

UNIVERSITÉ DE MONTRÉAL

CONSTRUCTION D'HORAIRES DE TRAVAIL DANS DES
ENVIRONNEMENTS HAUTEMENT FLEXIBLES

MONIA REKIK
DÉPARTEMENT DE MATHÉMATIQUES
ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)
(MATHÉMATIQUES DE L'INGÉNIEUR)

MAI 2006



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*
ISBN: 978-0-494-17983-3
Our file *Notre référence*
ISBN: 978-0-494-17983-3

NOTICE:

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

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

AVIS:

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

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

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

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

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

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


Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

CONSTRUCTION D'HORAIRE DE TRAVAIL DANS DES
ENVIRONNEMENTS HAUTEMENT FLEXIBLES

présentée par : REKIK Monia

en vue de l'obtention du diplôme de : Philosophiae Doctor

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

M. DESAULNIERS Guy, Ph.D., président

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

M. CORDEAU Jean-François, Ph.D., membre et codirecteur de recherche

M. ROUSSEAU Louis-Martin, Ph.D., membre

M. BARDE Jonathan F., D.Sc., membre externe

À mes très chers parents pour leur amour, leur soutien et leurs encouragements.

À mon cher Zied pour son amour, sa patience et sa présence si précieuse.

À Lilia et Kamel pour leur soutien dans les moments difficiles.

À mes adorables Mariouma et Ammoun.

REMERCIEMENTS

Je tiens d'abord à remercier très sincèrement M. François Soumis, directeur de recherche, pour son support tout au long de la thèse. Sa disponibilité, ses connaissances, ses conseils précieux, sa sensibilité humaine et ses encouragements m'ont permis de progresser rapidement.

Je suis également très reconnaissante envers M. Jean-François Cordeau, codirecteur de recherche, pour la pertinence de ses conseils, sa rigueur, sa disponibilité, et ses encouragements.

C'était à la fois un honneur et un plaisir de travailler avec vous deux.

Je remercie également M. Guy Desaulniers de bien vouloir présider mon jury ainsi que M. Louis-Martin Rousseau et M. Jonathan F. Bard d'avoir accepté d'être membres de ce jury.

Merci à tous mes amis et collègues de travail qui m'ont aidée durant ces années d'études. Un merci spécial à Hatem, Rim et Leila.

RÉSUMÉ

Cette thèse s'intéresse aux problèmes de construction d'horaires de travail anonymes dans des environnements d'opération incorporant un haut degré de flexibilité. Deux problèmes de construction d'horaires sont étudiés : le problème de construction de quarts de travail et le problème de construction simultanée de cycles et de quarts de travail. Construire des quarts consiste à spécifier les heures de début et de fin de travail durant une journée d'opération ainsi que la position d'une ou de plusieurs pauses à l'intérieur de la période de travail. Le problème de construction simultanée de cycles et de quarts de travail consiste à spécifier les jours de travail et de repos sur un horizon de planification donné ainsi que les quarts affectés à chaque jour de travail. Ces problèmes peuvent être simples à traiter dans les cas particuliers où aucune flexibilité dans la définition des quarts et des cycles n'est tolérée. La plupart des problèmes d'horaires rencontrés dans la vie réelle incorporent néanmoins beaucoup de flexibilité augmentant ainsi leur complexité.

Afin de résoudre ces problèmes, nous proposons des approches de modélisation implicites applicables à une grande variété d'environnements. Ces approches utilisent des problèmes de transport particuliers qui, selon le contexte, sont modélisés soit par un modèle classique d'affectation, soit par un ensemble de contraintes *forward* et *backward*. La première partie de la thèse étudie l'origine de ces contraintes et les conditions de leur application à un problème de transport. Ces contraintes généralisées sont ensuite utilisées pour formuler différentes formes de flexibilité a priori très complexes. Une autre partie de la thèse s'intéresse à une nouvelle forme de flexibilité relative aux pauses et étudie son impact sur le coût de la main-d'œuvre. La dernière partie est consacrée au développement de méthodes de résolution pour des problèmes de construction de quarts de grande taille.

Nous présentons d’abord une revue complète de la littérature reliée aux problèmes de construction d’horaires anonymes. Cette revue porte sur les trois catégories de problèmes rencontrés dans le domaine de construction d’horaires : le problème de construction de quarts, le problème de construction de cycles, et le problème de construction simultanée de cycles et de quarts. Pour chaque catégorie, nous rapportons les approches de modélisation ainsi que les méthodes de résolution proposées. Quelques études sur l’impact des formes de flexibilité sur le coût de la main-d’œuvre sont également exposées.

Nous proposons ensuite deux approches de modélisation implicite pour le problème de construction simultanée de cycles et de quarts de travail avec fenêtres de pauses et bandes d’heures de début. Le premier modèle utilise des problèmes de transport et s’applique à une panoplie d’environnements. Une preuve basée sur la décomposition de Benders permet, sous certaines hypothèses, de réduire considérablement le nombre de variables et de contraintes du modèle initial. Le modèle résultant est plus compact et utilise des contraintes *forward* et *backward* généralisées.

Nous étudions ensuite un problème de construction de quarts de travail avec plusieurs pauses, des pauses décomposables, et des restrictions sur la durée de travail consécutif minimal et maximal. La notion de pauses décomposables est introduite pour la première fois et ajoute beaucoup de flexibilité au problème. Elle permet à la pause d’être décomposée en plusieurs sous-pauses, sous certaines conditions. Une approche simple et efficace est proposée pour modéliser cette forme de flexibilité, apparemment complexe.

Deux formulations implicites sont proposées pour modéliser le problème. La première formulation utilise des contraintes de type “affectation” pour modéliser le concept de fenêtres de pause et des contraintes de type *forward* et *backward* pour modéliser

le concept de restrictions sur la durée de travail consécutif. La deuxième formulation n'incorpore que des contraintes *forward* et *backward* adaptées pour modéliser chacun de ces concepts. Une procédure de séparation et d'évaluation progressive est utilisée pour résoudre ces formulations. Les résultats obtenus donnent en moyenne avantage à la deuxième formulation utilisant uniquement des contraintes *forward* et *backward*. Cette formulation identifie en effet des solutions optimales en des temps très courts, et ce pour la majorité des instances.

L'utilisation des contraintes *forward* et *backward* permet de réduire la taille du modèle lorsque comparée à d'autres approches, explicites ou implicites. Nous montrons encore une fois les mérites de ces contraintes pour modéliser le concept de restrictions sur la durée de travail consécutif. Cependant, la densité de la matrice relative aux contraintes *forward* et *backward* est généralement assez élevée. Dans ce contexte, nous proposons une reformulation de ces contraintes qui permet de réduire considérablement la densité de la matrice sans augmenter le nombre de contraintes. Nous montrons, via une série de tests, que la reformulation proposée réduit les temps de calcul de façon importante.

Nous étudions ensuite l'impact de l'utilisation des pauses décomposables sur la taille de la main-d'œuvre. Nous montrons, à partir d'instances réelles, que les environnements tolérant des pauses décomposables requièrent moins de main-d'œuvre que ceux utilisant uniquement les pauses standards. Toujours dans le même cadre d'étude, nous montrons que les restrictions sur la durée de travail consécutif minimum et maximum avant et après chaque pause est un moyen très efficace pour placer les pauses au sein du quart. Ce moyen s'avère supérieur à d'autres approches classiques de placement des pauses.

Outre l'aspect modélisation, nous nous intéressons dans cette thèse aux méthodes de résolution des problèmes de construction de quarts de travail complexes s'étendant sur plusieurs jours d'opération. Trois approches de résolution sont proposées et comparées. La première approche est basée sur un branchement local adapté aux problèmes de construction de quarts. La deuxième approche est basée sur une décomposition temporelle du problème. La troisième approche correspond à la méthode de séparation et d'évaluation progressive de CPLEX 9.0. Une série de tests fournie par une agence de contrôleurs aériens montre que le branchement local proposé induit en général de meilleures solutions que la méthode de séparation et d'évaluation progressive classique. La décomposition temporelle présente, quant à elle, l'avantage de toujours identifier une solution réalisable en des temps relativement courts, et ce même pour des instances difficiles.

ABSTRACT

This dissertation addresses personnel scheduling problems for operating environments including a high degree of flexibility. Two particular scheduling problems are particularly studied : shift scheduling and tour scheduling problems. Shift scheduling problems consist of specifying daily work start and finish times as well as the position of one or more breaks within the work period. Tour scheduling problems deal with the assignment of work and rest days to employees over a given planning horizon as well as the assignment of daily shifts. These problems may be easy to solve in some simple cases where no flexibility is permitted in defining shifts and tours. However, almost all real-life problems incorporate a high degree of flexibility resulting in relatively difficult problems.

To solve these problems, we propose some implicit modeling approaches that can be applied to a large variety of scheduling environments. These approaches use transportation problems which are formulated either by a classical assignment model, or by a set of forward and backward constraints. We first study the origin of the forward and backward constraints and define the conditions in which such constraints can model transportation problems. These generalized constraints are then used to formulate different forms of flexibility. Another part of this work introduces a new form of break flexibility and studies its impact upon workforce size. The last part exposes two solution approaches for large shift scheduling problems.

We first present a complete literature review on personnel scheduling. Modeling and solution approaches are reported for shift, days-off and tour scheduling problems. We also report studies on the impact of some forms of flexibility upon workforce cost.

We then propose two implicit approaches to model a tour scheduling problem with break windows and start time bands. The first model uses transportation problems and can be applied to a large variety of problems. A proof based on a Benders decomposition considerably reduces the number of variables and constraints of the first formulation. The resulting model is more compact and uses generalized forward and backward constraints.

A shift scheduling problem with multiple breaks, fractionable breaks, and work stretch duration restrictions is then addressed. A fractionable break is a break that can be divided into several subbreaks under some conditions. This concept is new and adds a lot of flexibility to the problem. A simple and efficient approach is used to model this new difficult concept.

We propose two implicit formulations to model the problem. The first one uses classical assignment constraints to model break window constraints and forward and backward constraints to model work stretch duration restrictions. The second formulation uses only forward and backward constraints to model both concepts. A branch-and-bound procedure is used to solve these models. The instances considered show that, on average, the second formulation outperforms the first one in terms of solution times and the number of instances solved to optimality.

Using forward and backward constraints considerably reduces model size when compared to other implicit or explicit approaches. We prove once again the merits of using such an implicit approach to model work stretch duration restrictions. However, the density of the constraint matrix corresponding to the forward and backward constraints is relatively high. We thus propose a reformulation of these constraints which considerably reduces the density of the matrix without increasing

the total number of constraints. We demonstrate, through a set of real-life instances, that the proposed reformulation significantly reduces solution times.

We also study the impact of using fractionable breaks and show that it may yield a considerable workforce saving when compared to the use of only standard breaks. We also demonstrate that work stretch duration restrictions is an efficient approach to place breaks within shifts. This approach not only ensures high quality shifts but also results in a total workforce size smaller than that obtained with other traditional approaches.

Beyond the modeling aspect, this dissertation also focuses on the solution aspect for large shift scheduling problems over multiple-day planning horizons. Three solution approaches are proposed and compared. The first approach is based on a local branching strategy adapted to shift scheduling problems. The second one is based on a temporal decomposition of the problem. The third approach is a classical branch-and-bound procedure of CPLEX 9.0. A set of real-life instances from an air traffic control agency shows that the proposed local branching results, on average, in better solutions when compared to the classical branch-and-bound procedure. The decomposition approach has the advantage of yielding feasible solutions in short computing times, even for difficult instances.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	x
TABLE DES MATIÈRES	xiii
LISTE DES TABLEAUX	xix
LISTE DES FIGURES	xxi
INTRODUCTION	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE	9
1.1 Problèmes de construction de quarts de travail	9
1.1.1 Approche explicite	10

1.1.2	Approche implicite	15
1.2	Problèmes de construction de cycles de travail	21
1.3	Problèmes de construction simultanée de cycles et de quarts de travail	25
1.4	Impact des formes de flexibilité	41
CHAPITRE 2	ORGANISATION DE LA THÈSE	46
CHAPITRE 3	USING BENDERS DECOMPOSITION TO IMPLICITLY MODEL TOUR SCHEDULING	50
3.1	Introduction	56
3.2	Literature Review	59
3.3	Formulations of the Shift Scheduling Problem	63
3.3.1	Scheduling environment and assumptions	63
3.3.2	Basic formulation of the shift scheduling problem	64
3.3.3	Compact formulation of the shift scheduling problem	67
3.4	Equivalence between Formulations ($P1$) and ($P2$)	69
3.4.1	Benders reformulation	69

3.4.2	Feasibility cuts and extreme rays	72
3.4.3	Redundant constraints	73
3.5	The Tour Scheduling Problem	77
3.5.1	Additional assumptions for tour scheduling	77
3.5.2	Basic formulation of the tour scheduling problem	78
3.5.3	Compact model of the tour scheduling problem	81
3.5.3.1	Forward and backward constraints for shifts	81
3.5.3.2	Compact formulation of the tour scheduling problem	82
3.6	Extensions	83
3.7	Problem Size	85
3.8	Conclusion	90
CHAPITRE 4	IMPLICIT SHIFT SCHEDULING WITH MULTIPLE BREAKS AND WORK STRETCH DURATION RESTRICTIONS	95
4.1	Introduction	101
4.2	Literature Review	105
4.3	Implicit Formulations of the Shift Scheduling Problem	109

4.3.1	Problem definition	110
4.3.1.1	Work stretch duration restrictions	110
4.3.1.2	Fractionable breaks	112
4.3.2	Transportation problem substructure	113
4.3.2.1	Flow model	114
4.3.2.2	Forward and backward constraints	115
4.3.3	Extension of the implicit formulation of Aykin	117
4.3.4	Extension of the implicit formulation of Bechtold and Jacobs .	120
4.4	Computational Experiments	123
4.4.1	Scheduling environments	123
4.4.2	Problem size comparison of models	125
4.4.3	Solution time comparison of models	127
4.4.4	Impact of fractionable breaks on workforce size	133
4.4.5	Impact of work stretch duration restrictions	134
4.4.5.1	Work stretch duration restrictions approach versus ideal break start time approach	135

4.4.5.2	Modeling work stretch duration restrictions	138
4.5	Conclusion	143
CHAPITRE 5 SOLVING LARGE SHIFT SCHEDULING PROBLEMS . .		153
5.1	Introduction	158
5.2	Literature Review	160
5.3	Mathematical Models	164
5.3.1	Generalized forward and backward constraints	165
5.3.2	Problem definition	166
5.3.3	Implicit formulation ($P1$)	168
5.3.4	Implicit formulation ($P2$)	170
5.4	Solution Approaches	173
5.4.1	Adapted local branching strategy	173
5.4.2	Time windowing approach	178
5.5	Computational Experiments	183
5.5.1	Scheduling environments and problem size	183

5.5.2	Comparison of the solution approaches	185
5.6	Conclusion	191
	DISCUSSION GÉNÉRALE ET CONCLUSION	196
	BIBLIOGRAPHIE	203

LISTE DES TABLEAUX

Tableau 3.1	Comparison of model sizes	90
Tableau 4.1	Description of the scheduling environments of SET2	125
Tableau 4.2	Comparison of model size for SET1	127
Tableau 4.3	Comparison of model size for SET2	127
Tableau 4.4	Solution times for models ($P1$) – ($P4$) for environment #1 of SET2	130
Tableau 4.5	Computational results for SET1	130
Tableau 4.6	Computational results for SET2	131
Tableau 4.7	Impact of fractionable breaks with original demand patterns .	134
Tableau 4.8	Impact of fractionable breaks with perturbed demand patterns	134
Tableau 4.9	Number of employees with the work stretch duration restric- tion approach and the ideal break start time approach	137
Tableau 4.10	Number of employees and solution times for both approaches .	139
Tableau 4.11	Number of employees and solution times for the implicit approach and problem 1	142

Tableau 5.1	Model size for (P1) and (P2)	185
Tableau 5.2	Computational results for model ($P1$)	188
Tableau 5.3	Computational results for model ($P2$)	189

LISTE DES FIGURES

Figure 4.1	Feasible regions associated with the implicit and the enumerative approaches	143
Figure 4.2	Demand patterns arising from the air-traffic control agency	150
Figure 4.3	Demand patterns arising from the United State Postal Service of Oklahoma City	151
Figure 4.4	Perturbed demand patterns	152
Figure 5.1	Basic local branching strategy	176

INTRODUCTION

Le problème de confection d'horaires de travail a suscité beaucoup d'intérêt depuis plusieurs décennies. De façon simplifiée, le problème consiste à élaborer des horaires de travail répondant aux besoins de l'organisation tout en tenant compte des compromis entre le coût de la main-d'œuvre et les conditions de travail des employés.

Pour l'organisation, le soucis principal est de satisfaire la demande en personnel sur un horizon temporel donné. Ces besoins peuvent être soit fixes soit variables d'une saison à l'autre, d'un jour à l'autre, ou au courant d'une même journée. La prévision de ces besoins constitue un problème en soi. Par exemple, dans le milieu de la santé il est parfois difficile de prédire avec une précision raisonnable les effectifs requis durant une journée d'opération. Généralement, la détermination des besoins en employés est traitée à l'avance par un processus externe et ces besoins, bien qu'estimés par des méthodes probabilistes, sont considérés comme déterministes lors de la confection d'horaires. L'ajustement du personnel aux variations imprévues de ces besoins est fait en temps réel.

Pour les employés, les horaires établis doivent répondre à certaines règles de la convention collective. Ils doivent entre autres garantir une durée minimale de repos au courant d'une même journée, entre deux jours de travail consécutifs, et sur le nombre total de jours travaillés. Les horaires établis ne doivent cependant pas inclure beaucoup de périodes creuses afin d'assurer un bon niveau de productivité. Certains employés peuvent également exprimer leurs préférences quant aux jours de repos souhaités, aux périodes travaillées durant la journée (jour, soir, ou nuit), etc. Dans diverses organisations, on convient d'accorder des compensations monétaires aux employés ayant des horaires de mauvaise qualité. L'objectif inclura alors

la maximisation des préférences des employés ou la minimisation du coût des compensations monétaires.

La confection d'horaires de travail implique la résolution de plusieurs problèmes plus ou moins liés entre eux. Le processus de confection peut être séparé en deux étapes principales : la première étape consiste à construire des horaires anonymes, c'est-à-dire, des horaires qui répondent aux besoins de l'organisation et aux règles générales de la convention collective sans préciser quel horaire attribuer à quel employé. La seconde étape vise à affecter ces horaires anonymes aux employés en tenant compte de leurs préférences, de leur séniorité, de leurs compétences, etc. Dans le cadre de ce travail de recherche, nous nous intéressons uniquement aux problèmes de construction d'horaires anonymes.

Deux problèmes doivent être traités lors de la construction d'horaires anonymes : le problème de *construction de cycles de travail (days-off scheduling)* et le problème de *construction de quarts de travail (shift scheduling)*. La construction de cycles permet de spécifier les séquences de jours de travail et de repos sur un horizon de planification donné. Il convient de noter que le mot "cycle", tel qu'utilisé ici, n'implique pas obligatoirement un horaire cyclique mais désigne plutôt toute séquence de jours de travail et de repos, qu'elle soit périodique ou non. La construction de quarts détermine les heures de début et de fin de travail durant une journée d'opération ainsi que la durée et la position d'une ou de plusieurs pauses à l'intérieur de la période de travail.

Depuis quelques années, on a commencé à combiner les deux problèmes ci-dessus pour résoudre un problème plus global de *construction simultanée de cycles et de quarts de travail (tour scheduling)*. Ce dernier problème spécifie à la fois les jours travaillés et les périodes de travail et de repos correspondant à chaque jour de travail.

Pour chacun des problèmes de construction d'horaires, l'objectif le plus couramment considéré est la minimisation du coût total de la main-d'œuvre sur l'horizon de planification. L'étendue de cet horizon dépend de la nature du problème ainsi que de l'aptitude de l'organisation à prévoir ses besoins en main-d'œuvre.

Plusieurs contraintes sont imposées aux problèmes de construction d'horaires selon les exigences des organisations. Un ensemble de contraintes, dites *contraintes de demande*, est néanmoins présent dans tous les problèmes d'horaires. Une contrainte de demande exprime le besoin en main-d'œuvre de l'organisation pour une tranche de temps de l'horizon de planification. Elle indique le nombre minimal d'employés requis durant cette période. Dans la plupart des cas, les contraintes de demande sont des contraintes dures qu'il faut obligatoirement satisfaire. Dans d'autres cas, un déficit d'employés est toléré moyennant une certaine pénalité dans la fonction objectif.

Outre les besoins en main-d'œuvre, plusieurs autres contraintes peuvent être imposées au problème. Le nombre, la nature et la complexité de ces contraintes dépendent des règles de la convention collective de l'organisation et de sa volonté à construire des horaires de bonne qualité qui satisfont les employés tout en améliorant leur productivité. Dans le but de répondre au mieux à tous ces objectifs et afin d'avoir une marge de manoeuvre permettant d'optimiser le processus de construction d'horaires, les environnements de travail tolèrent généralement un certain degré de flexibilité dans la définition des cycles et des quarts. Plusieurs études ont rapporté qu'outre son aspect réaliste, l'incorporation des formes de flexibilité permet d'améliorer la qualité des horaires et de réduire considérablement le coût total de la main-d'œuvre.

La flexibilité peut intervenir à plusieurs niveaux. Pour le problème de construction de cycles, le degré de flexibilité est mesuré à partir du nombre et du type de séquences de jours de travail et de repos utilisées. La détermination de ces séquences dépend, entre autres, du nombre de jours travaillés, du nombre de jours travaillés consécutivement, du nombre de fins de semaine au repos, etc.

Pour le problème de construction de quarts, certaines organisations tolèrent des heures de début de travail à n'importe quelle période de la journée. D'autres limitent ces heures de début en imposant un temps de séparation minimal entre eux ou en utilisant un ensemble restreint d'heures de début. Bien que l'absence de restrictions sur les heures de début de quarts témoigne d'une grande flexibilité, elle rend certaines tâches administratives plus complexes. Les règles de la convention collective et les préférences des employés peuvent également encourager à les limiter. Un des problèmes étroitement liés à la construction des quarts est justement le choix d'un ensemble adéquat d'heures de début à partir duquel les quarts optimaux seront construits. Des études récentes ont même rapporté que des horaires de bonne qualité pouvaient être obtenus avec des ensembles relativement petits d'heures de début de travail (quatre à cinq heures par jour). Ces ensembles doivent cependant inclure les bonnes valeurs pour donner une solution comparable à celle obtenue avec toutes les heures de début de travail possibles, d'autant plus que la qualité de la solution se détériore rapidement lorsque l'ensemble adéquat n'est pas utilisé. Un environnement de construction de quarts flexible considère également des quarts de différentes longueurs. La *longueur d'un quart* correspond à la durée de temps écoulée entre le début et la fin du travail incluant les pauses. Ces longueurs peuvent être fixées à l'avance ou décidées par l'optimiseur. Dans ce dernier cas, des longueurs de quarts, minimale et maximale, sont imposées pour éviter de générer des quarts trop courts ou trop longs. Le placement des pauses au sein des quarts peut également renfermer de la flexibilité en permettant à la pause de débiter à l'intérieur d'une

fenêtre de temps plutôt qu'à une heure fixe. La position de ces fenêtres de pauses est généralement pré-spécifiée à l'avance par les dirigeants ou par certaines approches telles que l'approche du temps idéal d'heure de début de pause (*ideal break start time approach*).

Outre les formes énumérées ci-haut, d'autres formes de flexibilité peuvent apparaître lors de l'intégration des problèmes de construction de cycles et de quarts. Ainsi, en permettant aux employés de débiter le travail à n'importe quelle période de la journée, l'organisation peut se montrer encore plus flexible et utiliser des horaires incluant des quarts commençant à différentes heures pour chaque jour du cycle. Cependant, dans le cas où la durée d'opération de l'organisation approche les 24 heures, les employés doivent recevoir un minimum de temps de repos entre deux jours de travail consécutifs. Afin d'éviter ce problème et de faciliter la gestion de la main-d'œuvre, certaines compagnies exigent au contraire que les employés commencent leur travail toujours à la même heure pour chacun des jours du cycle. Une solution intermédiaire a été de définir une *bande d'heures de début de quarts* (*start time band*) par cycle. Une bande d'heures de début est un champ (qui peut être un intervalle de temps ou un ensemble de valeurs) à l'intérieur duquel les heures de début de travail peuvent varier d'un jour à l'autre du cycle. Une autre solution, contrôlant la variation des heures de début de travail, est de permettre un décalage entre les heures de début de quarts d'un jour à l'autre du cycle. Donc, au lieu de limiter les valeurs des heures de début à un ensemble restreint pour tout le cycle (comme c'est le cas pour la bande d'heures de début), le concept de décalage tient compte uniquement de l'heure de début du quart attribué au jour précédant le jour courant, si ces deux jours correspondent à des jours de travail.

