. L. et Nemhauser, G.L., éditeurs, “Network Routing, Handbooks in Operations Research and Management Science, Volume 8”, pages 35–139. Elsevier Science, Amsterdam.
- DUBOIS, M. (2001). *Planification stratégique en transport ferroviaire de passagers avec horaire fixe et horaire variable*. Mémoire de maîtrise, École Polytechnique, Montreal, Canada.

- FLORIAN, M., BUSHELL, G., FERLAND, J., GUÉRIN, G. ET NASTANSKY, L. (1976). The engine scheduling problem in a railway network. *INFOR* 14, 121–138.
- FORBES, M. A., HOLT, J. N. ET WATTS, A. M. (1991). Exact solution of locomotive scheduling problems. *Journal of the Operational Research Society* 42, 825–831.
- HAASE, K., DESAULNIERS, G. ET DESROSIERS, J. (2001). Simultaneous vehicle and crew scheduling in urban mass transit systems. *Transportation Science* 35, 281–292.
- KLABJAN, D., JOHNSON, E.L., NEMHAUSER, G.L., GELMAN, E. ET RAMASWAMY, S. (2002). Airline crew scheduling with time windows and plane count constraints. *Transportation Science* 36, 337–348.
- KOHL, N. ET MADSEN, O.B.G. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research* 45, 395–406.
- LINGAYA, N., CORDEAU, J.-F., DESAULNIERS, G., DESROSIERS, J. ET SOUMIS, F. (2000). Operational car assignment at VIA Rail Canada. Rapport technique 2000-55, GERAD, École des Hautes Études Commerciales, Montréal, Canada.
- RAMANI, K. V. (1981). An information system for allocating coach stock on Indian Railways. *Interfaces* 11(3), 44–51.
- RAMANI, K. V. ET MANDAL, B. K. (1992). Operational planning of passenger trains in Indian Railways. *Interfaces* 22(5), 39–51.
- RAMANI, K. V. ET RAGHURAM, G. (1980). Utilisation of transportation units, an optimal allocation study. *Opsearch* 17(1), 33–40.

- RIBEIRO, C. C. ET SOUMIS, F. (1994). A column generation approach to the multiple-depot vehicle scheduling problem. *Operations Research* 42, 41–52.
- VILLENEUVE, B. (2000). *Répartition et routage d'un système automatique de chariots à l'aide d'une méthode de génération de colonnes*. Mémoire de maîtrise, École Polytechnique, Montreal, Canada.
- WRIGHT, M. B. (1989). Applying stochastic algorithms to a locomotive scheduling problem. *Journal of the Operational Research Society* 40, 187–192.
- ZIARATI, K., SOUMIS, F., DESROSIERS, J., GÉLINAS, S. ET SAINTONGE, A. (1997). Locomotive assignment with heterogeneous consists at CN North America. *European Journal of Operational Research* 97, 281–292.
- ZIARATI, K., SOUMIS, F., DESROSIERS, J. ET SOLOMON, M. M. (1999). A branch-first, cut-second approach for locomotive assignment. *Management Science* 45, 1156–1168.