Les études effectuées dans le domaine de la construction d'horaires ont démontré que plus le degré de flexibilité est grand plus le problème de construction d'horaires

devient complexe et difficile à résoudre. Certaines de ces études ont également prouvé que l'incorporation des différentes formes de flexibilité devient d'autant plus complexe si elle est appliquée à un environnement d'opération continue. Un environnement continu suppose une demande en employés et donc une période d'opération de 24 heures sur 24 et de sept jours sur sept. La difficulté inhérente à ces *problèmes de construction d'horaires continus* réside, entre autres, dans le fait que, contrairement au cas discontinu (moins de 24 heures d'opération par jour), les quarts de travail peuvent chevaucher deux jours consécutifs.

Une autre classe de problèmes de construction d'horaires étudiée par beaucoup de chercheurs est celle des horaires cycliques. Ce type de problèmes est rencontré lorsque les ressources disponibles et la demande à satisfaire ont un profil cyclique. La matrice des contraintes de demande peut alors être soit 0-1 circulaire soit une sous-matrice d'une matrice 0-1 circulaire. Une matrice circulaire est obtenue en prenant arbitrairement un vecteur colonne et en le décalant cycliquement d'une position pour obtenir les colonnes successives. Dans le cas où les ressources sont continuellement disponibles, la matrice des contraintes est circulaire et le problème de construction d'horaires cycliques est résolu efficacement. Pour les problèmes plus réalistes, les ressources étant disponibles de façon intermittente (pauses, maintenance des machines, etc.), la matrice des contraintes ne possède plus la propriété de circularité. Ce type de problème est NP-complet.

Pour la plupart des problèmes traités dans la littérature, les employés sont supposés disponibles en tout temps durant la période d'opération. Quelques chercheurs ont néanmoins abordé le cas où les employés ne sont disponibles qu'à certaines périodes de la journée. L'heure de début et l'étendue des quarts construits doivent alors tenir compte de ces disponibilités limitées.

Une autre difficulté s'ajoute aux problèmes de construction d'horaires lorsque la main-d'œuvre est hétérogène. Les employés sont alors classés par catégorie selon leur ancienneté, leurs qualifications, leur productivité, leur salaire, etc. Les besoins en employés sont alors spécifiés pour chaque période et chaque catégorie. Le problème est, certes, plus conforme à la réalité, mais il devient plus complexe et de plus grande taille (les types de cycles et de quarts permis peuvent différer d'une catégorie à l'autre). Certains paramètres peuvent également entrer en considération tels que la hiérarchisation des catégories et les possibilités de substitution (un employé d'une certaine catégorie est capable d'effectuer le travail d'un employé d'une autre catégorie alors que l'inverse n'est pas possible). Un ensemble de contraintes limitant la proportion d'employés de chaque catégorie est parfois considéré afin de maintenir un bon niveau de la qualité de service. Un cas particulier de la classification des employés est de distinguer entre les employés à temps plein et ceux à temps partiel. Les employés à temps partiel sont dans la plupart des cas affectés à des quarts de courte durée et sans pause.

Une autre variante des problèmes de construction d'horaires est obtenue en classant, non pas les employés, mais les activités par catégorie. Cette caractéristique est présente, entre autres, dans les compagnies aériennes. La demande est alors spécifiée pour chaque période de l'horizon et pour chaque type d'activité. Une liste est dressée pour chaque employé indiquant les activités pour lesquelles il est qualifié. Devant tenir compte de cet aspect *multi-activités*, un horaire journalier n'est plus totalement défini par la spécification des heures travaillées, mais il faut en plus spécifier les activités qui le composent. L'objectif peut varier selon le type de problème traité. Pour le problème de construction de quarts, on peut minimiser le nombre de transitions entre les activités, c'est-à-dire, éviter qu'un employé change d'activités trop souvent durant un quart.

Étant donné le nombre important de contraintes à satisfaire, la difficulté inhérente à leur représentation, la taille importante et la nature combinatoire du problème résultant, la construction d'horaires de travail est considérée comme une activité très complexe qui demande beaucoup de temps et d'énergie. Les méthodes manuelles de type "essais et erreurs" exigent beaucoup de temps et donnent souvent des solutions médiocres. De plus, vu l'importance et l'omniprésence de ce problème, les spécialistes de la recherche opérationnelle ont fait beaucoup d'études dans ce domaine. Ils ont démontré que des résultats très satisfaisants, et surtout supérieurs à ceux obtenus à l'aide des méthodes manuelles, pouvaient être atteints en un temps et à un coût moindres. Ces études ont également prouvé qu'une utilisation efficace de la main-d'œuvre constitue un des facteurs essentiels pour améliorer la productivité. Une multitude d'applications a été traitée dans différents domaines tels que les compagnies de téléphone, les écoles, les hôpitaux, les banques, les pompiers, les sociétés de transport en commun, les compagnies aériennes, etc.

Le chapitre 1 présente une revue détaillée de la littérature concernant les problèmes de construction d'horaires anonymes. Cette revue rapporte les différentes approches de modélisation proposées jusque-là ainsi que quelques méthodes de résolution heuristiques et exactes. Cette thèse s'intéressant de près à la présence de la flexibilité lors de la construction d'horaires, nous énumérons à la fin du premier chapitre les différentes formes de flexibilité abordées dans la littérature de même que leur impact sur le coût total de la main-d'œuvre. Les chapitres 3, 4 et 5 représentent les trois articles élaborés dans le cadre de la thèse. Une brève description de ces articles, de leurs contextes et des résultats obtenus est présentée au chapitre 2.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

Différentes variantes des problèmes de construction d'horaires ont été traitées dans la littérature. L'objectif principal de ces travaux est de produire des horaires de bonne qualité susceptibles de répondre aux besoins de l'organisation tout en respectant les règles de la convention collective. Récemment, ERNST *et al.* (2003) ont dressé une revue relativement complète de la littérature des problèmes d'horaires (plus de 700 articles). Ce chapitre est une vue d'ensemble sur certains travaux effectués dans le domaine de la construction d'horaires anonymes. Nous nous sommes intéressés plus particulièrement aux travaux abordant des environnements d'opération plus ou moins semblables à ceux considérés dans le cadre de cette thèse.

BAKER (1976) a été le premier à classer les problèmes de construction d'horaires en trois catégories: la construction de cycles, la construction de quarts et la construction simultanée de cycles et de quarts. Nous adopterons cette classification pour rapporter les différents modèles et les différentes méthodes de résolution proposés pour chaque catégorie.

1.1 Problèmes de construction de quarts de travail

L'ensemble des travaux de recherche effectués dans le domaine de la construction de quarts de travail révèle deux principales approches de modélisation. La première approche, dite *explicite*, utilise une formulation généralisée du problème de recouvre-

ment et a été initialement introduite par DANTZIG (1954). Cette approche suppose une représentation explicite de chaque quart réalisable par une variable de décision. La deuxième approche, dite *implicite*, est une approche qui tend à réduire le nombre de variables de quart en considérant une variable de décision, non pas pour chaque quart explicite, mais plutôt pour chaque *type* de quart. Un type de quart regroupe les quarts ayant certaines caractéristiques en commun. Une telle représentation permet de réduire la taille de l'information fournie au modèle mais doit, en même temps, garantir une solution explicite réalisable à la fin du processus de résolution. Dans ce qui suit nous étudions plus en détail chacune de ces approches et nous rapportons différents travaux utilisant l'une ou l'autre de ces approches.

1.1.1 Approche explicite

DANTZIG (1954) est le premier à avoir proposé un modèle de recouvrement pour le problème de construction de quarts de travail. Nous présentons ci-dessous la formulation généralisée du modèle de recouvrement de DANTZIG (1954), notée (*FGR*), qui ne tient compte que des contraintes de demande.

Rappelons qu'une contrainte de demande est définie pour chaque période de la journée pour assurer la présence d'un nombre minimal d'employés durant cette période. L'étendue de la période dépend des données du problème telles que la variation de la demande, les heures de début des quarts, la durée des pauses, etc. La journée d'opération est ainsi divisée en des intervalles de temps de durées égales (périodes) et les données temporelles du problème ne sont plus exprimées en termes d'heures ou de minutes mais en termes de périodes. Pour le problème de construction de quarts, on considère communément des périodes de 15, 30 ou 60 minutes.

Dans la formulation (*FGR*) ci-dessous, I représente l'ensemble de toutes les périodes de la journée et d_i dénote le nombre minimal d'employés requis durant chaque période $i \in I$. L'ensemble J contient tous les quarts explicites réalisables qui satisfont les règles de la convention collective. Un ensemble de constantes a_{ij} est défini pour chaque quart $j \in J$ tel que a_{ij} vaut 1 si le quart j couvre la période i , et 0 sinon. Un quart j est considéré comme couvrant la période i si la période i appartient à l'étendue du quart et ne correspond pas à une période de pause pour ce quart. Une variable de décision entière x_j est associée à chaque quart j et représente le nombre d'employés qui lui sont affectés.

$$(FGR) \quad \text{Minimiser} \quad \sum_{j \in J} x_j \quad (1.1)$$

$$\sum_{j \in J} a_{ij} x_j \geq d_i \quad \forall i \in I \quad (1.2)$$

$$x_j \geq 0 \text{ et entier} \quad \forall j \in J \quad (1.3)$$

La fonction objectif (1.1) minimise le nombre total d'employés. Les contraintes (1.2) assurent que le nombre d'employés présents et qui ne sont pas en pause durant la période i est supérieur ou égal au nombre d'employés requis durant cette période. Les contraintes (1.3) correspondent aux contraintes d'intégrité.

DANTZIG (1954) a appliqué ce modèle de recouvrement au problème de construction de quarts de travail des employés des cabines de péage. Il a rapporté que la résolution de la relaxation linéaire du modèle donne souvent des solutions entières. Dans le cas contraire, il propose d'arrondir à l'entier le plus proche les valeurs fractionnaires, ce qui ne garantit pas une solution entière réalisable. Les modèles de type (*FGR*) ont ensuite été utilisés par plusieurs chercheurs.

SEGAL (1974), entre autres, s'est basé sur un modèle de recouvrement pour développer deux problèmes de flot à coût minimum. Le premier problème modélise le cas où aucune pause n'est considérée et permet de construire des quarts *idéaux* (sans pause) optimaux. L'attribution des pauses est faite ensuite de façon heuristique à l'aide d'un deuxième problème de flot. SEGAL (1974) a rapporté que de bonnes solutions pouvaient être obtenues efficacement avec une telle approche de décomposition.

HENDERSON et BERRY (1976) ont proposé un modèle de recouvrement pouvant inclure plusieurs quarts et où les pauses sont considérées à chaque étape de la procédure de résolution. Le modèle obtenu étant gros (jusqu'à 15 000 quarts), la résolution du problème est faite en deux étapes. À la première étape, l'ensemble des quarts est réduit à un sous-ensemble restreint moyennant une heuristique de sélection. Le modèle avec sous-ensemble restreint est ensuite résolu heuristiquement à la deuxième étape. Les auteurs proposent une heuristique de sélection, dénotée par *maxdif*, qui, à chaque itération, choisit le quart qui maximise la différence entre les périodes couvertes (par rapport aux quarts déjà sélectionnés). Un des avantages de cette méthode est que les besoins en employés ne sont pas considérés lors de la sélection des quarts. De ce fait, un changement dans la demande ne nécessite pas la régénération du sous-ensemble restreint. Cependant, la procédure de sélection doit évaluer chaque quart réalisable à chaque itération. Trois heuristiques sont ensuite proposées pour résoudre le problème avec sous-ensemble restreint. Ces heuristiques utilisent la solution fournie par la résolution de la relaxation linéaire du problème. Elles transforment différemment la solution linéaire fractionnaire en une solution entière. La première heuristique commence par arrondir la valeur de la solution fractionnaire à l'entier supérieur. Les variables sont ensuite examinées une à une et chaque variable non nulle est réduite d'une unité pourvu qu'aucune contrainte de demande ne soit violée. La deuxième heuristique utilise comme horaire

de départ celui fourni par la première heuristique. Les employés sont échangés entre les différents quarts pour tenter d'améliorer la solution courante. L'énumération des échanges possibles est arrêtée dès que la valeur de la fonction objectif atteint la borne inférieure calculée à partir de la valeur optimale de la relaxation linéaire. La troisième heuristique repose sur le même principe de recherche locale que la précédente sauf que la solution initiale entière est générée aléatoirement. HENDERSON et BERRY (1976) ont comparé les performances des différentes combinaisons de ces heuristiques pour plusieurs profils de données, et différentes tailles du sous-ensemble restreint de quarts (jusqu'à 50 quarts). Ils ont conclu que, pour les instances considérées, l'utilisation d'un sous-ensemble restreint de quarts pouvait quand même produire de bonnes solutions.

KEITH (1979) a considéré le problème de construction de quarts où deux types de pause sont alloués. Il a supposé, en un premier temps, que les pauses sont données à des heures fixes et il a résolu la relaxation linéaire du problème de recouvrement correspondant. Les valeurs de la solution optimale sont ensuite adéquatement arrondies pour obtenir une solution réalisable. En fait, KEITH (1979) propose une procédure d'ajout/suppression en trois phases. La première phase arrondit la valeur fractionnaire à l'entier le plus proche, la deuxième phase ajoute un employé à l'horaire qui réduit le plus la sous-couverture totale, et la dernière phase ré-examine les horaires affectés pour essayer de diminuer le nombre total d'employés. Dans la deuxième partie de l'algorithme, les intervalles de pauses et les possibilités de placement sont examinés heuristiquement pour tenter d'améliorer la solution courante.

MORRIS et SHOWALTER (1983) ont opté pour un environnement de travail moins flexible où tous les quarts ont la même durée (neuf heures), débutent à toutes les heures, et incluent une pause de 60 minutes au début de leur cinquième heure.

Toutes ces simplifications ont induit un modèle de recouvrement de taille réduite (24 variables et 24 contraintes) qui a pu être résolu à l'optimalité avec une méthode adaptée de séparation et d'évaluation progressive. L'approche de résolution proposée est générale et peut être appliquée à des problèmes plus complexes. La relaxation linéaire du modèle est d'abord résolue. La valeur optimale obtenue est arrondie à l'entier supérieur et constitue une borne inférieure pour le problème en nombres entiers. Cette borne est ensuite incorporée au modèle et la relaxation linéaire est résolue de nouveau. Si la nouvelle solution obtenue est entière, l'optimalité est atteinte et le processus s'arrête. Sinon, une procédure de séparation et d'évaluation progressive est exécutée pour voir si une solution entière avec un objectif égal à la borne inférieure courante existe. Si c'est le cas, la solution correspondante est optimale, sinon la borne inférieure est augmentée de 1, la coupe est mise à jour et la même procédure est répétée. Le succès de cette approche avec plans de coupes était prévisible puisque, toujours selon les auteurs, ces coupes sont très efficaces avec les problèmes en nombres entiers de type recouvrement.

BARTHOLDI (1981) a utilisé un modèle de recouvrement généralisé pour résoudre le problème d'horaires cycliques. La matrice des contraintes, telles que formulées dans (*FGR*), est composée de colonnes avec des 1 successifs, et peut être soit 0-1 circulaire soit une sous-matrice d'une matrice 0-1 circulaire. Rappelons qu'une matrice circulaire est obtenue en prenant arbitrairement un vecteur colonne et en le décalant cycliquement d'une position pour obtenir les colonnes successives. Dans le cas où les ressources sont continuellement disponibles, la matrice des contraintes est circulaire et le problème d'horaires cycliques est résolu efficacement avec une méthode d'arrondi basée sur la relaxation linéaire (BARTHOLDI, 1981). Pour les modèles plus réalistes, les ressources sont disponibles de façon intermittente, et la matrice des contraintes ne possède plus la propriété de circularité. BARTHOLDI (1981) a montré que ce type de problèmes est NP-complet.

Plus récemment, THOMPSON (1996a) a utilisé un modèle de recouvrement généralisé pour un problème de construction de quarts où les employés ne sont pas continuellement disponibles. Le modèle proposé utilise un ensemble de quarts admissibles pour chaque employé qui dépend de sa disponibilité. La sous-couverture et la surcouverture sont également tolérées moyennant une pénalité dans la fonction objectif. Le modèle est résolu avec la méthode du recuit simulé. L'auteur rapporte que cette méthode permet d'avoir des solutions de qualité légèrement inférieure aux solutions optimales.

HOEY et LOVE (1990) avaient déjà traité un problème similaire où il fallait tenir compte de la disponibilité des employés et de leurs qualifications. Le problème est formulé par deux modèles de flot. Le premier modèle génère des quarts dans le but de minimiser la surcouverture tandis que le deuxième modèle affecte les quarts obtenus à la première étape aux employés en tenant compte de leur disponibilité. Étant donné que la construction des quarts à la première étape suppose que les employés sont disponibles en tout temps, rien ne garantit une solution réalisable à la deuxième étape.

1.1.2 Approche implicite

La formulation explicite (*FGR*) requiert l'énumération de tous les horaires réalisables. Pour des environnements d'opération incorporant plusieurs formes de flexibilité, le nombre de quarts explicites réalisables et donc de variables entières (x_j) dans le modèle de recouvrement croît considérablement, rendant ainsi le modèle intraitable ou du moins difficile à résoudre à l'optimalité. Certains chercheurs ont opté pour des méthodes heuristiques qui produisent de bonnes solutions réalisables sans pour autant garantir leur optimalité. Quelques unes de ces heuristiques ont été

présentées à la section précédente. D'autres chercheurs se sont plutôt intéressés à l'aspect modélisation et ont essayé de réduire la taille des modèles. En fait, l'approche implicite a vu le jour lorsque l'on a réalisé qu'une réduction de la taille du modèle pouvait être obtenue en réduisant la taille des informations requises et en représentant implicitement certaines données du problème.

Dans ce cadre, MOONDRA (1976) a développé un modèle linéaire (contraintes d'intégrité relaxées) dans lequel les heures de début et les durées des quarts des employés à temps partiel sont représentées implicitement. Deux types de variables sont utilisés pour chaque heure de début et chaque heure de fin de travail possible. Ces variables représentent le nombre d'employés à temps partiel commençant ou finissant leur travail à ces périodes. Des contraintes limitant la durée des quarts des employés à temps partiel sont incorporées au modèle. Une contrainte est également ajoutée pour limiter le nombre total d'heures travaillées par les employés à temps partiel par rapport à la demande totale. Les employés à temps plein sont affectés à des quarts de huit heures débutant à une heure fixe. Ils sont autorisés à faire deux heures supplémentaires au maximum par jour. Des variables représentant le nombre d'employés à temps plein travaillant l'une ou l'autre des heures supplémentaires sont considérées. Les employés à temps partiel n'ont pas de pauses. Ceux à temps plein reçoivent une pause d'une heure. L'attribution de cette pause est à peine flexible dans la mesure où la moitié des employés à temps plein reçoivent leur pause à la troisième heure du quart et l'autre moitié au début de la quatrième heure. Le modèle implicite proposé garantit certes un horaire réalisable, mais l'auteur ne donne pas de détail sur la procédure à utiliser pour construire les quarts explicites à partir des heures de début et de fin de travail obtenues.

GABALLA et PEARCE (1979) ont, quant à eux, développé un modèle de programmation en nombres entiers qui modélise implicitement la flexibilité dans le placement

des pauses. La journée d'opération est discrétisée en des périodes de 30 minutes et utilise sept quarts différents. Chaque quart doit recevoir une pause d'une certaine durée devant débiter au sein d'un intervalle de temps spécifique. Pour modéliser ce problème, les auteurs utilisent deux types de variables entières. Une variable pour chaque quart qui représente le nombre d'employés affectés à ce quart et une variable pour chaque quart et pour chaque période pouvant correspondre à une pause pour ce quart qui représente le nombre d'employés affectés à cette pause. Une contrainte d'égalité entre la variable de quart et les variables de pause associées est incorporée au modèle afin d'assurer pour chaque quart une pause débutant dans l'intervalle correspondant. Le nombre de variables nécessaires au modèle proposé par GABALLA et PEARCE (1979) demeure néanmoins supérieur à celui requis par un modèle de recouvrement équivalent.

Dès lors, la flexibilité dans le placement des pauses a été dûment étudiée. Une réduction importante de la taille des modèles a été accomplie par l'association des variables de pause aux périodes de temps plutôt qu'explicitement aux quarts. L'affectation des pauses aux quarts est faite a posteriori lorsqu'une solution optimale du modèle implicite est obtenue.

Ce sont BECHTOLD et JACOBS (1990) qui ont introduit pour la première fois un modèle implicite, intéressant par rapport au modèle de recouvrement, pour représenter la flexibilité dans le placement des pauses. L'environnement d'opération considéré est discontinu et utilise des types de quarts plutôt que des quarts. Un type de quart est défini par une heure de début, une durée et un intervalle de temps spécifiant les périodes durant lesquelles une pause peut être attribuée (*break window*). La pause n'est plus associée au quart explicitement mais elle est spécifiée uniquement par son heure de début. L'ensemble de toutes les pauses possibles est déduit à partir de tous les intervalles de pauses associés à tous les types de quarts.

Ces intervalles de pauses sont composés de périodes de temps continues et supposent l'absence de ce que les auteurs ont défini comme *extra-chevauchement* (*extraordinary overlap*). Un extra-chevauchement a lieu lorsqu'il existe un type de quart pour lequel l'intervalle de pause débute strictement avant et finit strictement après l'intervalle de pause associé à un autre type de quart. Le modèle proposé par BECHTOLD et JACOBS (1990) utilise donc deux ensembles de variables : des variables de type de quart qui représentent le nombre d'employés affectés à ce type de quart et des variables de pause qui représentent le nombre d'employés recevant cette pause. Des contraintes, dites *forward* et *backward*, sont incorporées au modèle pour assurer l'attribution d'une pause à chaque quart à l'intérieur de son intervalle de pauses permises. Les contraintes *forward* et *backward* assurent que pour un intervalle de temps adéquatement défini, le nombre de pauses affectées est supérieur ou égal au nombre de quarts ayant leur intervalle de pauses totalement inclus dans cet intervalle. Nous développerons davantage ces contraintes au chapitre 3.

BECHTOLD et JACOBS (1990) ont opté pour la méthode de séparation et d'évaluation progressive avec plans de coupes proposée par MORRIS et SHOWALTER (1983) pour résoudre le modèle. Une fois la solution optimale trouvée, les variables de type de quart et les variables de pause sont fixées à leurs valeurs optimales et une simple procédure d'affectation est implantée pour obtenir les quarts optimaux explicites. Les auteurs ont comparé leur modèle implicite au modèle de recouvrement de DANTZIG (1954). Ils ont montré, à travers un ensemble de tests, que leur modèle est capable de résoudre les instances considérées en des temps de calcul plus courts et en utilisant moins d'espace mémoire. Plus tard, BECHTOLD et JACOBS (1996) ont démontré que leur modèle implicite est équivalent au modèle de recouvrement (*FGR*) sous l'hypothèse de continuité des intervalles de pauses et d'absence d'extra-chevauchement. Bien que l'interdiction de l'extra-chevauchement ne présente pas une grande restriction par rapport aux problèmes réels, BECHTOLD et JACOBS

(1990) ont démontré, à travers quelques instances, que leur modèle était capable de générer des solutions optimales réalisables en présence d'un certain degré d'extra-chevauchement.

Plus récemment ADDOU et SOUMIS (2004) ont proposé un modèle qui étend celui de BECHTOLD et JACOBS (1990) au cas où un extra-chevauchement est présent. Ils ont démontré qu'un ensemble très restreint de contraintes doit être ajouté au modèle de BECHTOLD et JACOBS (1990) pour traiter l'extra-chevauchement. Une unique contrainte doit en effet être ajoutée par extra-chevauchement présent. Plus encore, les auteurs ont démontré que, dans certains cas, les contraintes *forward* et *backward* de BECHTOLD et JACOBS (1990) pouvaient être redondantes. Ils ont proposé un ensemble plus restreint de contraintes et ont démontré que cet ensemble est minimal, c'est-à-dire, qu'il contient le nombre nécessaire et suffisant de contraintes *forward* et *backward*.

AYKIN (1996) a considéré la même approche proposée par GABALLA et PEARCE (1979) pour représenter implicitement les pauses. Il a étendu leur modèle au problème où plusieurs pauses doivent être attribuées à un quart dans des intervalles de temps pré-spécifiés. Contrairement aux instances considérées par GABALLA et PEARCE (1979) qui, rappelons-le, induisaient un modèle de plus grande taille que le modèle de recouvrement équivalent, AYKIN (1996) a montré que pour la plupart des instances testées, son modèle était largement supérieur au modèle de recouvrement en termes de temps de résolution et de nombre d'instances résolues à l'optimalité. AYKIN (1996) a introduit des bornes supérieures pour les variables de quart. Ces bornes sont calculées à partir des contraintes de demande. Il a rapporté que ces coupes réduisent les temps de calcul lors du processus de séparation et d'évaluation progressive.

Plus tard, AYKIN (1998) a proposé une méthode de résolution plus élaborée dans laquelle des coupes relatives à la fonction objectif et aux variables de quart sont générées dynamiquement à chaque étape du processus de résolution. La relaxation linéaire du modèle est d'abord résolue et donne une borne inférieure pour la fonction objectif. Une heuristique est ensuite appliquée à la solution de la relaxation linéaire pour avoir une solution entière réalisable. Une borne supérieure est ainsi obtenue pour la fonction objectif. Les coupes sont ajoutées et mises à jour à chaque étape et le problème est résolu par la méthode de séparation et d'évaluation progressive. AYKIN (1998) a rapporté que cette méthode de séparation et d'évaluation progressive avec coupes est capable de résoudre des problèmes incluant jusqu'à 32 928 quarts explicites.

AYKIN (2000) a comparé son approche à celle utilisée par BECHTOLD et JACOBS (1990). Pour cela, il a proposé une extension du modèle de BECHTOLD et JACOBS (1990) à un environnement d'opération continue incluant plusieurs pauses. Pour les instances considérées, son modèle identifie plus rapidement une solution optimale que celui de BECHTOLD et JACOBS (1990). AYKIN (2000) a rapporté que même si le modèle de BECHTOLD et JACOBS (1990) est plus compact, la matrice des contraintes du modèle d'AYKIN (1996) est nettement moins dense. Il s'est alors demandé si la supériorité de son modèle n'était pas justement due à cette différence considérable entre les densités.

THOMPSON (1995) a proposé un modèle "doublement" implicite dans lequel les quarts et les pauses sont représentés implicitement. Son modèle s'inspire des travaux de MOONDRA (1976), pour la représentation des quarts, et de BECHTOLD et JACOBS (1990), pour le placement des pauses. Il définit un type de quart par son coût, ses durées minimales et maximales, la durée de la pause de repas, et les durées minimale et maximale des périodes de travail consécutives avant et après la pause.

Un premier ensemble de contraintes égalisant le nombre de débuts et de fins de quarts est imposé pour chaque type de quart. Un deuxième ensemble de contraintes permet de respecter les restrictions sur les durées minimale et maximale du quart. Un troisième ensemble est utilisé pour affecter le nombre de pauses souhaitées (une pause par quart dans ce cas). Un dernier ensemble de contraintes fixe les durées minimale et maximale des périodes pré-pause et post-pause. La conversion des quarts implicites optimaux en des quarts explicites se fait à une étape postérieure avec une procédure *first in first out* qui construit des quarts explicites à partir des informations fournies sur les heures de début et de fin des quarts. La même procédure est utilisée pour l'attribution des pauses aux quarts. Cette procédure n'est cependant décrite que par un exemple très simple ne montrant pas les règles suivies dans le choix des quarts.

1.2 Problèmes de construction de cycles de travail

La construction de cycles implique la spécification des jours de travail et de repos pour les employés sur l'horizon de planification. Les séquences générées doivent respecter les règles de la convention collective et satisfaire les besoins de l'organisation. Ces besoins peuvent être exprimés en nombre d'employés requis par jour ou plus spécifiquement en nombre d'employés nécessaires par jour et par type de quart (quart de jour, de soir ou de nuit). Les règles de la convention collective permettent de garantir des horaires de bonne qualité pour les employés en imposant, par exemple, un nombre minimal de jours de repos par semaine, un nombre minimal de jours de travail consécutifs par semaine, un nombre minimal de fins de semaine au repos, etc.

De même que pour le problème de construction de quarts, le problème de construction de cycles peut être formulé avec un modèle de recouvrement généralisé.

Cependant, les problèmes rencontrés dans la réalité possèdent souvent une structure particulière qui peut être exploitée afin d'élaborer des algorithmes adaptés. Ces algorithmes calculent la main-d'œuvre minimale requise à partir de formules bien spécifiques et sont capables de produire un horaire réalisable en une itération. Cette approche, qualifiée de *combinatoire* par EMMONS (1985), a été adoptée par plusieurs chercheurs (BAKER, 1974; BROWNWELL et LOWERRE, 1976; BAKER et MAGAZINE, 1977; BURNS, 1978; BAKER *et al.*, 1979; BURNS et CARTER, 1985).

BROWNWELL et LOWERRE (1976) ont considéré un horizon de planification d'une semaine où la demande journalière est fixe. BAKER et MAGAZINE (1977) ont supposé que la demande peut varier la fin de semaine et prendre une valeur différente de celle requise les autres jours. BAKER *et al.* (1979) ont considéré un environnement de travail où les besoins en employés sont fixes pour chaque jour de la semaine. Chaque employé doit recevoir deux jours de repos par semaine et un certain nombre de fins de semaine au repos sur le nombre total de semaines travaillées. BURNS et CARTER (1985) ont considéré le même problème sauf que, dans leur cas, la demande en main-d'œuvre peut varier d'un jour à l'autre de la semaine. Ils ont établi des formules leur permettant de calculer trois bornes inférieures sur le nombre total d'employés requis. Ils ont démontré que le nombre optimal d'employés est nécessairement égal à la borne ayant la valeur la plus grande parmi ces trois. Les trois bornes en question découlent respectivement de la contrainte sur le nombre de fins de semaine au repos, de la contrainte sur le nombre de jours travaillés par semaine, et des contraintes de demande. L'algorithme proposé produit des horaires optimaux de bonne qualité et attribue autant de fins de semaine de repos que possible.

EMMONS (1985) a considéré un environnement de travail où la demande reste la même durant les jours de la semaine et prend une valeur différente la fin de semaine. L'algorithme proposé génère systématiquement un horaire principal rotatif (*master*

rotation schedule) utilisé périodiquement. Cet horaire satisfait les contraintes de demande, de jours de repos et de fins de semaine au repos en utilisant le nombre minimal d'employés. Il assure également l'attribution d'au moins deux et d'au plus cinq jours de travail consécutifs par semaine ainsi que deux jours de repos consécutifs en début de semaine pour chaque employé travaillant la fin de semaine d'avant.

EMMONS et BURNS (1991) ont été les premiers à élargir l'approche de BURNS et CARTER (1985) pour inclure différentes catégories d'employés. Les employés sont classés par ordre croissant par rapport à leurs qualifications de sorte qu'un employé d'une certaine classe est capable de substituer tout employé de classe supérieure (l'inverse est interdit). Des restrictions sur le nombre d'employés travaillant par catégorie sont imposées.

Lorsqu'applicable, l'approche combinatoire présente des avantages significatifs par rapport aux formulations mathématiques en nombres entiers. Elle permet en effet un calcul rapide du nombre d'employés nécessaire donnant ainsi la possibilité de mesurer immédiatement l'impact d'un changement dans les besoins ou dans les règles de travail sur la taille de la main-d'œuvre. Plus encore, la génération d'horaires se fait facilement et rapidement même pour les problèmes de grande taille dont la formulation mathématique nécessiterait un très grand nombre de variables. Ceci étant, la formulation par des programmes mathématiques demeure intéressante dans la mesure où elle incorpore plus facilement des contraintes additionnelles et peut donc être étendue à des problèmes plus généraux.

Dans ce cadre, BARTHOLDI et RATLIFF (1978) ont introduit le concept de *unnetworks*. Ils ont qualifié d'*unnetworks* les modèles dont la matrice des contraintes prend une forme complémentaire de celle de la matrice d'incidence des problèmes de réseaux, c'est-à-dire, les matrices dont les colonnes prennent uniquement deux va-

leurs nulles (plutôt que deux valeurs non nulles). Ils ont remarqué que plusieurs problèmes de construction de cycles formulés avec un modèle de recouvrement donnent des matrices de cette forme (les employés travaillent généralement cinq jours par semaine). BARTHOLDI et RATLIFF (1978) ont exploité cette structure particulière en considérant le problème complémentaire au problème de recouvrement. Ce problème complémentaire a pour objectif de maximiser le nombre d'employés pouvant recevoir deux jours de repos consécutifs durant la semaine.

Plus tard, BARTHOLDI *et al.* (1980) ont affirmé que la plupart des problèmes de construction de cycles bénéficient de la propriété de circularité par rapport aux lignes de la matrice des contraintes. Parmi ces problèmes, l'on retrouve le cas où deux jours de repos sont à attribuer par semaine, et le cas où deux paires de jours de repos consécutifs doivent être affectées toutes les deux semaines avec une des paires qui correspond à une fin de semaine. Les auteurs ont démontré que la matrice circulaire peut être transformée en une matrice ayant des colonnes avec uniquement deux valeurs non nulles (1 ou -1) sauf pour la dernière colonne. Celle-ci peut être traitée en tant que paramètre. La fonction objectif est convexe par morceaux et un algorithme de recherche binaire peut alors être utilisé efficacement pour encadrer la solution optimale.

ALFARES (1998) a proposé un algorithme en deux étapes pour résoudre le problème de construction de cycles dans un environnement cyclique où deux jours de repos par semaine sont affectés. Une formule mathématique simple est d'abord utilisée pour calculer la main-d'œuvre minimale requise. Cette valeur minimale est ensuite incorporée dans un modèle linéaire continu qui garantit une solution optimale entière.

Bechtold (1988) a présenté une autre variante du problème de construction de cycles avec objectifs multiples. L'environnement considéré permet aux employés d'être transférés dans d'autres locaux durant la semaine. Les objectifs consistent à minimiser le temps mort et le nombre de transferts. L'approche proposée fournit, en une première étape, le nombre minimal d'employés requis pour satisfaire la demande agrégée à travers tous les locaux. Cette valeur sert ensuite d'intrant à une série de modèles utilisés pour déterminer les affectations et les transferts nécessaires.

1.3 Problèmes de construction simultanée de cycles et de quarts de travail

Les deux problèmes de construction d'horaires : construction de quarts et construction de cycles ont souvent été traités séparément. La complexité de leur intégration a poussé les chercheurs à étudier différentes approches permettant d'aborder le problème globalement tout en donnant des solutions de bonne qualité.

MORRIS et SHOWALTER (1983) ont considéré un horizon de planification d'une semaine discrétisé en des périodes d'une heure. Chaque employé travaille cinq jours consécutifs par semaine et commence son quart toujours à la même heure à tous les jours du cycle. Les quarts considérés ont tous la même durée et peuvent débuter à n'importe quelle heure de la journée. Chaque quart reçoit une pause d'une heure au milieu. Les séquences de jours de travail et de repos sont énumérées pour la semaine et un type de cycle est défini pour chaque séquence de jours de travail et de repos et pour chaque heure de début de travail possible. Le problème est formulé avec un modèle de recouvrement dans lequel une variable de décision est associée à chaque type de cycle et représente le nombre d'employés qui lui sont affectés.

Pour résoudre le problème, MORRIS et SHOWALTER (1983) proposent une procédure heuristique qui fait partie d'une classe plus vaste d'heuristiques basées sur la programmation linéaire (*Linear programming based heuristic methods*). Les heuristiques basées sur la PL commencent par résoudre la relaxation linéaire du problème en nombres entiers. La solution fractionnaire optimale obtenue est ensuite transformée en une solution entière de différentes façons. La méthode de "transformation" proposée par MORRIS et SHOWALTER (1983) utilise une approche d'arrondi/construction. Tout d'abord, la solution fractionnaire est arrondie à l'entier inférieur pour donner une solution entière. La solution ainsi obtenue n'étant pas nécessairement réalisable, deux méthodes sont proposées pour rétablir la réalisabilité. La première méthode consiste à calculer, pour chaque horaire, la sous-couverture maximale parmi toutes les sous-couvertures associées aux périodes couvertes par l'horaire. Le nombre d'employés affectés à l'horaire ayant la sous-couverture maximale la plus élevée est augmenté de 1. Le nombre d'employés est séquentiellement augmenté jusqu'à ce que toutes les contraintes de demande soient satisfaites. La deuxième méthode choisit d'augmenter le nombre d'employés affectés à l'horaire couvrant des périodes de sorte que la somme des sous-couvertures correspondantes soit maximale. MORRIS et SHOWALTER (1983) ont comparé les solutions fournies par ces heuristiques à la borne inférieure donnée par la relaxation linéaire du problème. Ils ont rapporté que l'écart maximal enregistré pour les 30 instances testées ne dépasse pas quatre employés.

Dans le cadre des heuristiques basées sur la programmation linéaire, BECHTOLD *et al.* (1991) proposent une méthode qui s'inspire de celle rapportée par HENDERSON et BERRY (1976) pour la construction des quarts. La première étape s'appuie sur l'heuristique *maxdif* (HENDERSON et BERRY, 1976) pour sélectionner un sous-ensemble de cycles. La relaxation linéaire du problème restreint est ensuite résolue et les valeurs fractionnaires sont arrondies à l'entier supérieur. Les heuristiques

de recherche locales proposées par HENDERSON et BERRY (1976) sont ensuite appliquées pour améliorer la solution courante. Les auteurs ont également étudié une autre classe d'heuristiques dites *constructives*. Ces heuristiques démarrent avec zéro employé affecté et associent les employés aux horaires itérativement jusqu'à ce que tous les besoins soient satisfaits.

BECHTOLD et SHOWALTER (1987) ont proposé une méthode constructive manuelle dans un environnement d'opération discontinue. Les employés sont supposés travailler cinq jours consécutifs par semaine. Les quarts ont une durée de huit heures avec une heure de pause au milieu. Les jours de travail affectés à un cycle doivent débiter toujours à la même heure. Le problème est formulé avec un modèle de recouvrement classique. La méthode de résolution proposée comporte deux principales étapes : la première étape traite le sous-problème de construction de quarts (sept sous-problèmes au total, un problème pour chaque jour de la semaine). La deuxième étape permet de résoudre des sous-problèmes de construction de cycles. Un sous-problème est considéré pour chaque période et chaque jour d'opération. La composante "jour" spécifie le premier jour de travail de la séquence de jours de travail et de repos associée au cycle et la composante "période" détermine l'heure de début de travail pour chaque jour du cycle. BECHTOLD et SHOWALTER (1987) ont comparé la solution obtenue par leur procédure à la borne inférieure donnée par la relaxation linéaire du problème pour 108 profils de demande. Ils ont rapporté que leur solution excède la borne inférieure de 3,6% au maximum.

La méthode constructive proposée par BECHTOLD et SHOWALTER (1987) vient en fait améliorer la performance d'une autre procédure de résolution en deux étapes utilisée par BAKER (1976). Ce dernier divise également le problème en une série de sous-problèmes de quarts et de cycles. Il s'est cependant limité à la résolution des relaxations linéaires associées à ces sous-problèmes.

BECHTOLD *et al.* (1991) ont étudié les performances de quatre heuristiques basées sur la programmation linéaire : l'heuristique qui arrondit la solution fractionnaire à l'entier supérieur, leur adaptation de l'heuristique de HENDERSON et BERRY (1976) au problème de construction simultanée de cycles et de quarts, l'heuristique proposée par KEITH (1979), et l'heuristique établie par MORRIS et SHOWALTER (1983). Ils ont également inclus dans leur étude comparative cinq heuristiques constructives qui sont des variantes de celles proposées par BUFFA *et al.* (1976), MCGINNIS *et al.* (1978) et BECHTOLD et SHOWALTER (1987). Pour cela, ils ont considéré un environnement opérant 16 heures par jour et sept jours par semaine où la demande en employés varie à toutes les heures. Les séquences de jours de travail et de repos doivent inclure cinq jours de travail, consécutifs ou non. Les quarts s'étendent sur neuf heures avec une pause d'une heure à la cinquième heure (huit quarts au total). L'hypothèse que les jours de travail du cycle commencent toujours à la même heure est maintenue. Les instances testées correspondent à 36 profils de demande théoriques différents avec une demande moyenne de 50 employés. Une première fonction objectif qui minimise le nombre total d'heures travaillées est d'abord considérée. Pour cet objectif, les heuristiques de KEITH (1979), de MORRIS et SHOWALTER (1983) et de BECHTOLD et SHOWALTER (1987) sont supérieures aux autres heuristiques en terme de la qualité de la solution obtenue. Ces trois heuristiques ont ensuite été comparées en se basant sur les pourcentages d'employés recevant deux jours de repos consécutifs. Les auteurs ont rapporté que la procédure de BECHTOLD et SHOWALTER (1987) est supérieure.

EASTON et ROSSIN (1991) ont considéré deux environnements d'opération. Le premier environnement est similaire à celui considéré par BECHTOLD *et al.* (1991). Le deuxième suppose en plus une main-d'œuvre hétérogène, c'est-à-dire, incluant des employés à temps plein et des employés à temps partiel avec des salaires et des coefficients de productivité différents. Les auteurs proposent un modèle de

recouvrement classique auquel est ajouté un ensemble de contraintes limitant la proportion des employés de chaque classe par rapport à la main-d'œuvre totale. Dans le modèle proposé, les contraintes de demande prennent une forme particulière dans la mesure où la productivité des employés est prise en compte. Un ensemble d'horaires admissibles est spécifié pour chaque classe d'employés.

Pour résoudre le problème, EASTON et ROSSIN (1991) utilisent un sous-ensemble restreint d'horaires réalisables. La sélection de ce sous-ensemble repose sur la méthode de génération de colonnes. La même procédure que celle de MORRIS et SHOWALTER (1983) est ensuite utilisée pour résoudre le problème restreint. EASTON et ROSSIN (1991) ont comparé leur méthode à celle de HENDERSON et BERRY (1976), celle de MABERT et WATTS (1982), et à un ensemble d'heuristiques qualifiées de LP énumératives. Les heuristiques LP énumératives appartiennent en fait à la classe des heuristiques basées sur la programmation linéaire mais, contrairement aux procédures de sous-ensemble restreint, elles utilisent l'ensemble complet des horaires réalisables pour produire une solution réelle au problème relaxé (KEITH, 1979; MORRIS et SHOWALTER, 1983; MABERT et SHOWALTER, 1988). Les auteurs ont rapporté que, dans un environnement homogène, même si un ensemble restreint d'horaires est utilisé (4% des 1239 horaires disponibles), l'heuristique proposée donne rapidement des solutions identiques à celles obtenues avec les heuristiques LP énumératives. Pour le cas hétérogène, les solutions trouvées sont meilleures et s'obtiennent plus rapidement.

BECHTOLD et BRUSCO (1994) ont proposé trois nouvelles procédures heuristiques pour la génération de sous-ensembles restreints pour le problème de construction simultanée de cycles et de quarts. Deux de ces procédures, qualifiées de *structurelles*, sont basées sur la matrice de couverture de tous les horaires réalisables (HENDERSON et BERRY, 1976). La troisième heuristique s'appuie sur les besoins en main-d'œuvre

(MABERT et WATTS, 1982). L'environnement d'opération considéré est identique à celui de BECHTOLD *et al.* (1991).

Bien que les méthodes basées sur les sous-ensembles restreints permettent de résoudre des problèmes linéaires potentiellement grands, la qualité des horaires obtenus dépend étroitement du choix de l'ensemble initial.

Selon leur secteur d'activités, certaines organisations opèrent 24 heures sur 24 et sept jours sur sept alors que d'autres œuvrent dans un environnement d'opération discontinu. Le cas continu, augmentant la complexité des problèmes de construction d'horaires, certains chercheurs ont opté pour une formulation discontinue du problème et ce, même dans des environnements continus (BYRNE et POTTS, 1973; SEGAL, 1974; BAILEY, 1985; JARRAH *et al.*, 1994). Une formulation discontinue ne considère que les horaires commençant aux périodes ne permettant pas un chevauchement entre les quarts de travail d'un jour à l'autre de l'horizon de planification. Au contraire, une formulation continue considère l'ensemble de tous les horaires réalisables. BRUSCO et JACOBS (1995) ont étudié l'impact, sur la qualité de la solution, de l'utilisation des formulations continue et discontinue dans un environnement d'opération continue. Deux catégories d'employés sont considérées : des employés à temps plein travaillant neuf heures par jour avec une pause fixe au milieu et cinq jours consécutifs par semaine et des employés à temps partiel travaillant également cinq jours consécutifs par semaine à raison de quatre heures par jour sans pause. Le problème est modélisé à l'aide d'un problème de recouvrement auquel est ajoutée une contrainte limitant la proportion des employés à temps partiel. La formulation discontinue utilise 259 variables entières comparé à 336 variables pour la formulation continue, soit une réduction de 22,9% du nombre de variables. Les auteurs ont démontré que, pour certaines instances, l'utilisation d'une formulation discontinue pour un problème de construction d'horaires continu avec

une main d'oeuvre mixte (employés à temps plein et à temps partiel) résulte en une augmentation considérable du coût de la main-d'oeuvre. La différence de coût entre les formulations continue et discontinue devient d'autant plus importante lorsque la proportion des employés à temps partiel prend de petites valeurs (inférieure à 0,25)

BRUSCO et JACOBS (1995) ont proposé deux méthodes de résolution : la méthode de séparation et d'évaluation progressive avec un critère d'arrêt heuristique (meilleure solution trouvée après 1000 secondes) et une heuristique de recherche locale basée sur un recuit simulé qui s'inspire de celle proposée par BRUSCO et JACOBS (1993). L'heuristique comporte une phase de génération qui construit une solution initiale réalisable suivie d'une phase de recherche locale qui tente d'améliorer la solution courante en examinant les solutions réalisables voisines. La procédure de génération choisit aléatoirement un horaire réalisable. Si l'horaire choisi couvre la période de demande correspondant au déficit maximal, l'horaire est ajouté soit à l'ensemble des horaires "temps-partiel", si un employé à temps-partiel peut lui être affecté sans violation de la contrainte de proportion, soit à l'ensemble des horaires "temps-plein". La phase de recherche élimine itérativement un nombre constant d'employés de la solution courante. Le nombre d'employés affecté à un horaire donné est réduit de un si cette réduction résulte en une sous-couverture, au carré, qui est minimale. Des employés sont ensuite ré-affectés aux horaires pour rétablir la réalisabilité de la solution. Cette ré-affectation suit le même principe utilisé dans la phase de génération sauf que c'est le carré de la sous-couverture maximale qui constitue le critère de choix des horaires. Les auteurs ont prouvé, à partir de résultats expérimentaux, que l'heuristique de recuit simulé proposée donne de bonnes solutions en des temps raisonnables et reste dans la majorité des cas supérieure à la méthode de séparation et d'évaluation progressive.

BRUSCO *et al.* (1995) ont considéré un environnement d'opération continu dans lequel le nombre d'heures de début de quarts est limité à une valeur pré-déterminée. De plus, un temps de séparation minimal est imposé entre les heures de début de travail des employés à temps partiel. Les deux catégories d'employés (employés à temps plein et employés à temps partiel) doivent travailler cinq jours consécutifs par semaine et commencent et finissent leur travail toujours à la même heure durant le cycle. Trois types de variables sont utilisées : une variable entière pour chaque type de cycle et chaque période de temps indiquant le nombre d'employés affectés à ce type de cycle et débutant le travail à la période en question (un type de cycle est défini par le nombre d'heures travaillées par jour) ainsi que deux variables binaires, une pour les employés à temps plein et une pour les employés à temps partiel. Ces variables binaires sont définies pour chaque période de l'horizon de planification pour indiquer si un employé (à temps plein ou à temps partiel) commence son travail à cette période ou pas. Outre les contraintes de demande, le modèle proposé incorpore des contraintes limitant la proportion des employés à temps partiel par rapport aux employés à temps plein, des contraintes limitant le nombre d'heures de début de travail pour chaque catégorie d'employés, ainsi que des contraintes imposant un temps de séparation minimal (exprimé en nombre de périodes) entre les heures de début de travail des employés à temps partiel.

BRUSCO *et al.* (1995) ont appliqué leur modèle aux données de la compagnie aérienne *United Airlines* afin d'améliorer leur système de construction d'horaires. Ils ont proposé une stratégie de résolution en deux étapes. La première étape génère des quarts qui seront utilisés pour la construction des cycles à la deuxième étape. La qualité de la solution finale reste cependant étroitement liée à l'ensemble des quarts choisis. Ces quarts sont générés à la suite de la résolution d'un problème de construction de quarts associé à une journée d'opération *type (composite day)* pour laquelle la demande en employés pour une période est égale à la demande maximale

(pour cette période) à travers les jours de la semaine. La relaxation linéaire du modèle de construction de quarts est résolue avec la méthode de génération de colonnes (EASTON et ROSSIN, 1991) et donne un ensemble adéquat de quarts ainsi que le nombre d'employés affectés à chaque quart qui minimise le coût total de la main-d'œuvre pour la journée *type*. Cette solution est ensuite passée au système de fabrication d'horaires de la compagnie pour avoir une solution de départ réalisable pour le problème de construction simultanée de cycles et de quarts. BRUSCO *et al.* (1995) proposent une heuristique de recherche locale basée sur un recuit simulé pour tenter d'améliorer cette solution. Cette heuristique repose sur le même principe que celui proposé par BRUSCO et JACOBS (1995). Plus tard, JACOBS et BRUSCO (1998) ont examiné huit différentes heuristiques associées aux deux étapes de génération et d'amélioration. Ces heuristiques correspondent à différentes combinaisons de certaines règles de sélection de quarts et d'amélioration de la solution courante.

Dans tous les environnements discutés jusque là, les employés sont supposés débiter le travail toujours à la même heure à tous les jours du cycle. BAILEY (1985) est le premier à représenter implicitement le cas où les heures de début des quarts sont libres de varier d'un jour à l'autre de l'horaire. Le modèle proposé utilise deux types de variables. Une variable est associée à chaque séquence de jours de travail et de repos possible et représente le nombre d'employés qui lui sont affectés. Une deuxième variable est associée à chaque type de quart possible pour chaque jour d'opération de la semaine et indique le nombre d'employés affectés à ce quart pour ce jour. Des contraintes liant les séquences de jours de travail et de repos aux quarts sont incorporées. Le modèle est applicable à un environnement discontinu dans lequel les quarts appartenant à un même horaire ne se chevauchent pas d'un jour à l'autre. Pour le cas continu, et afin d'assurer un minimum de repos pour les employés entre deux jours de travail consécutifs, BAILEY (1985) divise les 24 heures de la journée d'opération en trois intervalles de temps de 12 heures. Les quarts affectés à un horaire

doivent commencer et finir au sein d'un même intervalle. BAILEY (1985) s'est limité à résoudre la relaxation linéaire du modèle et à arrondir la solution fractionnaire à l'entier supérieur. Il a rapporté que, pour les instances considérées, la résolution de la relaxation linéaire produit toujours une solution entière. La solution obtenue ne donnant pas explicitement les heures de début de travail ni les quarts qui sont affectés aux cycles individuels, une procédure heuristique d'affectation des quarts est proposée.

BRUSCO et JACOBS (1993) ont également considéré le cas où les heures de début de travail peuvent varier à travers l'horaire. Les quarts utilisés ont tous la même durée et reçoivent une pause entre la deuxième et la sixième heure. L'environnement considéré opère sept jours semaine à raison de 16 heures par jour. Les employés travaillent cinq jours par semaine (consécutifs ou non). Le problème est formulé avec un modèle de recouvrement et résolu avec une heuristique de recherche locale basée sur un recuit simulé similaire à celle de BRUSCO *et al.* (1995). Les auteurs ont comparé la solution fournie par leur heuristique à la borne inférieure donnée par la résolution de la relaxation linéaire du problème ainsi qu'à la solution donnée par la méthode de séparation et d'évaluation progressive pour 36 instances. Ils ont rapporté que la valeur de l'objectif obtenue excède la borne inférieure de 1,95% au maximum et celle donnée par la méthode de séparation et d'évaluation progressive de 1,88% au maximum. Le temps de calcul requis par l'heuristique est cependant nettement inférieur et représente 10% de celui requis par la méthode de séparation et d'évaluation progressive.

JARRAH *et al.* (1994) ont proposé un modèle implicite de construction d'horaires qui minimise le coût total de la main-d'œuvre. L'environnement considéré opère de façon continue et utilise des employés à temps plein et à temps partiel. Les employés à temps plein travaillent des quarts de huit heures et demi et cinq jours

par semaine. Des quarts de différentes durées peuvent par contre être attribués aux employés à temps partiel. Chaque quart dépassant une certaine durée reçoit une pause d'une demi-heure à l'intérieur d'une fenêtre de temps. L'extra-chevauchement, tel que défini par BECHTOLD et JACOBS (1990), est interdit entre ces intervalles de pauses. De plus, seuls les quarts qui ne se chevauchent pas d'un jour au jour suivant sont considérés (formulation discontinue du problème).

Le modèle proposé par JARRAH *et al.* (1994) s'inspire de celui de BECHTOLD et JACOBS (1990), pour le problème de construction de quarts, et de celui de BURNS et CARTER (1985), pour le problème de construction de cycles. Un type de quart est défini par une heure de début, une durée et un intervalle de pauses (BECHTOLD et JACOBS, 1990). Quatre types de variables sont utilisés. Une première variable est associée à chaque type de quart et à chaque jour de l'horizon pour représenter le nombre d'employés à temps plein affectés à ce type de quart et à ce jour. Une variable similaire est associée à chaque type de quart et à chaque jour pour représenter le nombre d'employés à temps partiel qui lui sont affectés. Une troisième variable est associée à chaque période et à chaque jour de l'horizon pour représenter le nombre de pauses accordées correspondantes. Finalement une variable est utilisée pour compter le nombre d'employés à temps plein nécessaire. Les contraintes *forward* et *backward* de BECHTOLD et JACOBS (1990) sont incorporées pour lier les variables de pauses aux deux types de variables de quart. Une autre contrainte restreint le nombre d'employés à temps partiel par rapport aux employés à temps plein. Le modèle incorpore également deux contraintes qui bornent inférieurement le nombre total d'employés à temps plein requis. Ces bornes inférieures sont calculées à partir des formules proposées par BURNS et CARTER (1985).

La méthode de résolution proposée par JARRAH *et al.* (1994) est basée sur une approche de décomposition combinée à une procédure de séparation et d'évaluation

progressive. La résolution du modèle implicite ne donnant pas les horaires explicites, deux procédures d'affectation sont appliquées a posteriori. La première procédure affecte les pauses aux quarts tout en respectant les fenêtres de temps. Il s'agit dans ce cas de déterminer une solution réalisable pour un problème de transport approprié associé à chaque jour de la semaine. La deuxième procédure affecte les quarts aux cycles en essayant de réduire au maximum les écarts entre les heures de début de travail au sein de l'horaire. Pour cela, tous les quarts des employés à temps plein sont classés par ordre croissant par rapport à leurs heures de début, et ce pour chaque jour de la semaine. Un algorithme de construction de séquences de jours de travail et de repos, similaire à celui de BURNS et CARTER (1985), est ensuite utilisé pour générer un nombre fixé d'horaires incluant pour chaque jour le quart débutant le plus tôt possible. Parmi les horaires obtenus, l'horaire pour lequel la variance des heures de début est maximale est sélectionné pour être amélioré. Si, parmi tous les autres horaires obtenus, il existe un horaire contenant un quart qui débute plus tôt que l'heure la plus tardive à laquelle commencent les quarts relatifs à l'horaire choisi et un quart qui débute plus tard que l'heure la plus tôt à laquelle commencent les quarts relatifs à l'horaire choisi, les échanges de quarts possibles entre ces deux horaires et donnant une meilleure solution sont examinés. Si un tel échange n'existe pas, aucune amélioration n'est possible (par rapport à l'écart entre les heures de début de travail à tous les jours du cycle).

Pour tous les environnements qui tolèrent des heures de début variables entre deux jours de travail consécutifs, les procédures de construction d'horaires explicites proposées tentent toujours de réduire la fluctuation entre ces heures de début afin de garantir, entre autres, un temps de repos minimal entre les jours de travail. JACOBS et BRUSCO (1996) ont, de ce fait, introduit la notion de bandes d'heures de début qui permet une variation contrôlée des heures de début de travail à travers l'horaire. Les heures de début des quarts affectés à un horaire donné doivent ainsi

appartenir au même ensemble d'heures de début à tous les jours du cycle. JACOBS et BRUSCO (1996) traitent le cas où deux bandes d'heures de début associées à deux horaires différents peuvent se chevaucher. Un modèle implicite utilisant deux types de variables entières est proposé. Une variable entière est associée à chaque période de chaque bande d'heures de début et à chaque jour de l'horizon. Elle représente le nombre d'employés affecté à un quart débutant à cette période pour ce jour et pour cette bande d'heures de début. Une deuxième variable entière est associée à chaque séquence de jours de travail et de repos énumérée et à chaque bande d'heures de début pour désigner le nombre d'employés qui lui sont affectés. Un ensemble de contraintes d'égalité est incorporé pour lier convenablement les quarts affectés à une bande aux séquences de jours de travail et de repos correspondant à cette bande.

JACOBS et BRUSCO (1996) ont appliqué la méthode classique de séparation et d'évaluation progressive avec un nombre limité d'itérations (100 000). La solution fournie par le modèle ne donnant pas des horaires globaux explicites (mais plutôt des affectations de quarts et de cycles aux bandes d'heures de début), un post-processeur est utilisé pour compléter les horaires. Le post-processeur opère sur chaque bande individuellement et construit les horaires en liant les quarts aux cycles. Les auteurs ont comparé leur modèle à un modèle de recouvrement équivalent. Ils ont rapporté que dans le cas où la largeur des bandes d'heures de début est supérieure ou égale à deux (c'est-à-dire, deux heures de début possibles dans la bande), leur modèle requiert considérablement moins de variables que le modèle de recouvrement contre une légère augmentation du nombre de contraintes. Par contre, une largeur de bande de un (donc, aucune variation n'est tolérée entre les heures de début) induit des modèles de recouvrement de taille relativement plus petite que les modèles implicites. Pour les instances offrant de la flexibilité en terme de bandes d'heures de début, le modèle implicite atteint, dans la majorité des cas, la solution optimale en des temps raisonnables.

BRUSCO et JACOBS (2000) ont proposé un modèle implicite pour un problème de construction simultanée de cycles et de quarts de travail qui tient compte de plusieurs formes de flexibilité en terme de placement des pauses (*break window*), d'heures de début de quarts et de bandes d'heures de début. Leur modèle s'inspire du travail de BECHTOLD et JACOBS (1990) pour représenter implicitement l'attribution des pauses (*forward et backward*), de celui de JACOBS et BRUSCO (1998) pour modéliser le temps de séparation entre les heures de début de quarts, et de celui de JACOBS et BRUSCO (1996) pour les bandes d'heures de début. BECHTOLD et JACOBS (1990) avaient traité le cas discontinu uniquement. BRUSCO et JACOBS (2000) proposent une extension de leur modèle au cas continu et introduisent des variables de pauses *chevauchantes* (*wraparound break variables*). Des formules mathématiques calculant le nombre de variables et de contraintes nécessaires à leur modèle implicite et au modèle de recouvrement équivalent sont établies. Ces formules montrent la supériorité de leur modèle par rapport au modèle de recouvrement pour des environnements offrant une grande flexibilité. Une procédure de séparation et d'évaluation progressive est utilisée pour résoudre le problème.

ÇEZIK *et al.* (2001) ont considéré le problème de construction simultanée de cycles et de quarts dans un environnement d'opération discontinue ayant pour objectif de minimiser le coût total de la main-d'œuvre et les pénalités de sous-couvertures. Les horaires construits doivent répondre à plusieurs des règles discutées plus haut (durées des quarts, durées et positions des pauses, surcharge de travail, type de séquences de jours de travail et de repos, nombre de fins de semaine au repos, etc). Une variation contrôlée des heures de début de travail à travers le cycle est assurée, non pas en considérant des bandes d'heures de début qui restent fixes pour tous les jours de travail de l'horaire, mais en tolérant un décalage maximal entre les heures de début de quarts affectés à deux jours de travail consécutifs. Les auteurs supposent que les horaires fabriqués sont cycliques de sorte que la contrainte de décalage maximal

doit être satisfaite entre le dernier jour et le premier jour du cycle, si ces jours correspondent à des jours de travail.

Le modèle proposé par ÇEZİK *et al.* (2001) utilise sept problèmes de construction de quarts, un pour chaque jour de la semaine, et un problème de réseaux pour construire les horaires finaux. Les auteurs rapportent que plusieurs des formulations proposées dans la littérature peuvent être utilisées pour modéliser les problèmes de construction de quarts. Pour les instances testées, un modèle très semblable à celui de THOMPSON (1995) est adopté. Une méthode de fixation et de séparation (*fix-and-branch*) est implantée pour résoudre le problème. La relaxation linéaire du modèle est d'abord résolue, plusieurs des valeurs prises par les variables sont fixées heuristiquement et le modèle avec variables fixées est résolu avec la méthode de séparation et d'évaluation progressive avec un temps limite de cinq minutes. Des solutions de bonne qualité (dépassant la borne inférieure de la relaxation linéaire d'au plus deux unités) sont obtenues en des temps relativement courts (151 secondes en moyennes pour 15 instances).

TOPALUĞLUA et OZKARAHAN (2002) ont proposé un modèle implicite qui combine le travail d'AYKIN (1996) et celui de BAILEY (1985) pour le problème de construction simultanée de cycles et de quarts avec pauses. Le modèle proposé s'applique au cas discontinu et impose que les quarts affectés à un même horaire soient de même durée. TOPALUĞLUA et OZKARAHAN (2002) exposent le cas où une seule pause est attribuée par quart et rapportent que leur modèle s'étend facilement au cas où plusieurs pauses de différentes durées sont à affecter. Les auteurs utilisent des ensembles qui définissent les types de quarts admissibles pour chaque jour, les types de séquences de jours de travail et de repos associées aux quarts d'une certaine durée, ainsi qu'un ensemble d'heures de début possibles pour les types de quarts pour chaque jour de l'horizon. La définition de ces ensembles permet ainsi d'adapter

facilement le modèle proposé à des environnements très flexibles où le nombre de jours travaillés par cycle peut varier d'un employé à un autre de même que le type de séquences de jours de travail et de repos, etc. Les auteurs ont également démontré la supériorité de leur formulation implicite au modèle de recouvrement équivalent en termes de taille et de temps de résolution.

BARD *et al.* (2003) ont considéré un environnement d'opération continue dans lequel les employés à temps plein et à temps partiel travaillent cinq jours par semaine, débutant toujours le travail à la même heure. Les employés à temps plein sont affectés à des quarts d'une durée fixe alors que la durée de travail des employés à temps partiel peut varier d'un jour à l'autre. De plus, dès qu'un quart dépasse une certaine durée (six heures), une pause de 30 minutes lui est attribuée dans un intervalle de pauses approprié. Un modèle implicite en nombres entiers est proposé. Ce modèle utilise les contraintes introduites par BURNS et CARTER (1985) pour borner inférieurement la taille de la main-d'œuvre et les contraintes *forward* et *backward* de BECHTOLD et JACOBS (1990) pour l'affectation des pauses aux quarts. Le problème est résolu par une méthode de séparation et d'évaluation progressive. La solution obtenue donne le nombre de quarts de chaque type requis pour chaque jour ainsi que le nombre de pauses débutant à chaque période de la journée, sans spécifier les jours de repos ni les pauses associées à chaque quart. Pour cela, les auteurs proposent deux algorithmes d'affectation : le premier algorithme utilise une version modifiée de la procédure de BURNS et CARTER (1985) pour affecter deux jours de repos et le deuxième algorithme résout un problème de transport pour affecter les pauses aux quarts. BARD *et al.* (2003) proposent plusieurs extensions de leur modèle, entre autres, au cas où les deux jours de repos doivent être consécutifs et au cas où les heures de début de travail peuvent varier librement d'un jour à l'autre au sein d'un même horaire.

1.4 Impact des formes de flexibilité

En pratique, les environnements d'opération incorporent un certain degré de flexibilité qui leur procure une marge de manœuvre pour répondre aux besoins de l'organisation, d'une part, et aux attentes des employés, d'autre part. Plusieurs chercheurs ont étudié l'impact de quelques unes des formes de flexibilité sur le coût de la main-d'œuvre et la qualité des horaires obtenus.

D'après l'étude menée par BECHTOLD et JACOBS (1993) et pour les instances considérées, tolérer la flexibilité dans le placement des pauses, accroître le nombre d'heures de début des quarts, et utiliser des quarts de différentes durées résultent en une diminution significative de la taille de la main-d'œuvre. Par contre, permettre aux employés de débiter leur travail à différentes heures à tous les jours du cycle n'implique aucune amélioration par rapport aux heures de début fixes. Plus encore, imposer que les deux jours de repos soient consécutifs n'affecte pas non plus le nombre d'employés requis. Cependant, certains des résultats obtenus par BECHTOLD et JACOBS (1993) restent très dépendants des instances considérées.

À cet égard, BARD *et al.* (2003) ont obtenu des résultats différents quant à l'impact des deux jours de repos consécutifs sur la taille et le coût de la main-d'œuvre. Pour les instances testées, imposer que les deux jours de repos soient consécutifs augmente le nombre total d'employés à temps plein de 2% en moyenne (sept employés sur 101) et celui des employés à temps partiel de 8% (2 employés sur 25), résultant en une augmentation du coût hebdomadaire de 7,6%.

Dans un autre contexte, JARRAH *et al.* (1994) ont rapporté que des pertes relativement peu significatives sont encourues lorsque l'on impose des heures de début de travail fixes durant le cycle. Ils en ont conclu qu'il serait peut-être plus

souhaitable d'utiliser des heures de début fixes et de garantir des horaires de bonne qualité pour les employés. Ceux-ci préfèrent généralement avoir le même quart à chaque jour de leur cycle. Plus tard, BARD *et al.* (2003) ont abouti aux mêmes résultats. Dans leur cas, une variation des heures de début de travail dans un intervalle de huit heures n'avait aucun effet sur la taille totale de la main-d'œuvre et permettait de sauver uniquement 480 dollars par semaine.

JARRAH *et al.* (1994) ont aussi étudié l'impact de la longueur des cycles sur le coût de la main-d'œuvre. Ils ont rapporté qu'une semaine de travail de quatre jours réduit considérablement ces coûts (réduction annuelle de 1,3 millions de dollars) lorsque comparée à une semaine de travail de cinq jours. L'environnement considéré par JARRAH *et al.* (1994) utilise des employés à temps partiel et des employés à temps plein. Les auteurs ont constaté qu'une diminution de la valeur de ce ratio (30 employés à temps partiel au maximum pour 70 employés à temps plein) entraîne des gains significatifs pouvant atteindre les 1,9 millions de dollars par an. Beaucoup d'autres chercheurs ont également étudié l'impact de l'utilisation des employés à temps partiel (MABERT et SHOWALTER 1990, BRUSCO et JACOBS 1993, BARD *et al.* 2003).

Contrairement aux résultats obtenus par BECHTOLD et JACOBS (1993) et JARRAH *et al.* (1994), JACOBS et BRUSCO (1998) ont rapporté le gain en main-d'œuvre découlant de la variation des heures de début de travail durant le cycle. Ils ont constaté que plus la largeur de la bande d'heures de début est grande plus la taille de la main-d'œuvre est réduite. Par exemple, augmenter la largeur de la bande d'une à deux heures permet de sauver un employé sur 18. Considérer trois heures de début de travail par bande permet d'en sauver encore un. Cependant, il existe une largeur de bande limite au delà de laquelle aucune amélioration n'est enregistrée.

Au lieu d'utiliser le concept de bandes d'heures de début, ÇEZİK *et al.* (2001) ont défini ce que nous avons appelé à la section 1.3 le décalage entre les heures de début de travail. Ils ont comparé le nombre d'employés requis dans deux environnements : le premier environnement ne permet aucun décalage entre les heures de début de travail associées à deux jours consécutifs alors que le deuxième environnement tolère un décalage de trois heures. Un excès d'à peu près 2% d'employés en moyenne est enregistré pour le premier environnement. Les auteurs ont rapporté que, lorsque d'autres courbes de demande générées aléatoirement sont considérées, cette différence pouvait atteindre 25%.

D'autre part, ÇEZİK *et al.* (2001) ont rapporté que l'utilisation des heures supplémentaires pouvait résulter en une réduction atteignant 7% du coût total de la main-d'œuvre. Cet avantage considérable découlant de l'incorporation de différentes longueurs de quarts a également été souligné par TOPALOGLUA et OZKARAHAN (2002). Ces derniers ont comparé le cas où une seule durée de quart (8 heures) était considérée au cas où les quarts pouvaient avoir trois durées différentes (8, 10 et 12 heures). Le nombre d'employés requis a été réduit de presque 20% en moyenne à la suite de l'utilisation de trois longueurs de quarts. Plus encore, la variété d'environnements considérée par TOPALOGLUA et OZKARAHAN (2002) permet de confirmer l'impact des largeurs des fenêtres de pauses sur la taille de la main-d'œuvre. Un gain de 17% en moyenne est obtenu lorsque la largeur de ces fenêtres passe d'une heure à deux heures pour l'environnement n'incorporant que des quarts de huit heures. Ce gain est de 7% pour l'environnement avec trois longueurs de quarts.

AYKIN (1996) a étudié l'influence de la flexibilité dans le placement des pauses dans le cas où plusieurs pauses sont attribuées au quart. Pour les instances testées, chaque quart doit recevoir trois pauses, une pause de repas de 30 minutes au milieu et deux

pauses de 15 minutes avant et après le repas. AYKIN (1996) a comparé le nombre d'employés requis dans un environnement qui tolère cinq heures de début possibles pour la première et la troisième pause et dans un environnement moins flexible qui fixe les heures de début de ces deux pauses. La comparaison est faite pour différentes largeurs de la fenêtre de temps associée à la pause de repas. Un gain de 4,1% du nombre d'employés (six employés au total) est enregistré dans le cas où six heures de début sont possibles pour la pause de repas. Ce gain atteint les 8,9% (14 employés au total) lorsque la pause de repas commence à une heure fixe. AYKIN (1996) a également comparé la taille de la main-d'œuvre nécessaire dans les cas où la largeur de l'intervalle associé à la pause de repas prend les valeurs un (pause fixe) et cinq, et ce pour différentes largeurs des fenêtres de temps correspondant aux première et troisième pauses. AYKIN (1996) aboutit à la même conclusion : plus les intervalles sont larges plus le nombre d'employés requis est réduit.

BRUSCO et JACOBS (2000) ont étudié l'influence de la largeur des fenêtres de pauses, des bandes d'heures de début et du temps de séparation minimal entre les heures de début de quarts sur la taille de la main-d'œuvre. Ils ont considéré plusieurs combinaisons possibles de ces paramètres. Ceci leur a permis de voir que ces paramètres restent en fait étroitement liés et qu'il faut déterminer les valeurs optimales de ces paramètres pris ensemble et non chacun à part. Parmi les huit combinaisons obtenues en prenant des fenêtres de pauses de largeur un (pause fixe) et quatre, des bandes d'heures de début de largeur un et deux, et un temps de séparation minimal de trois et de quatre périodes, c'est la combinaison (4,2,3) qui a donné la main-d'œuvre la plus petite pour toutes les instances. Ceci étant, les combinaisons (1,1,4) et (1,2,4) ont donné de meilleurs résultats que les combinaisons (1,1,3) et (1,2,3).

Ceci vient renforcer l'idée que ce n'est pas seulement le nombre d'heures de début qui est important mais également le choix d'un ensemble adéquat de ces heures. Ce résultat a déjà été constaté par JACOBS et BRUSCO (1998) et confirmé par BRUSCO et JACOBS (2001). Ces derniers ont, de plus, démontré, via une grande étude expérimentale, qu'un petit nombre d'heures de début de quarts permettait de produire une taille de main-d'œuvre minimale, pourvu que le bon ensemble soit choisi.

CHAPITRE 2

ORGANISATION DE LA THÈSE

Ce travail de recherche aborde certains aspects du problème de construction d'horaires anonymes. Nous nous intéressons particulièrement aux problèmes de construction de quarts de travail et aux problèmes de construction simultanée de cycles et de quarts de travail dans des environnements d'opération incorporant un haut degré de flexibilité. De tels problèmes induisent généralement des modèles de très grande taille qui sont difficiles à résoudre à l'optimalité. Le premier objectif de cette thèse est de proposer des modèles implicites généraux qui captent toutes les difficultés liées à l'incorporation des formes de flexibilité. Ces modèles doivent néanmoins garantir de bonnes solutions en des temps de calcul raisonnables. Un deuxième objectif, étroitement lié au premier, consiste à améliorer ces formulations en essayant de réduire le nombre de variables, le nombre de contraintes, et la densité de la matrice des contraintes. Ceci passe par une description rigoureuse des contextes dans lesquels l'une ou l'autre des réductions est possible. Le troisième objectif de cette thèse est de modéliser de nouvelles formes de flexibilité présentes dans la vie réelle et qui ont été, à notre connaissance, peu ou jamais étudiées dans la littérature. Il s'agit également d'étudier l'impact de ces nouvelles formes sur le coût total de la main-d'œuvre. Le dernier objectif consiste à proposer et à comparer certaines approches de résolution pour des instances réelles de grande taille fournies par une agence de contrôleurs aériens.

Le troisième chapitre présente un modèle implicite général basé sur des problèmes de transport pour le problème de construction simultanée de cycles et de quarts

de travail. L'environnement d'opération considéré incorpore beaucoup de flexibilité, notamment, les fenêtres de pauses et les bandes d'heures de début. Une reformulation de Benders est appliquée à ce modèle et les coupes de réalisabilité correspondantes sont déterminées. Une preuve mathématique permet, entre autres, de démontrer que les contraintes dites *forward* et *backward*, initialement introduites pour la modélisation du placement flexible des pauses au sein des quarts, représentent des coupes de réalisabilité de Benders. Sous certaines hypothèses, nous prouvons que ces contraintes, relativement peu nombreuses, dominent toutes les autres coupes de réalisabilité de Benders. Elles sont dès lors rajoutées au problème maître et le modèle implicite résultant contient beaucoup moins de variables et de contraintes que le modèle initial. Nous prouvons également que les relaxations linéaires des deux modèles implicites les plus utilisés dans la littérature, à savoir celui de BECHTOLD et JACOBS (1990) et celui d'AYKIN (1996), sont équivalentes. Ce résultat implique que les modèles en nombres entiers correspondants induisent le même saut d'intégrité. Finalement, nous comparons notre approche implicite pour modéliser les bandes d'heures de début à la seule approche utilisée jusqu'à date dans la littérature. Nous montrons que notre approche incorpore beaucoup moins de variables contre une légère augmentation du nombre de contraintes.

Le quatrième chapitre traite un problème de construction de quarts de travail très flexible découlant de la vie réelle. Nous introduisons une nouvelle forme de flexibilité dans la définition des pauses : la *pause décomposable*. Nous étendons également la notion de *restrictions sur la durée de travail consécutif* au contexte multi-pauses. Deux modèles implicites sont proposés et comparés. Ces modèles utilisent des contraintes *forward* et *backward* généralisées et reformulées de manière à réduire la densité de la matrice des contraintes. Nous montrons, à travers des résultats expérimentaux, que la réduction de la densité des modèles améliore considérablement les temps de calcul. Ce chapitre étudie également l'impact des pauses décomposables

sur le nombre d'employés requis. Des gains en main-d'œuvre dépassant les 8% sont obtenus. Plus encore, les restrictions sur la durée de travail consécutif avant et après chaque pause constituent en fait une autre approche pour placer les pauses au sein du quart. Nous comparons cette approche à une autre couramment utilisée dans la littérature et nous montrons qu'elle induit un nombre moins élevé d'employés. La fin du chapitre montre les mérites de la modélisation implicite des restrictions sur la durée de travail consécutif par rapport à une modélisation explicite.

Le dernier chapitre propose trois approches de résolution pour des problèmes de construction de quarts complexes sur des horizons de planification relativement longs. La première méthode utilise le concept de branchement local introduit par FISCHETTI et LODI (2003) pour résoudre des problèmes en nombres entiers relativement complexes. Les coupes de branchement local que nous proposons s'inspirent de celles introduites par ces auteurs mais sont mieux adaptées aux problèmes de construction de quarts de travail. La deuxième méthode de résolution proposée est basée sur une décomposition temporelle de l'horizon de planification en des fenêtres de temps. Plusieurs sous-problèmes réduits de construction de quarts de travail sont ainsi résolus, un sous-problème pour chaque fenêtre de temps. Afin de tenir compte du caractère continu des environnements d'opération, les fenêtres de temps utilisées se chevauchent sur un ensemble de périodes défini dynamiquement au cours du processus de décomposition. La troisième approche correspond à la méthode de séparation et d'évaluation progressive de CPLEX. Chacune de ces trois approches est appliquée à deux formulations implicites proposées pour modéliser le problème. Ces formulations étendent en fait les modèles implicites proposés au quatrième chapitre pour une seule journée d'opération à un horizon de planification de plusieurs jours d'opération. Les trois approches de résolution sont comparées à travers des instances réelles fournies par une agence de contrôleurs aériens. Les résultats obtenus montrent que le branchement local que nous proposons donne de

meilleures solutions pour les instances relativement faciles. Pour les instances plus difficiles, la décomposition par fenêtres de temps est meilleure : elle identifie de bonnes solutions en des temps relativement courts.

CHAPITRE 3

USING BENDERS DECOMPOSITION TO IMPLICITLY MODEL TOUR SCHEDULING

Monia Rekik, Jean-François Cordeau, et François Soumis, *Annals of Operations Research* 128, pages 111–133, 2004.

L'approche de modélisation implicite se présente comme l'un des moyens les plus efficaces pour formuler des problèmes de construction d'horaires complexes. Dans cet article, nous proposons des modèles implicites pour le problème de construction simultanée de cycles et de quarts de travail avec pauses et bandes d'heures de début. En développant ces modèles, nous avons essayé de tenir compte de toutes les formes de flexibilité considérées jusqu'à présent dans la littérature de façon à faciliter l'adaptation des modèles à une grande variété d'environnements d'opération.

La représentation implicite de certaines données du problème utilise des contraintes *forward* et *backward* adaptées. Ces contraintes ont été initialement introduites par BECHTOLD and JACOBS (1990) pour l'attribution des pauses aux quarts. Afin de comprendre l'origine de telles contraintes et de déterminer les conditions nécessaires à leur application, nous nous sommes intéressés en un premier temps à un problème de construction de quarts similaire à celui de BECHTOLD and JACOBS (1990) et pour lequel ces contraintes ont été initialement considérées.

Nous proposons deux modèles pour le problème de construction de quarts. Le premier modèle est basé sur un problème de transport pour affecter les pauses aux quarts dans les intervalles appropriés. Le deuxième modèle utilise, à la place,

des contraintes *forward* et *backward*. Une reformulation de Benders du premier modèle est proposée et les coupes de réalisabilité correspondantes sont déterminées. En fait, cette reformulation de Benders utilise un sous-problème qui correspond à un problème de transport pur avec des variables d'affectation. Ces variables n'intervenant pas dans la fonction objectif du problème maître, seules les coupes de réalisabilité sont à déterminer (aucune coupe d'optimalité). Pour voir l'aspect général des coupes de réalisabilité, nous considérons le dual de la relaxation linéaire du sous-problème de transport. Nous cherchons ensuite l'ensemble minimal qui contient tous les rayons extrêmes du sous-problème dual et nous montrons que les contraintes *forward* et *backward* découlent de rayons extrêmes particuliers. La preuve est ainsi faite que ces contraintes représentent des coupes de réalisabilité de Benders. Nous prouvons enfin que, sous les hypothèses d'absence d'entraçage (BECHTOLD and JACOBS, 1990) et de continuité des fenêtres de pauses, les contraintes *forward* et *backward* dominent toutes les autres coupes de réalisabilité relatives aux autres rayons extrêmes.

La preuve que nous avons établie montre également que les relaxations linéaires des deux modèles proposés sont équivalentes. Plus encore, nous avons montré que le modèle d'AYKIN (1996) correspond en fait à notre premier modèle (basé sur le problème de transport) à quelques petites modifications près. Dès lors, nous avons pu affirmer que les formulations de BECHTOLD and JACOBS (1990) et d'AYKIN (1996) sont équivalentes et induisent le même saut d'intégrité.

Nous nous sommes ensuite intéressés au problème de construction simultanée de cycles et de quarts de travail. Après avoir défini le problème, nous avons proposé, de la même façon que pour le problème de construction de quarts, un premier modèle basé sur des problèmes de transport particuliers et un modèle équivalent mais plus compact qui utilise des contraintes *forward* et *backward* adaptées. Outre l'affectation des pauses aux quarts, des sous-problèmes de transport sont définis

pour l'affectation des quarts aux cycles. En respectant certaines conditions (absence d'extra-chevauchement et continuité des bandes d'heures de début), chacun de ces problèmes de transport possède la structure particulière considérée dans la preuve pour être modélisé par des contraintes *forward* et *backward* convenablement définies.

Des extensions du modèle compact sont ensuite proposées au cas où plusieurs pauses sont attribuées à chaque quart et au cas où l'extra-chevauchement est présent.

Enfin, nous avons comparé la taille de notre modèle implicite à une extension du modèle proposé par BRUSCO and JACOBS (2000) au cas où plusieurs durées de quarts et plusieurs formes de séquences de jours de travail et de repos sont considérées. Nous avons établi des formules mathématiques qui calculent le nombre de variables et de contraintes nécessaires pour chaque modèle pour différentes instances. Les résultats obtenus montrent que notre modèle implicite utilise beaucoup moins de variables que celui de BRUSCO and JACOBS (2000) et légèrement plus de contraintes.

Cet article contribue de plusieurs façons au domaine de la modélisation implicite. D'abord, en comprenant l'origine des contraintes *forward* et *backward*, qui constituent un des piliers de la modélisation implicite, nous avons pu décrire la structure particulière dont doit bénéficier le problème auquel ces contraintes peuvent s'appliquer. Ceci permet entre autres d'utiliser ces contraintes pour modéliser d'autres formes de flexibilité telle que la variation des heures de début de travail à travers le cycle. De plus, nous avons confirmé les mérites de l'incorporation de ce type de contraintes dans la réduction de la taille des modèles par rapport à d'autres approches. Avec la preuve que nous avons développée, nous avons montré que les relaxations linéaires de deux approches classiques de modélisation implicite sont équivalentes et que les modèles en nombres entiers correspondants donnent le même saut d'intégrité. Finalement, le modèle implicite proposé permet de tenir compte d'un très haut degré de flexibilité et peut s'adapter à une grande variété de problèmes

de construction d'horaires. Dans le cas où les contraintes *forward* et *backward* ne sont plus suffisantes ou que leur nombre devient relativement grand, une décomposition de Benders du problème peut être appliquée. Nous avons, pour cela, décrit la forme générale de toutes les coupes de réalisabilité correspondantes.

Using Benders Decomposition to Implicitly Model Tour Scheduling

MONIA REKIK

École Polytechnique de Montréal

JEAN-FRANÇOIS CORDEAU

HEC Montréal

FRANÇOIS SOUMIS

École Polytechnique de Montréal

April 2004

Abstract

This paper introduces a new integrated model for the combined days-off and shift scheduling problem (the tour scheduling problem). This model generalizes the *forward* and *backward* constraints, previously introduced by Bechtold and Jacobs for the shift scheduling problem, to the tour scheduling problem. This results in a general and compact formulation that can handle several types of scheduling flexibility. We also provide a new proof of the correctness of forward and backward constraints based on Benders decomposition. The latter approach is interesting in itself because it can be used to solve the problem when extraordinary overlap of break windows or start-time bands is present. A discussion of model size for a set of hypothetical test problems is presented to show the merits of the new formulation.

3.1 Introduction

In the last two decades, personnel scheduling problems have received a lot of attention in the operations research literature. Several real-life applications of mathematical programming techniques (BLAIS et al., 1990; BRUSCO et al., 1995; HUETER and SWART, 1998) have also demonstrated that improved personnel scheduling may provide significant productivity gains and labor savings.

Personnel scheduling problems can be classified, as first suggested by BAKER (1976), into three categories: days-off, shift, and tour scheduling problems. *Days-off scheduling problems* deal with the assignment of work and rest days to employees over a given planning horizon. *Shift scheduling problems* arise when one must specify, for a given day, the starting time and duration of shifts assigned to employees and, possibly, the position of rest and meal breaks within each shift. Finally, when both daily shift and days-off schedules should be determined simultaneously, *tours* are developed over the planning horizon. This problem is called the *tour scheduling problem* and is the object of this paper.

Generally, the complexity of the scheduling task depends on two factors: the continuous or discontinuous nature of the scheduling environment and the level of flexibility considered in creating the schedules.

Continuous problems arise in organizations that operate 24 hours a day and 7 days a week. These organizations face demand for labor on a continuous basis and often allow their schedules to begin at any hour during the day. Shifts are thus allowed to overlap from one day to the next. Computational studies (BARTHOLDI, 1981; BRUSCO and JACOBS, 1995) have shown that continuous models are difficult to

solve optimally. Some problems of this type have also been shown to be NP-complete (BARTHOLDI, 1981).

Scheduling flexibility may concern shift, days-off and tour scheduling. Shift scheduling flexibility includes multiple shift-lengths (i.e., shifts can be of different durations), break-placement flexibility which enables a break to start within a time window rather than at a fixed time, and shift starting time flexibility: shifts may often start at any time during the day (given a certain discretization of the planning horizon) but they may also be restricted by rules that impose a maximum number of starting times or a minimum time separation between them (SCHINDLER and SEMMEL, 1993; JACOBS and BRUSCO, 1998). The provision of unrestricted shift starting times is often coupled with a start-time float flexibility that enables the construction of tours that include shifts starting at a different time on each day of the tour. The concept of start-time band is then used to limit the variation of starting times within an individual tour (JACOBS and BRUSCO, 1996). Similarly, a shift-length float enables the assignment of work shifts of different lengths within an individual tour while a break-placement float allows the assignment of breaks that can be placed at different periods of the day from one day to the next within an individual tour. Days-off scheduling flexibility alternatives include different numbers of consecutive or non-consecutive work days over the planning horizon, different possibilities of rest day placement in the planning horizon, or, more generally, different models of days-on patterns (e.g., two days off each week, two consecutive days off each week). Although the incorporation of scheduling flexibility may considerably complicate the problem, it often has a very positive impact on labor utilization (MABERT and WATTS, 1982; MABERT and SHOWALTER, 1990; BECHTOLD and JACOBS, 1993).

In this paper, we present a comprehensive and general model for the continuous tour scheduling problem. This model implicitly supports most forms of flexibility described above. More precisely, it incorporates flexibility in terms of shift starting time and length, break placement, days-on patterns, start-time float, shift-length float and break-placement float. It can be extended to situations in which multiple breaks are required. The proposed compact model is based on the use of *forward* and *backward* constraints to match break, shift and tour variables. This results in a compact model that is easily adaptable to a large number of contexts.

The concept of forward and backward constraints was first introduced by BECHTOLD and JACOBS (1990) for the shift scheduling problem and is here adapted to the tour scheduling problem. BECHTOLD and JACOBS (1996) proved the equivalence of their compact shift scheduling model (involving forward and backward constraints) to a generalized set covering formulation under the assumption of no extraordinary overlap (the concept of extraordinary overlap will be defined in the next section). In this paper, we present a proof of the correctness of forward and backward constraints to match breaks to shifts and to match shifts to tours. The proof is based on Benders decomposition applied to a formulation of the tour scheduling problem that uses transportation subproblems. We show that forward and backward constraints are a particular class of Benders cuts that are sufficient for optimality when there is no extraordinary overlap. When extraordinary overlap is present, it can be shown that additional Benders cuts must be generated to identify an optimal solution.

The remainder of the paper is organized as follows. In the next section, we present an overview of the literature in personnel scheduling, focusing on continuous environments and implicit formulations. In Section 3.3, we present a simple model for the shift scheduling problem, followed by a compact model that uses the forward and backward constraints. In Section 3.4 we then demonstrate the equivalence

between the two models when there is no extraordinary overlap. Relying on this proof, we also present a similar formulation for the problem of matching shifts to eligible days-on patterns. An implicit formulation of the tour scheduling problem is then presented in Section 3.5, followed by extensions of this model in Section 3.6. We analyze the size of the compact model in Section 3.7 and offer conclusions in Section 3.8.

3.2 Literature Review

Personnel scheduling problems have been the object of a large number of studies in the last fifty years. On the whole, two important approaches have been adopted to model such problems. The first one, initially introduced by DANTZIG (1954), is based on a generalized set covering formulation that explicitly represents each feasible work schedule by an integer decision variable (see, e.g., MORRIS and SHOWALTER, 1983; EASTON and ROSSIN, 1991; BRUSCO and JACOBS, 1993; BECHTOLD and BRUSCO, 1994). When several forms of flexibility are considered, the number of feasible schedules is large and the resulting set covering problem may be very difficult to solve. Some attention has thus been given to the development of both heuristic and exact solution methods to solve these large-scale set covering problems (see, e.g., MORRIS and SHOWALTER, 1983; BECHTOLD and SHOWALTER, 1987; EASTON and ROSSIN, 1991; BRUSCO et al., 1995). In a similar vein, considerable progress has been made in solving large-scale airline crew scheduling problems (see, e.g., HOFFMAN and PADBERG, 1993; VANCE et al., 1997; GAMACHE et al., 1999). Crew scheduling problems deal mainly with the assignment of crews to flights to create work schedules. Although these problems involve discrete tasks, many concepts used in their solution can be applied to shift and tour scheduling given that demand is discretized according to successive time periods.

Rather than focusing on the heuristic solution of large-scale problems, other researchers have tried to model the personnel scheduling problem by using a second approach based on an implicit representation of feasible work schedules. For problems that incorporate a high degree of flexibility, as is often the case nowadays, this approach considerably reduces the size of the model when compared to the set covering approach (BECHTOLD and JACOBS, 1990; AYKIN, 1996, 2000; BRUSCO and JACOBS, 2000).

Along these lines, MOONDRA (1976) proposed a method to implicitly represent shifts without considering meal breaks. His approach provides an efficient modeling of multiple shift lengths and starting times. BECHTOLD and JACOBS (1990) later presented a compact integer model to handle meal-break placement flexibility for the shift scheduling problem. They consider shift variables as well as break variables and ensure a correct matching between them by using forward and backward constraints. They showed that their formulation could be solved faster than the generalized set covering formulation. They also proved in a subsequent paper (BECHTOLD and JACOBS, 1996) the equivalence between their implicit model and the generalized set covering model under the assumption of no extraordinary overlap (E.O.). They defined E.O. as the condition in which there exist two shifts such that the break window for one shift starts strictly earlier and ends strictly later than the break window for the other shift. THOMPSON (1995) presented a doubly-implicit integer shift scheduling model that combines the work of MOONDRA (1976) to implicitly represent shifts and the work of BECHTOLD and JACOBS (1990) to implicitly match shifts with eligible breaks.

AYKIN (1996) considered a more general shift scheduling problem with multiple breaks and disjoint break windows. He implicitly modeled the placement of these breaks by defining a break variable for each type of break, each shift (specified by a

starting time, a length and break windows) and each possible starting time in the associated break window. Given this definition of break and shift variables, E.O. is permitted and the matching between shifts and eligible breaks is done through a set of equality constraints, one for each break type and each shift. AYKIN (2000) showed that, for the family of problems considered, his formulation, although less compact, leads to optimal solutions more quickly than the extended formulation of BECHTOLD and JACOBS (1990).

BAILEY (1985) was the first to present a tour scheduling formulation that implicitly models the start-time float flexibility. JARRAH et al. (1994) proposed an implicit model for the discontinuous tour scheduling problem. Their model is inspired by the work of BURNS and CARTER (1985) on the days-off scheduling problem and the work of BECHTOLD and JACOBS (1990) on the shift scheduling problem. It incorporates the forward and backward constraints for the matching between breaks and shifts and other constraints that place lower bounds on the total number of full time employees needed.

AYKIN (1995) considered a weekly planning horizon and extended his approach for the shift scheduling problem (AYKIN, 1996) to the tour scheduling problem. In his model, a tour is specified by daily shift starting times and shift lengths for a week. For a given type of daily break (lunch or relief breaks), a break variable is defined for each tour, each day, and each possible starting time within a day. Finally, a day-off variable is defined for each tour and each day of the week. A set of equality constraints is then written for each type of daily break, each tour and each day of the week to match tours with eligible breaks. Another set of constraints is also needed to correctly match tours and day-off variables. AYKIN (1995) limited his model to the case where each employee works five days and receives two (consecutive or non

consecutive) days off during a week. This assumption made the matching between tours and day-off variables easy to model.

JACOBS and BRUSCO (1996) proposed a model that implicitly represents overlapping start-time bands. They define the start-time band as the range in which starting times can vary from one day to the next within an individual tour. A shift variable as well as a days-on variable is associated to each possible start-time band and a set of constraints matching shifts to days-on patterns is considered for each day and each start-time band.

Recently, BRUSCO and JACOBS (2000) presented a compact integer programming model for the continuous tour scheduling problem that supports flexibility in terms of meal-break placement and starting times. Their model extends the meal-break placement formulation of BECHTOLD and JACOBS (1990), integrates the model of JACOBS and BRUSCO (1996) for the overlapping start-time bands, and handles the case where a minimum time separation is imposed between shift starting times (JACOBS and BRUSCO, 1998).

Exact solution methods proposed for the implicit formulations are based on branch-and-bound techniques (MORRIS and SHOWALTER, 1983; BECHTOLD and JACOBS, 1990; THOMPSON, 1995; BRUSCO and JACOBS, 2000), branch-and-cut techniques (AYKIN, 1998), and branch-and-bound techniques with decomposition procedures (JARRAH et al., 1994). The branch-and-bound process is often initiated by adding to the model a cut obtained by solving the LP relaxation of the problem and rounding up the objective function value so as to obtain a lower bound on the cost of an optimal integer solution. This cut can also be updated after each branching decision.

3.3 Formulations of the Shift Scheduling Problem

In this section, we present implicit models for the continuous shift scheduling problem. The models presented differ from the classical generalized set-covering formulation, introduced by DANTZIG (1954), in which each feasible shift schedule is represented explicitly by a decision variable. Instead, they are based on an implicit representation of shifts and breaks. The proposed models take into account several constraints regarding labor requirements and the collective agreement. They also incorporate start-time, shift-length and break-placement flexibilities. We first describe the context in which the models can be applied. We then define their parameters and decision variables. In Section 3.3.2, we develop a basic formulation of the problem which is simple but large. A second formulation, in which the number of variables and constraints is considerably reduced, is then proposed in Section 3.3.3. As mentioned before, this compact formulation uses the forward and backward constraints introduced by BECHTOLD and JACOBS (1990).

3.3.1 Scheduling environment and assumptions

We consider a continuous planning horizon of w days. As is common in shift and tour scheduling models, the planning horizon is divided into *periods* of equal length. These periods are indexed in a chronological ascending order and the set of all periods is denoted by I . The number of employees required for each demand period $i \in I$ is denoted by d_i and is assumed to be known in advance.

Let J denote the set of all possible shifts. A *shift* is defined by a starting date and time, a length and a break window. As in most work on shift scheduling, a *break window* is specified for each shift $j \in J$ and consists of a set of consecutive periods

$P_j \subseteq I$ in which the break must take place. It is worth noting that the number of shifts depends on the length of the planning horizon since the shift starting dates have to be considered. Scheduling flexibility is thus provided by allowing shifts to be of different lengths and by allowing the break assigned to a working shift to begin in a specified interval. Here, we assume that each shift is assigned exactly one break. The case where several breaks are assigned to each shift can be easily modeled as explained in Section 3.6. Breaks are permitted to be of different lengths but the only restriction on the assignment of breaks to shifts is given by the break window. We also assume that a break must completely cover one or more periods, i.e., the length of a break must be a multiple of that of a period. This can always be accomplished by choosing a sufficiently small length for the periods. Understaffing is forbidden but could be taken into account by introducing an understaffing slack variable in each demand constraint and penalizing understaffing in the objective function (see, e.g., BAILEY, 1985).

3.3.2 Basic formulation of the shift scheduling problem

We first present a general and simple model for the shift scheduling problem. As stated before, instead of assigning a variable to each shift-break combination, we consider shift variables as well as break variables. Since only one variable is used to represent all shifts that differ only in the placement of the break, the number of shift variables is considerably reduced in comparison with set-covering formulations. For each shift $j \in J$, let S_j represent the number of employees working shift j .

To determine the set K of all possible breaks, one must consider the different break lengths as well as the break windows of all shifts. Remember that break windows are given (or can be easily converted) in terms of periods and that the length of the

break represents one or more periods. One can thus easily determine all potential breaks having a given length. For example, if a break must cover m periods (with $m \geq 1$) then, a break $k \in K$ is defined for each set of m consecutive periods entirely contained in the break window of at least one shift. Instead of characterizing a break window by the demand periods it covers, one can redefine the break window by specifying the break indices that it contains. Thus, if a break window P_j is initially defined as the set of l consecutive periods, $i, i + 1, \dots, i + l - 1$, it can be redefined as the set of all indices $k \in K$ such that all periods covered by the break k are contained in the set $\{i, i + 1, \dots, i + l - 1\}$. In the following we consider the latter, more convenient, representation. For each break k , let B_k denote the total number of breaks k assigned to the employees.

In our first formulation, the matching between shifts and eligible breaks is done through additional variables X_{jk} that denote the number of shifts j to which a break k is assigned. To describe the model, we also introduce the following parameters:

c_j = cost associated to shift $j \in J$;

δ_{ij} = 1 if shift $j \in J$ covers period $i \in I$; 0 otherwise;

ρ_{ik} = 1 if break $k \in K$ covers period $i \in I$; 0 otherwise.

The basic model, denoted by (P1), is written as follows:

$$(P1) \quad \text{minimize} \quad \sum_{j \in J} c_j S_j \quad (3.1)$$

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} B_k \geq d_i \quad \forall i \in I \quad (3.2)$$

$$\sum_{k \in P_j} X_{jk} - S_j = 0 \quad \forall j \in J \quad (3.3)$$

$$\sum_{j: k \in P_j} X_{jk} - B_k = 0 \quad \forall k \in K \quad (3.4)$$

$$X_{jk} \geq 0 \text{ and integer} \quad \forall j \in J; \forall k \in P_j \quad (3.5)$$

$$S_j \geq 0 \text{ and integer} \quad \forall j \in J \quad (3.6)$$

$$B_k \geq 0 \text{ and integer} \quad \forall k \in K. \quad (3.7)$$

The objective function (3.1) minimizes the total cost of all shifts over the planning horizon. Demand constraints (3.2) ensure that the number of employees working in a certain period, i.e., those who are present and not on break, is at least equal to the demand for that period. Constraints (3.3)-(3.4) ensure that each shift receives exactly one break within its break window.

The X_{jk} variables can be seen as network flow variables in a bipartite network $G = (J \cup K, A)$ where an arc (j, k) is defined if break k belongs to the break window associated with shift j , i.e., $k \in P_j$. The assignment of a compatible break to a shift is then done by dispatching all supplies (i.e., shifts) towards all demands (i.e., breaks) using the arcs of A (i.e., satisfying the break window of each shift). In fact, formulation (P1) is similar to Aykin's formulation (AYKIN, 1996) in the context of a single break. In comparison with (P1), Aykin's formulation uses variables X_{jk} as

break variables and reduces the number of variables and constraints by using the equations (3.4) to replace B_k with a set of X_{jk} variables.

Given the structure of problem (P1) and the assumption of no E.O., we show in the next section that it is possible to reduce the number of variables and constraints by removing variables X_{jk} and replacing constraints (3.3)-(3.5) with a relatively small set of constraints that correspond to the forward and backward constraints plus an additional equality constraint.

3.3.3 Compact formulation of the shift scheduling problem

In order to describe forward and backward constraints, we first define total order relations \prec on the set J of shifts and the set K of breaks. In this order, a shift j_1 is smaller than a shift j_2 (denoted $j_1 \prec j_2$) if shift j_1 starts before shift j_2 or if they both start at the same time and j_1 finishes before j_2 ; if both shifts have the same starting time and duration, then $j_1 \prec j_2$ if the break window of shift j_1 starts before that of j_2 or if they both start at the same time but that of j_1 finishes before that of j_2 . Similarly, a break k_1 is smaller than a break k_2 (denoted $k_1 \prec k_2$) if break k_1 starts before break k_2 or if they both start at the same time and k_1 finishes before k_2 . These orders define unique minimal and maximal elements in the sets J and K .

To write forward and backward constraints, let K^s (resp. K^e) be the set of first (resp. last) breaks associated with the break windows of all shifts: $K^s = \cup_{j \in J} \{\min(P_j)\}$ and $K^e = \cup_{j \in J} \{\max(P_j)\}$. Also, for any break $k \in K^s$, consider the set $K^B(k)$ of all breaks k' greater than or equal to k , i.e., $K^B(k) = \{k' \in K | k \preceq k'\}$, and the set $J^B(k)$ of all shifts $j \in J$ whose break window P_j starts with or after break k , i.e., $J^B(k) = \{j \in J | P_j \subseteq K^B(k)\}$. Similar definitions adapted for last

breaks $k \in K^e$ yield the set $K^F(k) = \{k' \in K | k' \preceq k\}$ of all breaks smaller than or equal to k and the set $J^F(k) = \{j \in J | P_j \subseteq K^F(k)\}$ of all shifts whose break windows P_j finishes with or before break k . Finally, let also $k_s = \min(K)$ and $k_e = \max(K)$ denote the first and last breaks, respectively, in set K .

Forward and backward constraints are then given, respectively, by:

$$\sum_{k' \in K^F(k)} B_{k'} \geq \sum_{j \in J^F(k)} S_j \quad \forall k \in K^e \setminus \{k_e\} \quad (3.8)$$

$$\sum_{k' \in K^B(k)} B_{k'} \geq \sum_{j \in J^B(k)} S_j \quad \forall k \in K^s \setminus \{k_s\}. \quad (3.9)$$

Forward constraints (3.8) ensure that in the time interval that goes from the beginning of break k_s to the end of any break $k \in K^e \setminus \{k_e\}$, the number of breaks assigned is larger than or equal to the number of shifts with a break window fully contained in this interval. Similarly, backward constraints (3.9) ensure that in the time interval that goes from the beginning of any break $k \in K^s \setminus \{k_s\}$ to the end of break k_e , the number of breaks assigned is larger than or equal to the number of shifts with a break window fully contained in this interval.

As stated before, the compact formulation of the shift scheduling problem, denoted by (P2), is obtained from (P1) by removing the variables X_{jk} and replacing the set of constraints (3.3)-(3.5) by the forward and backward constraints (3.8)-(3.9) and the following equality constraint:

$$\sum_{j \in J} S_j - \sum_{k \in K} B_k = 0 \quad (3.10)$$

The equality constraint (3.10) ensures the assignment of a unique break to each shift. This constraint was implicitly satisfied in model (P1).

3.4 Equivalence between Formulations (P1) and (P2)

Consider again formulation (P1). For any feasible solution (\bar{S}, \bar{B}) to constraints (3.2), (3.6) and (3.7) that involve only the S_j and B_k variables, the problem reduces to constraints (3.3)-(3.5) which define a transportation subproblem $SP1(\bar{S}, \bar{B})$, in the X_{jk} variables, for which a feasible flow must be determined. Given this and the assumptions made in Section 3.3.1, we show the derivation of forward and backward constraints from the algebraic structure of the problem as well as the equivalence between formulations (P1) and (P2) when there is no E.O.. Our approach is based on Benders decomposition. For reasons of clarity, we will prove the equivalence for the shift scheduling problem. However, the result also applies to any problem having the same structure, such as the tour scheduling problem we formulate later.

3.4.1 Benders reformulation

Let R be the set of vectors (S, B) satisfying constraints (3.2), (3.6) and (3.7) of (P1). For any given vector $(\bar{S}, \bar{B}) \in R$, the resulting *primal subproblem*, $SP1(\bar{S}, \bar{B})$, is defined as:

$$SP1(\bar{S}, \bar{B}) \quad \text{minimize} \quad \sum_{j \in J} c_j \bar{S}_j$$

$$\sum_{k \in P_j} X_{jk} = \bar{S}_j \quad \forall j \in J \quad (3.11)$$

$$\sum_{j: k \in P_j} X_{jk} = \bar{B}_k \quad \forall k \in K \quad (3.12)$$

$$X_{jk} \geq 0 \text{ and integer} \quad \forall j \in J, \forall k \in P_j.$$

Since the objective function of $(P1)$ does not involve the X_{jk} variables, the optimal value of subproblem $SP1(\bar{S}, \bar{B})$ has a constant value. $SP1(\bar{S}, \bar{B})$ is feasible only if the following equality is satisfied:

$$\sum_{j \in J} \bar{S}_j = \sum_{k \in K} \bar{B}_k. \quad (3.13)$$

Let $\lambda = (\lambda_j : j \in J)$ and $\mu = (\mu_k : k \in K)$ denote the dual variables associated with constraints (3.11) and (3.12), respectively. The dual of the linear relaxation of $SP1(\bar{S}, \bar{B})$, the *dual subproblem*, is defined as follows:

$$\begin{aligned} SD1(\bar{S}, \bar{B}) \quad & \text{maximize} \quad \sum_{j \in J} \lambda_j \bar{S}_j + \sum_{k \in K} \mu_k \bar{B}_k \\ & \lambda_j + \mu_k \leq 0 \quad \forall j \in J, \forall k \in P_j. \end{aligned} \quad (3.14)$$

Recall that our objective is restricted to ensuring the feasibility of the primal subproblem. By duality theory, we know that the primal is feasible only if its dual is bounded. Moreover, a theorem in polyhedral theory asserts that a feasible linear program $(P) = \{\max cx \text{ s.t. } x \in X\}$ is bounded if there is no extreme ray $r^* \in X$ with $cr^* > 0$ (see, e.g., NEMHAUSER and WOLSEY, 1988). This result, however, relies on the assumption that the constraint matrix is of full rank. This assumption is not necessarily satisfied for the dual subproblem $SD1(\bar{S}, \bar{B})$ because the number of variables may be larger than the number of constraints. To get around this difficulty, we restrict the feasible region of $SD1(\bar{S}, \bar{B})$. Let Q denote the feasible region of $SD1(\bar{S}, \bar{B})$. Observe that Q does not depend on \bar{S} and \bar{B} . The restricted feasible region, denoted by Q^+ , imposes sign constraints on variables and is given by :

$$Q^+ = \{(\lambda, \mu) | \lambda_j + \mu_k \leq 0, \forall j \in J, \forall k \in P_j; \lambda \geq 0; \mu \leq 0\}.$$

For any feasible solution $(\lambda, \mu) \in Q$, let $\alpha_{\max} = \max\{|\lambda_j|, |\mu_k| : j \in J, k \in K\}$ and (λ^+, μ^+) be the vector defined as: $\lambda_j^+ = \lambda_j + \alpha_{\max}, \forall j \in J$ and $\mu_k^+ = \mu_k - \alpha_{\max}, \forall k \in K$. It is clear that $\lambda_j^+ \geq 0, \forall j \in J$ and $\mu_k^+ \leq 0, \forall k \in K$. Also, $\lambda_j^+ + \mu_k^+ = \lambda_j + \mu_k \leq 0, \forall j \in J, \forall k \in P_j$. In addition,

$$\sum_{j \in J} \lambda_j^+ \bar{S}_j + \sum_{k \in K} \mu_k^+ \bar{B}_k = \sum_{j \in J} \lambda_j \bar{S}_j + \sum_{k \in K} \mu_k \bar{B}_k + \alpha_{\max} \left(\sum_{j \in J} \bar{S}_j - \sum_{k \in K} \bar{B}_k \right)$$

and, by equality (3.13),

$$\sum_{j \in J} \lambda_j^+ \bar{S}_j + \sum_{k \in K} \mu_k^+ \bar{B}_k = \sum_{j \in J} \lambda_j \bar{S}_j + \sum_{k \in K} \mu_k \bar{B}_k.$$

This shows that to each feasible solution $(\lambda, \mu) \in Q$ is associated a feasible solution $(\lambda^+, \mu^+) \in Q^+$ with equal cost. We can thus assert that the dual subproblem is bounded if and only if it is bounded on the restricted region Q^+ . Because the constraint matrix defining Q^+ is of full rank, the theorem stated above asserts that the dual subproblem $SD1(\bar{S}, \bar{B})$ is bounded if and only if, for every extreme ray (λ^r, μ^r) in Q^+ , one has:

$$\sum_{j \in J} \lambda_j^r \bar{S}_j + \sum_{k \in K} \mu_k^r \bar{B}_k \leq 0. \quad (3.15)$$

Inequalities (3.15) correspond to the feasibility cuts of the *Benders reformulation*. This reformulation corresponds to the objective (3.1) and the constraints (3.2), (3.6) and (3.7) of the basic formulation (P1) together with the inequalities (3.15).

3.4.2 Feasibility cuts and extreme rays

To determine a minimal set of rays containing all extreme rays of Q^+ , we proceed by elimination, i.e., we consider any ray of Q^+ and establish necessary conditions for this ray to be an extreme ray. Observe that Q^+ has a unique extreme point, the null vector, and a multitude of extreme rays.

Assume that (λ^r, μ^r) is an extreme ray of Q^+ , and define the sets $\bar{J} = \{j \in J | \lambda_j^r \neq 0\}$ and $\bar{K} = \{k \in K | \mu_k^r \neq 0\}$. $\bar{K} \neq \emptyset$ since $\bar{K} = \emptyset \Rightarrow \bar{J} = \emptyset$ which corresponds to the null vector. Let $R(\bar{J}, \bar{K})$ be a ray defined on $\mathbb{R}^{|\bar{J}|+|\bar{K}|}$ by:

$$R_l(\bar{J}, \bar{K}) = \begin{cases} 1 & \text{if } l \in \bar{J} \\ -1 & \text{if } l \in \bar{K} \\ 0 & \text{otherwise} \end{cases}$$

We now show that extreme rays in Q^+ are of the form $R(\bar{J}, \bar{K})$. Consider the value $\epsilon = \frac{1}{2} \min\{|\lambda_j^r|, |\mu_k^r| : j \in \bar{J}, k \in \bar{K}\}$. The two rays $(\lambda^1, \mu^1) = (\lambda^r, \mu^r) - \epsilon \times R(\bar{J}, \bar{K})$ and $(\lambda^2, \mu^2) = (\lambda^r, \mu^r) + \epsilon \times R(\bar{J}, \bar{K})$ are both in Q^+ and satisfy the relation $(\lambda^r, \mu^r) = \frac{1}{2} (\lambda^1, \mu^1) + \frac{1}{2} (\lambda^2, \mu^2)$. However, by definition, an extreme ray cannot be written as a positive linear combination of two other distinct rays. Thus, (λ^r, μ^r) is an extreme ray only if (λ^1, μ^1) and (λ^2, μ^2) represent the same ray, i.e., $\exists \eta > 0$ such that $(\lambda^1, \mu^1) = \eta \times (\lambda^2, \mu^2)$, which is equivalent to saying that $\exists \eta' > 0$ such that $(\lambda^r, \mu^r) = \eta' \times R(\bar{J}, \bar{K})$. Extreme rays are thus of the form $R(\bar{J}, \bar{K})$.

Now, consider the bipartite network $G = (J \cup K, A)$ where an arc $(j, k) \in A$ is defined if $k \in P_j$. Consider the subgraph $G(\bar{J}, \bar{K}) = (\bar{J} \cup \bar{K}, \bar{A})$ of G , where an arc $(j, k) \in \bar{A}$ is defined if the corresponding constraint (3.14) is satisfied as an equality, i.e., $\lambda_j^r + \mu_k^r = 0$. Let \bar{J}^i and \bar{K}^i be subsets of \bar{J} and \bar{K} respectively, such that

every set of nodes $(\bar{J}^i \cup \bar{K}^i)$ represents a connected component of $G(\bar{J}, \bar{K})$. Then $(\lambda^r, \mu^r) = \sum_{i=1}^t R(\bar{J}^i, \bar{K}^i)$ where t denotes the number of connected components. Observe also that $R(\bar{J}^{i_1}, \bar{K}^{i_1})$ and $R(\bar{J}^{i_2}, \bar{K}^{i_2})$ are different if and only if $\bar{J}^{i_1} \neq \bar{J}^{i_2}$ and $\bar{K}^{i_1} \neq \bar{K}^{i_2}$. These observations prove the following proposition.

Proposition 1. (λ^r, μ^r) is an extreme ray of Q^+ only if $(\lambda^r, \mu^r) = R(\bar{J}, \bar{K})$ where $(\bar{J} \cup \bar{K})$ is a connected component of $G(\bar{J}, \bar{K})$.

3.4.3 Redundant constraints

In Section 3.4.1, we established, through Benders decomposition, that the feasibility of the primal subproblem $SP1(\bar{S}, \bar{B})$ can be ensured by considering feasibility cuts of the form

$$\sum_{j \in J} \lambda_j^r \bar{S}_j + \sum_{k \in K} \mu_k^r \bar{B}_k \leq 0$$

for every extreme ray (λ^r, μ^r) of Q^+ .

Now, observe that the connected components of $G(\bar{J}, \bar{K})$ may take one of two forms. First, there are connected components including a single node $k \in \bar{K}$. By Proposition 1, the rays associated to these particular connected components are $R(\emptyset, \{k\})$, $k \in \bar{K}$. They generate feasibility cuts of the form $\bar{B}_k \geq 0$ which are already present in the original formulation. Hence extreme rays of this type are called trivial rays.

Second, there are connected components of the form $(\bar{J} \cup P(\bar{J}))$ where $P(\bar{J}) = \cup_{j \in \bar{J}} P_j$. By Proposition 1, one can conclude that the set of extreme rays, other than the trivial ones, is contained in Q^{min} where $Q^{min} = \{R(\bar{J}, P(\bar{J})) \mid \bar{J} \subseteq J \text{ and } (\bar{J} \cup P(\bar{J})) \text{ is a connected component of } G(\bar{J}, P(\bar{J}))\}$.

We prove next that, when extraordinary overlap is absent, most of the rays contained in Q^{min} generate feasibility cuts that are in fact dominated by the equality constraint (3.13) and other cuts generated from a small subset of Q^{min} . These dominating cuts correspond to rays $R(J^F(k), K^F(k))$, $k \in K^e \setminus \{k_e\}$ and $R(J^B(k), K^B(k))$, $k \in K^s \setminus \{k_s\}$, where K^e , K^s , k_e , k_s , $J^F(k)$, $K^F(k)$, $J^B(k)$ and $K^B(k)$ are defined in Section 3.3.3. These cuts are forward and backward constraints described in that same section. This results in the following proposition.

Proposition 2. Under the assumption of no extraordinary overlap, feasibility cuts generated by rays $R(\bar{J}, P(\bar{J}))$ of Q^{min} are dominated by forward and backward constraints and the equality constraint.

Proof: Let $R(\bar{J}, P(\bar{J}))$ be a ray of Q^{min} . By (3.15), this ray yields the following cut:

$$\sum_{j \in \bar{J}} \bar{S}_j - \sum_{k \in P(\bar{J})} \bar{B}_k \leq 0. \quad (3.16)$$

Now, let $k^1 = \max\{k \in P(\bar{J})\}$ and $k^2 = \min\{k \in P(\bar{J})\}$. Then $k^1 \in K^e$ and $k^2 \in K^s$. The cuts generated by $R(J^F(k^1), K^F(k^1))$ and $R(J^B(k^2), K^B(k^2))$ are, respectively,

$$\sum_{j \in J^F(k^1)} \bar{S}_j - \sum_{k \in K^F(k^1)} \bar{B}_k \leq 0 \quad (3.17)$$

$$\sum_{j \in J^B(k^2)} \bar{S}_j - \sum_{k \in K^B(k^2)} \bar{B}_k \leq 0. \quad (3.18)$$

From (3.17) and (3.18), it follows that

$$\sum_{j \in J^F(k^1)} \bar{S}_j + \sum_{j \in J^B(k^2)} \bar{S}_j - \sum_{k \in K^F(k^1)} \bar{B}_k - \sum_{k \in K^B(k^2)} \bar{B}_k \leq 0. \quad (3.19)$$

From the assumptions of Section 3.3.1, one can also derive the following results:

$$\begin{aligned}
K^F(k^1) \cup K^B(k^2) &= K \\
K^F(k^1) \cap K^B(k^2) &= \{k \in K \mid k^2 \preceq k \preceq k^1\} \\
&= P(\bar{J}) \\
J^F(k^1) \cap J^B(k^2) &= \{j \in J \mid P_j \subseteq K^F(k^1) \cap K^B(k^2)\} \\
&= \{j \in J \mid P_j \subseteq P(\bar{J})\} \\
&= \bar{J} \cup J' \text{ where } J' = \{j \in J \setminus \bar{J} \mid P_j \subseteq P(\bar{J})\}.
\end{aligned}$$

Of course, $J^F(k^1) \cup J^B(k^2) \subseteq J$. We can prove by contradiction that $J \subseteq J^F(k^1) \cup J^B(k^2)$ if extraordinary overlap is absent. Assume that there exists $j \in J$ such that $j \notin (J^F(k^1) \cup J^B(k^2))$. Thus, $j \notin J^F(k^1) \Leftrightarrow \exists k' \in P_j$ such that $k' \succ k^1$ and $j \notin J^B(k^2) \Leftrightarrow \exists k'' \in P_j$ such that $k'' \prec k^2$. This is equivalent to saying that the break window P_j of shift j contains in the strict sense all break windows $P_{j'}$ of shifts $j' \in \bar{J}$, which contradicts the assumption of the absence of extraordinary overlap. Thus, under the absence of extraordinary overlap, $J^F(k^1) \cup J^B(k^2) = J$.

As a consequence, inequality (3.19) implies that

$$\sum_{j \in J} \bar{S}_j + \sum_{j \in \bar{J}} \bar{S}_j + \sum_{j \in J'} \bar{S}_j - \sum_{k \in K} \bar{B}_k - \sum_{k \in P(\bar{J})} \bar{B}_k \leq 0. \quad (3.20)$$

But because of equality (3.13), inequality (3.20) reduces to

$$\sum_{j \in \bar{J}} \bar{S}_j + \sum_{j \in J'} \bar{S}_j - \sum_{k \in P(\bar{J})} \bar{B}_k \leq 0. \quad (3.21)$$

Since the S_j variables are non negative, inequality (3.21) yields to

$$\sum_{j \in \bar{J}} \bar{S}_j - \sum_{k \in P(\bar{J})} \bar{B}_k \leq 0.$$

which corresponds exactly to cut (3.16) generated by $R(\bar{J}, P(\bar{J}))$. \square

It is worth mentioning that when extraordinary overlap is present, forward and backward constraints no longer dominate all other feasibility cuts. This comes from the fact that there then exists a ray for which the associated indices k^1 and k^2 and the associated sets $J^F(k^1)$ and $J^B(k^2)$ are such that $J^F(k^1) \cup J^B(k^2) \neq J$.

In conclusion, forward and backward constraints represent particular Benders feasibility cuts that are generated from a subset of Q^{min} . By considering only forward and backward cuts, and under equality (3.13), one can generate all cuts defining the feasible region of the primal Benders subproblem, given that extraordinary overlap is absent. Because the number of such cuts is small, they can be explicitly enumerated and added to the Benders master problem to obtain formulation (P2). Moreover, the proof of the equivalence between (P1) and (P2) highlights the relation between the formulation of AYKIN (1996) and that of BECHTOLD and JACOBS (1990). This results in the following proposition.

Proposition 3. Under the assumption of no extraordinary overlap, the linear relaxations of the formulation of AYKIN (1996) and that of BECHTOLD and JACOBS (1990) for the shift scheduling problem with a single break are equivalent and both formulations yield the same integrality gap.

Despite this result, either Aykin's formulation or that of Bechtold and Jacobs may be easier to solve depending on the context. If the number of alternative shifts (i.e.,

combinations of shift starting time, shift length, break types and break windows) is large, the number of break variables and matching constraints between shifts and eligible breaks will grow considerably in Aykin’s model compared with a slight increase in the model of Bechtold and Jacobs extended to the context of multiple breaks (as presented in Section 3.6). On the other hand, if the number of shift instances is small, Aykin’s formulation becomes more efficient (see AYKIN, 2000).

3.5 The Tour Scheduling Problem

In this section, we consider the general problem of tour scheduling defined in Section 3.1. Other assumptions, presented in Section 3.5.1, become necessary to properly define this problem. As in the case of the shift scheduling problem, we first present a comprehensive but large formulation for the tour scheduling model followed by a more compact one.

3.5.1 Additional assumptions for tour scheduling

In addition to the assumptions made in Section 3, we now consider an environment that incorporates different days-on patterns. A tour is defined as a days-on pattern with a specified start-time band. The start-time band defines the range in which work starting times can vary within an individual tour. In our formulation, start-time bands consist of a set of consecutive periods of any width (the width is defined as the number of starting times included in the band). The set of all feasible tours is denoted by L . For each tour $l \in L$, the associated pattern and start-time band are denoted, respectively, by M_l and H_l . As for the break windows, a start-time band can be redefined by specifying the shift indices it contains. Hence, a start-time

band H_l associated with tour $l \in L$ contains shift j if shift j starts within the time interval corresponding to H_l on a work day in M_l .

3.5.2 Basic formulation of the tour scheduling problem

In addition to the S_j , B_k and X_{jk} variables defined in Section 3.3.2, we define a tour variable T_l , for each $l \in L$, that represents the number of employees assigned to this tour. As for the shift scheduling problem, the first formulation of the tour scheduling problem uses linking variables, Y_{lj}^q , that ensure the matching between tours and eligible shifts. Variable Y_{lj}^q denotes the number of tours l to which a shift j is assigned on day q . Observe, that, contrarily to the shift scheduling problem, we introduce here another index for the day of the planning horizon. This is necessary since one shift must be assigned to each work day of an individual tour. For each day q in the planning horizon, let $H_l(q)$ be the set of eligible shifts associated with tour l that belong to day q , i.e., shifts that start in a period of day q and within the start-time band associated with tour l (a day is itself represented as a set of consecutive periods).

To describe the model, we introduce for each day q of the planning horizon the set $L(q)$ of all tours $l \in L$ for which day q is a work day as well as a set $J(q)$ of all shifts $j \in J$ that can be matched to tours $l \in L(q)$ on day q (i.e., $J(q) = \cup_{l \in L(q)} H_l(q)$). In the context considered here, $J(q)$ is composed of all shifts starting on day q . The basic model for the tour scheduling problem, denoted by $(P3)$, is obtained from the basic shift scheduling model $(P1)$ by adding the tour variables T_l , the linking

variables Y_{lj}^q and the following set of constraints to match tours with eligible shifts:

$$\sum_{j \in H_l(q)} Y_{lj}^q - T_l = 0 \quad \forall q = 1, \dots, w; \forall l \in L(q) \quad (3.22)$$

$$\sum_{l: j \in H_l(q)} Y_{lj}^q - S_j = 0 \quad \forall q = 1, \dots, w; \forall j \in J(q) \quad (3.23)$$

$$Y_{lj}^q \geq 0 \text{ and integer} \quad \forall q = 1, \dots, w; \forall l \in L; \forall j \in J \quad (3.24)$$

$$T_l \geq 0 \text{ and integer} \quad \forall l \in L. \quad (3.25)$$

It is noteworthy that, when variables S and T are fixed to values (\bar{S}, \bar{T}) and there is no start-time band which overlaps from one day to the next, the resulting subproblem formed by constraints (3.22)-(3.24) can be separated into w subproblems, $SP3(q, \bar{S}, \bar{T})$, one for each day q of the planning horizon. However, the separability of these subproblems can always be ensured even when such *day-overlapping* start-time bands exist. This can be done by adding a small set of shift variables called *wraparound shift variables*. A wraparound shift variable is a copy of an existing shift variable that starts within the intersection of a day-overlapping start-time band and the following day. Thus, for a start-time band, defined as the set of $(l + 1)$ consecutive periods $i, i + 1, \dots, i + l$, and which overlaps from day q to day $(q + 1)$ on the last $(m + 1)$ periods $i + l - m, \dots, i + l$, one must duplicate all existing shift variables starting at periods $i + l - m, \dots, i + l$. These additional wraparound shift variables are then added to the set $J(q)$ to ensure the separability of subproblems $SP3(q, \bar{S}, \bar{T})$ and $SP3(q + 1, \bar{S}, \bar{T})$. At first sight, one must define wraparound shift variables only for the $(w - 1)$ first days of the planning horizon. However, these additional variables also become necessary for the last day of the planning horizon in order to provide all the possible starting times for the tours in $L(w)$ having a day-overlapping start-time band.

Subproblems, $SP3(q, \bar{S}, \bar{T})$, are defined as follows:

$$SP3(q, \bar{S}, \bar{T}) \quad \text{minimize} \quad \sum_{j \in J(q)} c_j \bar{S}_j$$

$$\sum_{j \in H_l(q)} Y_{lj}^q = \bar{T}_l \quad \forall l \in L(q) \quad (3.26)$$

$$\sum_{l: j \in H_l(q)} Y_{lj}^q = \bar{S}_j \quad \forall j \in J(q) \quad (3.27)$$

$$Y_{lj}^q \geq 0 \text{ and integer} \quad \forall l \in L(q); \forall j \in J(q)$$

Observe that each subproblem $SP3(q, \bar{S}, \bar{T})$ is a transportation problem in a bipartite network $G^q = (L(q) \cup J(q), A(q))$ where an arc $(l, j) \in A(q)$ is defined if shift j is eligible for tour l . This problem has exactly the same structure as subproblem ($SP1$). As shown in Section 3.4, it is thus possible to replace this set of constraints by appropriate forward and backward constraints as well as an equality constraint by following the same approach as in Section 3.3, given that extraordinary overlap between start-time bands is absent. The same definition of extraordinary overlap for break windows is adopted here for start-time bands. Benders feasibility cuts associated with day q are given by:

$$\sum_{l \in L(q)} \eta_l^{qr} \bar{T}_l + \sum_{j \in J(q)} \gamma_j^{qr} \bar{S}_j \leq 0.$$

where (η^{qr}, γ^{qr}) is an extreme ray of the restricted feasibility region of the dual of the linear relaxation of $SP3(q, \bar{S}, \bar{T})$.

3.5.3 Compact model of the tour scheduling problem

3.5.3.1 Forward and backward constraints for shifts

For each day q of the planning horizon, let $J^s(q)$ (resp. $J^e(q)$) be the set of first (resp. last) shifts starting on day q and associated with the start-time bands of all tours for which day q is a work day. Thus $J^s(q) = \cup_{l \in L(q)} \{\min(H_l(q))\}$ and $J^e(q) = \cup_{l \in L(q)} \{\max(H_l(q))\}$. Also, for each day q of the planning horizon and for each $j \in J^e(q)$, define $J^F(j, q) = \{j' \in J(q) | j' \preceq j\}$ and $L^F(j, q) = \{l \in L(q) | H_l(q) \subseteq J^F(j, q)\}$. Similar definitions for day q and shift $j \in J^s(q)$ yield the sets $J^B(j, q) = \{j' \in J(q) | j \preceq j'\}$ and $L^B(j, q) = \{l \in L(q) | H_l(q) \subseteq J^B(j, q)\}$. Finally, for each day q , let also $j_s(q) = \min\{j | j \in J^s(q)\}$ and $j_e(q) = \max\{j | j \in J^e(q)\}$ denote the first and last shifts, respectively, in set $J(q)$.

For each day q of the planning horizon, forward and backward constraints are then given, respectively, by

$$\begin{aligned} \sum_{j' \in J^F(j, q)} S_{j'} &\geq \sum_{l \in L^F(j, q)} T_l & \forall j \in J^e(q) \setminus \{j_e(q)\} \\ \sum_{j' \in J^B(j, q)} S_{j'} &\geq \sum_{l \in L^B(j, q)} T_l & \forall j \in J^s(q) \setminus \{j_s(q)\}. \end{aligned}$$

One can verify that forward and backward constraints for matching tours with shifts play the same role as those for matching shifts with breaks.

3.5.3.2 Compact formulation of the tour scheduling problem

For each day q of the planning horizon, subproblem $SP3(q, \bar{S}, \bar{T})$ has exactly the same structure as subproblem $SP1(\bar{S}, \bar{B})$ described in Section 3.4.1. Therefore, in order to have an equivalent compact formulation of (P3), one must add equality constraints between total supplies (i.e., tours) and total demands (i.e., shifts) for each transportation problem (i.e., each day). This leads to the following set of constraints:

$$\sum_{l \in L(q)} T_l = \sum_{j \in J(q)} S_j \quad \forall q = 1, \dots, w.$$

The compact formulation, (P4), is finally given by:

$$(P4) \quad \text{minimize } \sum_{j \in J} c_j S_j$$

subject to

$$\begin{aligned} \sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} B_k &\geq d_i && \forall i \in I \\ \sum_{k' \in K^F(k)} B_{k'} - \sum_{j \in J^F(k)} S_j &\geq 0 && \forall k \in K^e \setminus \{k_e\} \\ \sum_{k' \in K^B(k)} B_{k'} - \sum_{j \in J^B(k)} S_j &\geq 0 && \forall k \in K^s \setminus \{k_s\} \\ \sum_{j \in J} S_j - \sum_{k \in K} B_k &= 0 \\ \sum_{j' \in J^F(j,q)} S_{j'} - \sum_{l \in L^F(j,q)} T_l &\geq 0 && \forall q = 1..w; \forall j \in J^e(q) \setminus \{j_e(q)\} \\ \sum_{j' \in J^B(j,q)} S_{j'} - \sum_{l \in L^B(j,q)} T_l &\geq 0 && \forall q = 1..w; \forall j \in J^s(q) \setminus \{j_s(q)\} \\ \sum_{l \in L(q)} T_l - \sum_{j \in J(q)} S_j &= 0 && \forall q = 1, \dots, w \\ S_j &\geq 0 \text{ and integer} && \forall j \in J \\ B_k &\geq 0 \text{ and integer} && \forall k \in K \\ T_l &\geq 0 \text{ and integer} && \forall l \in L. \end{aligned}$$

3.6 Extensions

In order to simplify the presentation of the models, we assumed in Section 3.3.1 that each shift receives a single break. The case where several breaks (e.g., morning rest, lunch, afternoon rest) must be assigned to each shift can be modeled by following, for

each type of break, the same approach used for the single break problem. Assume that V breaks must be assigned to each shift. As in the case of a single break, we can easily enumerate all possible breaks of each type and associate a break variable to each. Hence, for each break k of type $v = 1, \dots, V$, we define a break variable B_k^v that denotes the number of such breaks assigned to the set of all shifts. Barring extraordinary overlap for each type of break (extraordinary overlap is allowed between different break types), a compact formulation can be used in the same way as in the single break problem. Thus, a set of forward and backward constraints as well as an equality constraint should be considered for each break type v . In other words, variables and constraints considered in the single break problem are rewritten for each break type with respect to the associated break windows and lengths.

The assumption of the absence of extraordinary overlap yields the compact formulation ($P4$) of the tour scheduling problem. When such overlap is present, one can use formulation ($P3$). This formulation is of large size and may be difficult to solve but, as shown in Section 3.4.1, model ($P3$) is amenable to solution by Benders decomposition. Feasibility cuts can be generated iteratively by solving the dual subproblems or, equivalently, the primal subproblems which are transportation problems. Starting from the relaxed Benders master problem obtained by dropping inequalities (3.15) from the Benders reformulation, cuts should be generated and added to this relaxed master problem until a feasible (and thus, optimal) solution is found. Notice that in the proof of the equivalence between formulations ($P1$) and ($P2$), we considered the absence of extraordinary overlap only to demonstrate that feasibility cuts associated to forward and backward constraints dominate the other feasibility cuts (see Section 3.4.3). Hence, in the case where extraordinary overlap is present, forward and backward cuts are no longer sufficient to totally

define the feasible region of the primal subproblem and other feasibility cuts must be considered.

3.7 Problem Size

This section provides estimates of the size of the compact formulation ($P4$) in hypothetical work environments that satisfy all the assumptions made in Section 3.3.1. A comparison of the sizes of model ($P4$) and an extension of the model proposed by BRUSCO and JACOBS (2000) is also presented. For convenience of presentation, we assume that all breaks have the same length and must start between the α_f^{th} and the α_l^{th} period of all shifts, where $\alpha_f < \alpha_l$. Thus, the number of periods in any shift that can contain a break is $\alpha = \alpha_l - \alpha_f + 1$. We also assume that start-time bands start every e periods such that $|I|$ is integer divisible by e , and that for a given instance all the bands have the same width b such that $b \geq e$. Finally, shifts are permitted to start at any period of the planning horizon.

Letting $|E|$ denote the number of elements in set E , the total number of shift, break and tour variables in model ($P4$) is given by:

$$V(P4) = |J| + |K| + |L|. \quad (3.28)$$

When there are no day-overlapping start-time bands (as defined in Section 3.5.2), the number of shift variables depends on the number of shift lengths, N_{sl} , and the number of possible starting times, N_{st} . As a result, $|J| = N_{sl} \times N_{st} = N_{sl} \times |I|$. In the case where day-overlapping start-time bands exist, one must consider the wraparound shift variables. One can verify that the number of all possible starting times of the wraparound shift variables on a given day is equal to $(b - e)$. Thus, the

total number of wraparound variables is $w \times N_{sl} \times (b - e)$. Hence,

$$|J| = N_{sl} \times (N_{st} + w \times (b - e)) = N_{sl} \times (|I| + w \times (b - e)).$$

Next, the number of break variables depends on the break windows of all considered shifts. One must notice that under the additional assumptions made in the beginning of this section, the number of all possible positions of break windows will depend only on the number of different shift starting times. Then, $|K| = |I| + b - e + \alpha - 1$. Finally, the number of tour variables depends on the number of days-on patterns, N_p , and the number of start-time bands, N_{sb} . Thus, $|L| = N_p \times N_{sb}$. By substitution in equality (3.28), one obtains

$$V(P4) = N_{sl} \times (|I| + w \times (b - e)) + |I| + b - e + \alpha - 1 + N_p \times N_{sb}. \quad (3.29)$$

The number of demand constraints is equal to the number of demand periods $|I|$. The number of matching constraints between shift and break variables is given by $BMC = (|K^e| - 1) + (|K^s| - 1) + 1$. Since the number of different break windows depends only on the number of different shift starting times, $|K^e|$ and $|K^s|$ are both equal to $|I| + b - e$ and $BMC = 2(|I| + b - e - 1) + 1 = 2(|I| + b - e) - 1$. Similarly, the number of matching constraints between tour and shift variables is given by $SMC = \sum_{q=1}^w (|J^e(q)| - 1) + \sum_{q=1}^w (|J^s(q)| - 1) + w = 2w \times N_{sb} - w$. Thus, the total number of constraints in (P4) is:

$$C(P4) = 3|I| + 2(b - e) + 2w \times N_{sb} - w - 1. \quad (3.30)$$

When considering an operational environment that supports flexibility in terms of break-placement, shift starting time, shift length, days-on pattern, and start-time band, the model proposed by BRUSCO and JACOBS (2000), thereafter denoted by

(P'), differs from model ($P4$) in two points. First, (P') was developed in the context of a single shift length and uses a shift variable for each possible starting time and for each start-time band involving the corresponding starting time. In order to adapt (P') to the current context, one needs to define a shift variable for each possible starting time, for each possible shift length and for each start-time band involving the corresponding starting time. The same shift (i.e., the combination of a starting time and a length) is thus replicated as many times as the number of start-time bands including the corresponding starting time. For example, if the operational environment uses start-time bands of width b that are permitted to begin at any period of the day (i.e., $e = 1$), then the number of shift variables needed in model (P') is equal to $|J(P')| = N_{st} \times |I| \times b$ which represents approximately b times the number of shift variables used in ($P4$). More generally, one can verify that, under the assumptions made in this section,

$$|J(P')| = N_{st} \times \left(\left(\left\lfloor \frac{b-1}{e} \right\rfloor + 1 \right) \times \frac{|I|}{e} + \left\lceil \frac{b-1}{e} \right\rceil \left(|I| - \frac{|I|}{e} \right) \right).$$

In fact, for a given length, the shifts starting at the lower bounds of the start-time bands (there are $|I|/e$ possible starting times in the planning horizon) are replicated $(\lfloor (b-1)/e \rfloor + 1)$ times. The remaining shift starting times $(|I| - |I|/e)$ are replicated $\lceil (b-1)/e \rceil$ times.

Second, BRUSCO and JACOBS (2000) considered a planning horizon of 7 days in which each employee must work five consecutive days. They define a tour variable for each days-on pattern and each start-time band. The matching between tours and eligible shifts is then done through a set of equality constraints, one for each start-time band and for each day in the planning horizon. The same definition of tour variables is here adopted in the extended formulation. The number of matching constraints between shifts and tours is then given by $SMC(P') = w \times N_{sb}$ which

represents approximately half the number of shift matching constraints used in $(P4)$. It is worth mentioning that except the number of shift variables and the number of matching constraints between shifts and tours, all the formulas elaborated for break variables, tour variables, demand constraints and matching constraints between breaks and shifts are the same for both models $(P4)$ and (P') . Thus, the number of variables and constraints in (P') are respectively:

$$V(P') = N_{sl} \times \left(\left(\left\lfloor \frac{b-1}{e} \right\rfloor + 1 \right) \times \frac{|I|}{e} + \left\lceil \frac{b-1}{e} \right\rceil \left(|I| - \frac{|I|}{e} \right) \right) + |I| + b - e + \alpha - 1 + N_p \times N_{sb} \quad (3.31)$$

and

$$C(P') = 3|I| + 2(b - e) + w \times N_{sb} - 1. \quad (3.32)$$

Notice here that the number of shift variables considerably affects the total number of variables in the model when shift length flexibility is considered. In an opposite manner, the number of matching constraints between shifts and tours has less effect on the total number of constraints.

The formulation of AYKIN (1995) for the tour scheduling problem is not considered in this study for two reasons. First, when applied to a context of starting-time float and shift-length float, the number of tour variables in AYKIN's model would grow dramatically. Second, AYKIN limited his model to the case where employees must work five days and rest for two consecutive or non consecutive days. We expect the number of day-off variables and tour variables to be even larger if different days-on patterns were considered.

One can verify from the formulas (3.29)-(3.32) that $(P4)$ always require fewer variables and more constraints than (P') . For problems having a unique shift length,

a small band width (e.g., five periods) and start-time bands starting every hour or half-hour (i.e., $e = 4$ or $e = 2$), the model sizes are almost equal (i.e., the difference between the number of constraints is almost equal to the difference between the number of variables). However, when the number of shift lengths and the band width become relatively large, the number of variables increases less rapidly for ($P4$) than for (P') at the cost of a small increase in the number of constraints for ($P4$). A set of hypothetical problems has been selected to illustrate the sizes of models ($P4$) and (P'). All these problems consider a 14-day planning horizon in which each employee must work at least three and no more than six consecutive days for a total of ten work days (105 patterns enumerated). Periods of fifteen minutes and a break window of 2 hours ($\alpha = 9$) are also considered. Recall that the width of a break window (or the start-time band) is given in terms of the number of periods contained in this interval. Each problem is defined as a combination of three parameters: the number of shift lengths N_{sl} (two possible values 6 and 12), the band width b (three values: 5, 9 and 13), and the separation time between start-time bands e (two values: 2 and 4). Table 3.1 reports the number of variables and constraints, in models ($P4$) and (P') for the problems considered. For example, for an operational environment which involves 6 different shift lengths, a band width of 3 hours starting every 30 minutes, model ($P4$) requires 15391 variables and 5383 constraints compared with 58819 variables and 4725 constraints for (P'). Recall that AYKIN (2000) presented a model that incorporates more variables and less constraints than the model of BECHTOLD and JACOBS (1990) and showed that, for a family of problems, his model is computationally more efficient. Although Aykin's conclusion questions the merits of reducing the number of variables, this conclusion can not be generalized to all problems. Furthermore, in comparison with (P'), model ($P4$) considerably decreases the number of variables at the cost of a small increase in the number of constraints.

Table 3.1 Comparison of model sizes

N_{sl}	b	e	Number of shift variables		Number of tour and shift matching constraints		Number of variables		Number of constraints	
			(P')	(P4)	(P')	(P4)	(P')	(P4)	(P')	(P4)
6	5	4	10080	8148	336	658	13953	12021	4369	4691
6	9	4	18144	8484	336	658	22021	12361	4377	4699
6	13	4	26208	8820	336	658	30089	12701	4385	4707
6	5	2	20160	8316	672	1330	26555	14711	4709	5367
6	9	2	36288	8652	672	1330	42687	15051	4717	5375
6	13	2	52416	8988	672	1330	58819	15391	4725	5383
12	5	4	20160	16296	336	658	24033	20169	4369	4691
12	9	4	36288	16968	336	658	40165	20845	4377	4699
12	13	4	52416	17640	336	658	56297	21521	4385	4707
12	5	2	40320	16632	672	1330	46715	23027	4709	5367
12	9	2	72576	17304	672	1330	78975	23703	4717	5375
12	13	2	104832	17976	672	1330	111235	24379	4725	5383

3.8 Conclusion

We have described a comprehensive and general model for the continuous tour scheduling problem. This model possesses a simple and flexible structure that facilitates its adaptation to different work environments encountered in practice without overly increasing its size. The use of forward and backward constraints to match shifts with tours has proven to be very efficient in reducing the size of the problem when compared to alternative formulations proposed in the literature. In the presence of break window or start-time band extraordinary overlap, forward and backward constraints are not sufficient but the problem can then be solved by Benders decomposition, using the forward and backward constraints as a set of initial feasibility cuts. The use of Benders decomposition is also interesting in cases where a total ordering of variables does not exist. This is the case, for example, when assigning schedules to employees with respect to their preferences. This paper focused on the modeling aspect of the continuous tour scheduling problem. Future

work will concentrate on the development of specialized Benders decomposition and branch-and-cut techniques to solve large-scale instances of this problem. Other extensions of this work are to consider a shift scheduling problem where employees have limited availability or a multi-activity work environment in which employees may perform several tasks within each shift. In this case, the tour scheduling problem is no longer limited to the specification of work periods and days but must also handle the assignment of activities to employees.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant CRDPJ228083-99 and by AD OPT Technologies Inc. The paper has been greatly improved thanks to the suggestions of the Guest Editor and two anonymous referees.

Bibliography

- T. AYKIN. "Reformulating the tour scheduling problem to improve solvability." GSM Working paper 95-29 , Graduate School of Management, Rutgers University, Newark, NJ.
- T. AYKIN. "Optimal shift scheduling with multiple break windows." *Management Science*, **42**:591–602 (1996).
- T. AYKIN. "A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows." *Journal of the Operational Research Society*, **49**:603–615 (1998).

- T. AYKIN. "A comparative evaluation of modeling approaches to the labor shift scheduling problem." *European Journal of Operational Research*, **125**:381–397 (2000).
- J. BAILEY. "Integrated days off and shift personnel scheduling." *Computers and Industrial Engineering*, **9**:395–404 (1985).
- K.R. BAKER. "Workforce allocation in cyclic scheduling problems: A survey." *Operational Research Quarterly*, **27(1)**:155–167 (1976).
- J.J. BARTHOLDI. "A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering." *Operations Research*, **29**:501–510 (1981).
- S.E. BECHTOLD AND M.J. BRUSCO. "Working set generation methods for labor tour scheduling." *European Journal of Operational Research*, **74**:540–551 (1994).
- S.E. BECHTOLD AND L.W. JACOBS. "Implicit modeling of flexible break assignments in optimal shift scheduling." *Management Science*, **36**:1339–1351 (1990).
- S.E. BECHTOLD AND L.W. JACOBS. "Labor utilization effects of labor scheduling flexibility alternatives in a tour scheduling environment." *Decision Sciences*, **24**:148–166 (1993).
- S.E. BECHTOLD AND L.W. JACOBS. "The equivalence of general set-covering and implicit integer programming formulations for shift scheduling." *Naval Research Logistics*, **43**:233–249 (1996).
- S.E. BECHTOLD AND M.J. SHOWALTER. "A methodology for labor scheduling in a service delivery system." *Decision Sciences*, **18**:89–107 (1987).
- J.Y. BLAIS, J. LAMONT AND J.M. ROUSSEAU. "The hastus vehicle and manpower scheduling at the Société de Transport de la Communauté Urbaine de Montréal." *Interfaces*, **20**:26–42 (1990).

- M.J. BRUSCO AND L.W. JACOBS. "A simulated annealing approach to the cyclic staff-scheduling problem." *Naval Research Logistics*, **40**:69–84 (1993).
- M.J. BRUSCO AND L.W. JACOBS. "Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations." *European Journal of Operational Research*, **86**:249–261 (1995).
- M.J. BRUSCO AND L.W. JACOBS. "Optimal models for meal-break and start-time flexibility in continuous tour scheduling." *Management Science*, **46**:1630–1641 (2000).
- M.J. BRUSCO, L.W. JACOBS, R.J. BONGIORNO, D. V. LYONS AND B. TANG. "Improving personnel scheduling at airline stations." *Operations Research*, **43**:741–751 (1995).
- R.N. BURNS AND M.W. CARTER. "Work force size and single shift schedules with variables demands." *Management Science*, **31**:599–607 (1985).
- G.B. DANTZIG. "A comment on Edie's traffic delays at toll booths." *Operations Research*, **2**:339–341 (1954).
- F.F. EASTON AND D.F. ROSSIN. "Sufficient working subsets for the tour scheduling problem." *Management Science*, **37**:1441–1451 (1991).
- M. GAMACHE, F. SOUMIS, G. MARQUIS AND J. DEROSIERS. "A column generation approach for large scale aircrew rostering problems." *Operations Research*, **47**(2):247–262 (1999).
- K.L. HOFFMAN AND M. PADBERG. "Solving airline crew scheduling problems by branch-and-cut." *Management Science*, **39**:657–682 (1993).
- J. HUETER AND W. SWART. "An integrated labor-management system for Taco-Bell." *Interfaces*, **28**:75–91 (1998).

- L.W. JACOBS AND M.J. BRUSCO. "Overlapping start-time bands in implicit tour scheduling." *Management Science*, **42**:1247–1259 (1996).
- L.W. JACOBS AND M.J. BRUSCO. "Personnel tour scheduling when starting time restrictions are present." *Management Science*, **44**:534–547 (1998).
- A.I.Z JARRAH, J.F BARD AND H. DESILVA. "Solving large-scale tour scheduling problems." *Management Science*, **40**:1124–1144 (1994).
- V.A. MABERT AND M.J. SHOWALTER. "Measuring the impact of part-time workers in service organizations." *Journal of Operational Management*, **9**:209–229 (1990).
- V.A. MABERT AND C.A. WATTS. "A simulation analysis of tour-shift scheduling construction procedures." *Management Science*, **28**:520–532 (1982).
- S.L. MOONDRA. "An L.P. model for workforce scheduling in banks." *Journal of Bank Research*, **6**:299–301 (1976).
- J.G. MORRIS AND M.J. SHOWALTER. "Simple approaches to shift, days-off and tour scheduling problems." *Management Science*, **29**:942–950 (1983).
- G. NEMHAUSER AND L. WOLSEY. *Integer and combinatorial optimization*. Wiley, New York, 1988.
- S. SCHINDLER AND T. SEMMEL. "Station staffing at Pan American World Airways." *Interfaces*, **23**:91–98 (1993).
- G.M. THOMPSON. "Improved implicit optimal modeling of the labor shift scheduling problem." *Management Science*, **41**:595–607 (1995).
- P.H. VANCE, C. BARNHART, E.L. JOHNSON AND G.L. NEMHAUSER. "Airline crew scheduling: a new formulation and decomposition algorithm." *Operations Research*, **45**:188–200 (1997).

CHAPITRE 4

IMPLICIT SHIFT SCHEDULING WITH MULTIPLE BREAKS AND WORK STRETCH DURATION RESTRICTIONS

Monia Rekik, Jean-François Cordeau, et François Soumis, soumis pour publication à *Naval Research Logistics*, novembre 2005.

Cet article s'intéresse à un problème de construction de quarts réel fourni par une agence de contrôleurs aériens. L'environnement d'opération considéré incorpore beaucoup de flexibilité. Deux formes relativement complexes sont étudiées : les pauses décomposables et les restrictions sur la durée de travail consécutif. Une pause décomposable est une pause qui peut être divisée en plusieurs sous-pauses. À notre connaissance, une telle flexibilité dans la définition des pauses n'a jamais été considérée jusqu'à date dans la littérature. Les restrictions sur la durée de travail consécutif imposent des limites sur la durée de travail avant et après chaque pause. Ceci permet d'assurer, entre autres, une bonne distribution des périodes de travail et de repos au sein du quart.

Après avoir défini le problème, nous présentons deux formulations implicites pour le modéliser. La première formulation s'inspire du travail de BECHTOLD and JACOBS (1990) et utilise des contraintes *forward* et *backward* pour placer les pauses dans les fenêtres appropriées. Les restrictions sur les durées minimale et maximale de travail consécutif sont également modélisées par des contraintes de type *forward* et *backward* adaptées. La deuxième formulation proposée est basée sur le travail d'AYKIN (1996) pour affecter les pauses aux quarts dans les intervalles pré-spécifiés. Des contraintes de type *forward* et *backward* sont incorporées pour satisfaire les

restrictions sur la durée de travail consécutif. Les résultats établis au premier article nous permettent de prouver l'équivalence entre les relaxations linéaires de ces deux formulations. Pour chacune de ces formulations, le concept de pauses décomposables est modélisé par une approche simple et efficace qui énumère les différentes séquences de durées de pauses admissibles pour les quarts ainsi que les fenêtres d'heures de début correspondantes. Ces séquences sont jumelées aux pauses standards.

En comparant son modèle à celui de BECHTOLD and JACOBS (1990), AYKIN (2000) a rapporté que, pour les instances considérées, son modèle résout les problèmes plus rapidement. Il s'est alors demandé si cette différence entre les performances n'était pas due au fait que son modèle était beaucoup moins dense. La preuve établie au chapitre 3 montre que les deux modèles induisent toujours le même saut d'intégrité, éliminant ainsi une des explications possibles de la différence entre les temps de résolution. Dans cet article, nous avons étudié l'importance du rôle que peut jouer la densité de la matrice des contraintes pour chacune de ces formulations implicites. Nous avons d'abord proposé une reformulation des contraintes *forward* et *backward* dans le but de réduire la densité du modèle de BECHTOLD and JACOBS (1990). La reformulation proposée utilise des variables d'écart additionnelles sans augmenter le nombre total des contraintes. Nous avons reformulé toutes les contraintes *forward* et *backward* présentes dans les deux modèles proposés et nous avons comparé la densité des modèles obtenus par rapport aux modèles originaux. Nous avons conclu que la reformulation proposée permet de réduire considérablement la densité des matrices de contraintes pour les deux modèles. Cette réduction est beaucoup plus importante pour le modèle s'inspirant de BECHTOLD and JACOBS (1990). Plus encore, réduire la densité a permis de diminuer considérablement les temps de calcul.

Nous avons ensuite comparé les performances des deux modèles proposés et nous avons déduit que l'extension proposée du modèle de BECHTOLD and JACOBS (1990) était meilleure pour la plupart des instances.

La notion de pause décomposable étant abordée pour la première fois dans la littérature, nous avons voulu connaître l'impact de cette nouvelle forme de flexibilité sur la taille de la main-d'œuvre. Pour cela, nous avons comparé la taille de la main-d'œuvre nécessaire dans un environnement utilisant trois pauses de durées fixes et dans un environnement avec une pause décomposable. Pour les profils de demande réels, le gain en nombre d'employés a atteint 6%. Pour des profils légèrement perturbés, ce gain a atteint 8,7%.

Nous avons ensuite étudié l'impact du concept de "restrictions sur la durée de travail consécutif" sur la taille de la main-d'œuvre. Ce concept garantit une bonne distribution des périodes de travail et de repos au sein du quart. Ceci était traditionnellement assuré en associant au quart des fenêtres de pause suffisamment espacées. Nous avons comparé notre approche à celle proposée par SEGAL (1974) et utilisée par la suite par AYKIN (1996) pour placer les fenêtres de pauses. Nous en avons conclu que, pour les instances considérées, notre approche est meilleure dans le sens où, en plus d'assurer une bonne répartition des périodes de travail et de repos, elle permet d'utiliser moins d'employés.

Enfin, nous avons étudié l'efficacité des contraintes *forward* et *backward* pour modéliser les contraintes d'espacement entre les pauses. Nous avons comparé cette approche de modélisation à l'approche énumérative utilisant toutes les possibilités de placement des fenêtres de pauses qui respectent les contraintes d'espacement. Des solutions optimales sont obtenues en des temps très courts avec l'approche implicite utilisant les contraintes *forward* et *backward*, et ce pour toutes les instances considérées. L'approche énumérative a nécessité, quant à elle, des temps de calcul beaucoup plus longs sans pour autant atteindre la solution optimale, pour certaines instances.

Cet article contribue au domaine de la construction d'horaires au niveau de la modélisation implicite en proposant une reformulation intelligente des contraintes *forward* et *backward* qui réduit considérablement la densité de la matrice des contraintes correspondantes de même que les temps de calcul. Plus encore, nous avons proposé une modélisation simple et efficace d'une nouvelle forme de flexibilité, à première vue, très complexe (les pauses décomposables). Nous avons encore une fois utilisé des contraintes *forward* et *backward* adaptées pour modéliser le concept de restrictions sur la durée de travail consécutif minimum et maximum. Dans ce contexte, nous avons montré les mérites d'une telle formulation implicite par rapport à une formulation explicite. Une deuxième contribution a été de considérer une nouvelle forme de flexibilité dans la définition des pauses : les pauses décomposables. Une telle forme, non seulement libère les compagnies du souci de répartition des durées de pause au sein des quarts, mais nous avons prouvé en plus qu'elle réduit la taille de la main-d'œuvre lorsque comparée aux pauses standards. Une troisième contribution a été de prouver, à travers des résultats expérimentaux, que les restrictions sur la durée de travail consécutif constituent une bonne approche pour placer les pauses au sein des quarts qui induit une meilleure solution que l'approche classique basée sur le temps idéal d'heures de début de pauses.

Implicit Shift Scheduling with Multiple Breaks and Work Stretch Duration Restrictions

MONIA REKIK

École Polytechnique de Montréal

JEAN-FRANÇOIS CORDEAU

HEC Montréal

FRANÇOIS SOUMIS

École Polytechnique de Montréal

November 2005

Abstract

This paper considers a shift scheduling problem which includes different forms of flexibility in terms of shift starting times, break lengths and break placement. Two particular forms are studied: fractionable breaks and work stretch duration restrictions. For some instances the use of fractionable breaks results in considerable labor savings. In addition, imposing work stretch duration restrictions is a very convenient approach to place multiple breaks within a shift. In this context, we show the merits of using so-called forward and backward constraints to model work stretch duration restrictions. We also introduce a new formulation of the forward and backward constraints and demonstrate that it considerably reduces the density of the constraint matrix of both models.

Keywords: shift scheduling, flexibility, breaks, work stretch restrictions, forward and backward constraints.

4.1 Introduction

Personnel scheduling problems are classified by BAKER (1976) into three categories: days-off, shift and tour scheduling problems. *Days-off scheduling problems* deal with the assignment of work and rest days to employees over a given planning horizon. *Shift scheduling problems*, the object of this paper, deal with the assignment of starting and finishing times to employees, and possibly the placement of relief and meal breaks within each shift. They are encountered in many service organizations that face demand varying throughout the day. *Tour scheduling problems* arise when both daily shift and days-off schedules must be determined simultaneously.

Organizations can adopt a variety of scheduling flexibility alternatives. Shift scheduling flexibility includes multiple shift-lengths (i.e., shifts can be of different durations), break placement flexibility which enables a break to start within a time window rather than at a fixed time, and shift starting time flexibility: shifts are allowed to start at any time during the day (given a certain discretization of the planning horizon). Several real-life applications have shown that the incorporation of scheduling flexibility has a very positive impact on labor utilization although it complicates the problem. BAILEY and FIELDS (1985) studied the effects of using alternative shift starting times and lengths in reducing workforce costs. BECHTOLD and JACOBS (1993) considered a scheduling environment with a single meal-break and reported that the break-placement and the shift-length flexibilities always improved labor utilization for all the demand distributions considered. AYKIN (1996) studied the effect of break-placement flexibility in problems involving multiple breaks and break windows. He showed that the number of employees needed is significantly smaller when either relief break, meal break or both types of breaks are free to start within a time interval.

Beyond the level of flexibility considered, the scheduling task may be more complex for *continuous* environments that face demand for labor 24 hours a day and seven days a week. In this context, shifts are often allowed to overlap from one day to the next. BARTHOLDI (1981) showed that some continuous problems are NP-complete. BRUSCO and JACOBS (1995) later confirmed the difficulty related to continuous problems from an experimental point of view. They considered a *discontinuous* formulation of a continuous tour scheduling problem in which daily shift schedules are prohibited to overlap from one day to the next and compared it to a *continuous* formulation. They reported that, in some cases, the use of discontinuous formulations results in a substantial amount of excess labor.

In this paper, we study a shift scheduling problem that incorporates all the complexity related to flexibility and continuous operation. In fact, the scheduling environments considered support a high level of flexibility in terms of shift starting time and length as well as the number, the duration and the placement of breaks within each shift. Moreover, we introduce a new form of break-flexibility (*fractionable breaks*) that does not assume a fixed length for each break within a shift. The concept of fractionable break (which will be defined formally in Section 4.3.1.2) is incorporated to allow the division of a break of a pre-specified total duration into multiple *subbreaks*. In most problems considered in the personnel scheduling literature, the number, duration and placement of breaks are assumed to be fixed in advance by work rules. BECHTOLD and JACOBS (1990) tried to add some flexibility by introducing the concept of break windows, i.e., by enabling a break to start within a time interval. However, the duration of each break and the position of the associated break window were assumed to be set in advance either by managers or by adopting some approach such as the ideal break start time approach (for the placement of break windows). Generally, the definition of such parameters (break length and break window placement) is done to ensure shifts of a certain quality in

the sense that periods of work and rest are well distributed. With the new concept of fractionable break, a convenient mix of work and rest periods within a shift is guaranteed by *work stretch duration restrictions*, i.e., restrictions on the duration of continuous work before and after each subbreak.

Hence, the shift scheduling approach discussed here includes almost all the flexibility alternatives considered in the shift scheduling literature to date and may address either continuous or discontinuous operating environments. A new form of flexibility, the so-called fractionable breaks, is introduced. With this new concept, the managers are no longer required to specify the number of breaks to be assigned to a shift, nor their duration or starting time. They only need to set the total duration of the breaks that must be assigned to a given shift (and possibly the number of subbreaks). The number, the position and the starting time of each subbreak is determined optimally by the model. At first sight, considering fractionable breaks adds a lot of complexity to the scheduling problem. However, we are able to model it in an efficient and simple way that enables us to reach optimal solutions in reasonable computing times. We also consider the work stretch duration restrictions in a multiple break context. Previous works on shift scheduling often define a time window within which each break can start. A correct mix of periods of work and rest is then partially ensured by choosing well spaced break windows for each shift. We introduce here a different way of optimally placing breaks within a shift without restricting a break to start within a fixed time window. Work stretch duration restrictions are modeled by the use of adapted *forward* and *backward* constraints. As initially introduced by BECHTOLD and JACOBS (1990), forward and backward constraints are usually used to model the break window concept. Relying on the results reported by ÇEZİK and GÜNLÜK (2004) and REKİK et al. (2004), we illustrate how these constraints can be extended to model another type of operating conditions (i.e., work stretch duration restrictions). Previous studies

reported the gain in model size when using forward and backward constraints to implicitly model break starting time versus the explicit formulation (i.e., generalized set covering model). We show here the merits of using the concept of forward and backward constraints versus an explicit enumerative approach to model another type of operating condition (i.e., work stretch duration restrictions).

Two implicit formulations of the problem are proposed. The first is an extension of the formulation of AYKIN (1996) and the second is inspired by the work of BECHTOLD and JACOBS (1990) in representing the break-placement flexibility. A proof of the equivalence between the LP relaxation of the two proposed models is presented. Forward and backward constraints are used intensively in both models. We propose here a general reformulation of these constraints that considerably reduces the density of the constraint matrix without increasing the number of constraints when compared to the classic forward and backward formulation. We will show, through computational experiments, that reducing the density without increasing the number of constraints results in a considerable improvement in model size and solution time. Complex real-life and academic instances including a high level of flexibility are considered. Optimal solutions are often obtained for these instances within reasonable times.

Finally, we study the impact of using fractionable breaks and work stretch duration restrictions on the total workforce size. We show, through computational experiments, that a considerable saving in workforce size can be obtained by using fractionable breaks rather than standard fixed breaks. We also show that the number of required employees is reduced when work stretch duration restrictions are used to place break windows rather than another classical approach already used in some previous studies (the ideal break start time approach).

The remainder of this paper is organized as follows. In the next section we give an overview of the shift scheduling literature by focusing on implicit models. In Section 4.3 we define the operating conditions to which the proposed models are applied and describe the new reformulation of forward and backward constraints. The extended models of AYKIN and BECHTOLD and JACOBS are also introduced. In Section 4.4 we then illustrate model characteristics for two sets of data as well as the gain in density offered by the reformulation of forward and backward constraints. The computational performance of the proposed models is then compared. We also report the impact of introducing fractionable breaks and work stretch duration restrictions on workforce size, and finally show the merits of modeling work stretch duration restrictions with forward and backward constraints.

4.2 Literature Review

Shift scheduling problems have received a lot of attention in the operations research literature. The first mathematical formulation was proposed by DANTZIG (1954). It is based on a generalized set covering model in which each shift (i.e., a combination of starting time, finishing time and break placement) is explicitly represented by a decision variable. For problems involving a high degree of flexibility with respect to shift starting times, shift lengths, or the timing of breaks, the number of enumerated shifts is very large and the resulting set covering problem may be very difficult to solve. To circumvent this difficulty, heuristic approaches have been studied extensively (see, e.g., SEGAL, 1974; KEITH, 1979; HENDERSON and BERRY, 1976). One popular strategy is to first solve a much smaller problem by relaxing some constraints or prefixing some variables. Feasible shift schedules are then constructed and improved heuristically.

Instead of focusing on solution methods, some researchers have tried to reduce model size by implicitly representing some forms of flexibility. BECHTOLD and JACOBS (1990) were the first to implicitly model the break placement for the shift scheduling problem with a single break. Instead of defining an explicit variable for each possible shift-break combination, they define *shift variables* for each possible combination of shift starting time, shift length and break window, and *break variables* for each period during which a break can take place. The matching between shifts and admissible breaks is done through a set of *forward* and *backward constraints*. This set of constraints ensures an eligible break for each shift in the associated break window without specifying its actual starting time. Complete shift schedules with appropriate break assignments are constructed *a posteriori* using a break allocation algorithm. The allocation procedure arranges shifts in a nondecreasing order with respect to the latest period in which a break may occur and assigns shifts to the earliest available breaks. BECHTOLD and JACOBS (1996) later proved that the implicit model they proposed is equivalent to the generalized set covering formulation given that break windows are composed of contiguous periods for which there is no *extraordinary overlap*. They define extraordinary overlap (E.O.) as the situation where there exist two shifts such that the break window for one shift begins strictly earlier and ends strictly later than the break window for the other shift.

AYKIN (1996) considered a more general shift scheduling problem with multiple breaks and disjoint break windows. He represented the placement of relief and lunch breaks by considering a break variable for each shift and each possible starting time within its break window. A shift is defined as the combination of a starting time, a length, the break types it contains and the associated break windows. Equality constraints are used to match each shift with the associated type of break and break window. The definition of such separate break variables, for every period for which a break is allowed for each shift, results in a general formulation applicable to all cases

to which the set covering approach is applicable. Unlike the implicit formulation of BECHTOLD and JACOBS (1990), no restrictive assumptions regarding the break windows are needed. Moreover, BECHTOLD and JACOBS (1990) considered the case where each shift receives exactly one break and all the breaks have the same duration for all shifts. This assumption is also relaxed in AYKIN's model for which the number and the duration of breaks can vary from one shift to another.

The approach presented by AYKIN (1996) extends the implicit break representation approach, introduced by GABALLA and PEARCE (1979) for the single-break context, to the context of multiple relief and lunch breaks. Although for the family of problems considered the number of variables and constraints in the model of GABALLA and PEARCE (1979) was greater than in the equivalent set covering formulation, AYKIN (1996) showed that for some instances his approach is very efficient in reducing model size with respect to the set covering formulation. AYKIN (1996) used the demand constraints to calculate upper bounds for the shift variables. These upper bounds were included in the proposed formulation and were helpful in reducing the solution time in the branch-and-bound process. In a subsequent paper, AYKIN (1998) proposed a more elaborate branch-and-cut algorithm that uses objective value cuts and dynamically computed upper bounds on shift variables. The linear relaxation of the implicit formulation proposed in AYKIN (1996) is first solved to obtain a lower bound. A rounding heuristic is then applied to the fractional solution to get a feasible schedule and thus an upper bound. Cuts are added and iteratively updated while using the rounding heuristic and a limited branch-and-bound search to find an optimal schedule. Good results were obtained for large instances. An optimal solution was found in several cases; in the remaining ones, the "best non-optimal" solution (i.e., superior by one to the lower bound) was identified.

AYKIN (2000) showed that, for the family of instances considered, his model is computationally superior to an extension of the model of BECHTOLD and JACOBS (1990) to the multiple-breaks context. REKIK et al. (2004) recently showed that the forward and backward constraints used by BECHTOLD and JACOBS (1990) are in fact particular Benders cuts that can be generated from a Benders reformulation of a transportation problem assigning breaks to shifts. Observations made by REKIK et al. (2004) suggest that, depending on the family of problems considered, either the formulation of BECHTOLD and JACOBS or that of AYKIN may lead to models that are easier to solve. Recently, ÇEZİK and GÜNLÜK (2004) used similar transportation problems to model some personnel scheduling problems and reached similar conclusions on the origin of forward and backward constraints. Their proof is based on the use of the min-cut max-flow theorem.

Implicit modeling was also adopted to handle flexibility on shift starting times and shift lengths. MOONDRA (1976) was the first to implicitly represent shifts. Two types of shifts were considered: full-time shifts with a fixed length and a lunch break window allowing two break starting times; and part-time shifts with a variable length and no lunch break. The length of the part-time shift was represented implicitly by considering shift starting time and shift finishing time variables, and using a set of constraints to impose limits on the shift duration. To handle the break placement flexibility for the full-time shifts, 50% of the full-time employees were assumed to take their lunch in the first period of the break window and the remaining employees in the second period.

THOMPSON (1995) combined the work of MOONDRA (1976) to implicitly represent shifts and the work of BECHTOLD and JACOBS (1990) to implicitly represent break placement. Each shift is assumed to receive at most one break with some restrictions on pre- and post-break work stretch durations. Shifts having the same cost per

working period, the same restrictions on minimum and maximum shift length, the same break duration and the same restrictions on the minimum and maximum pre- and post-break work stretch durations are regrouped in a same type. For each type of shift, one defines shift starting time, shift finishing time and break starting time variables. A set of constraints matching shift starting time and shift finishing time variables limit the minimum and maximum length for each shift type. Another set of constraints including shift starting time and break variables ensure the minimum and maximum pre-break work stretch durations for each shift type. Similar constraints involving shift finishing time and break variables are used to impose the minimum and maximum duration of post-break work stretch durations. THOMPSON (1995) reported that his model can be extended to a multiple-break context by defining additional relief break variables and matching shift starting time and finishing time variables to these relief break variables in the same manner as for the single-break case. A branch-and-bound procedure followed by a post-processor were used to generate actual schedules. The post-processor constructs explicit shifts and assigns breaks to them by using a first-in-first-out (FIFO) procedure. THOMPSON (1995) reported that the FIFO matching enables to bypass the E.O. problem. However, this doubly-implicit representation of shifts and breaks results, as observed by THOMPSON (1995) and AYKIN (2000), in an inability to accurately represent some cost structures.

4.3 Implicit Formulations of the Shift Scheduling Problem

This section presents two approaches of implicit modeling for a continuous shift scheduling problem that incorporates fractionable breaks, standard breaks, break windows and restrictions on pre- and post-break work stretch durations. The first approach is inspired by the work of AYKIN (1996). The second is an extension of

the formulation proposed by BECHTOLD and JACOBS (1990) and is based on the use of forward and backward constraints.

4.3.1 Problem definition

We consider a cyclic problem of a continuous 24-hour work day. As is common in shift scheduling models, the operating day is divided into *periods* of equal lengths. The set of all periods is denoted by I . The number of employees required in each period $i \in I$ is assumed to be known in advance and is denoted by d_i .

A shift is specified by a starting time, a length, the number of breaks it must receive as well as the length and the break window associated with each break. A break window consists of a set of consecutive periods at the beginning of which a break can start. The length of a break is assumed to be a multiple of that of a period. This can always be accomplished by choosing a sufficiently small length for periods. Finally, under-staffing is forbidden but could be taken into account by introducing a slack variable in each demand constraint with a penalty in the objective function (see, e.g., BAILEY, 1985).

This problem is a classical shift scheduling problem with multiple breaks and break windows. We introduce here two practical elements that offer a higher degree of flexibility: work stretch duration restrictions and fractionable breaks.

4.3.1.1 Work stretch duration restrictions

Studies on shift scheduling have often considered operating environments in which break windows are fixed in advance. BECHTOLD and JACOBS (1990), for example,

used large break windows that cover almost all the shift except the first and last periods in a single meal break context. AYKIN (1996) adopted the ideal break start time approach already used by SEGAL (1974) to set the break windows associated with relief and meal breaks. THOMPSON (1995) considered an environment in which shifts and thus associated break windows were implicitly defined by the use of minimum and maximum pre- and post-meal break work stretch durations. These constraints specify the minimum and maximum number of periods that must be worked before and after the meal break within each shift. THOMPSON (1995) considered a single break context and reported that the formulation proposed can be extended to a multiple-break context. In this paper, we consider this type of constraints in a multiple-break environment with explicit shifts and break windows. Minimum and maximum work stretch duration restrictions ensure that employees will have an appropriate mix of work and rest periods. Thus, there is no need to impose a minimum time separation between break windows when defining shifts. Consecutive breaks will be conveniently placed by minimum pre- and post-break work stretch duration constraints.

In the problem considered, we assume that minimum and maximum pre- and post-break parameters are known in advance but depend on the time interval, or *day-part*, of the operating day. Thus, to each day-part dp are associated a minimum and a maximum number of consecutive work periods denoted, respectively, by α_{dp} and μ_{dp} . The set of all day-parts is denoted by TDP (generally, there are three day-parts: day, evening and night). The parameters that apply to a given shift are those associated with the day-part covering the most periods in the shift.

4.3.1.2 Fractionable breaks

A fractionable break is a break that is not required to be attributed as a whole. It can be divided into fractions of breaks, or *subbreaks*, such that the sum of the subbreak lengths is equal to the total length of the fractionable break. The length of a subbreak is assumed to be a multiple of that of a period. In order to avoid very short or long subbreaks, minimum and maximum subbreak lengths are imposed for each fractionable break. It is also assumed that each shift receives at most one fractionable break. We show next that this problem can be modeled as a shift scheduling problem with multiple breaks and break windows.

The use of fractionable breaks provides the organization with a high degree of flexibility in defining break length and placement. We will show in Section 4.4.4 that this new flexibility alternative may result in important labor savings for some instances. In the shift scheduling literature, break lengths and break windows associated with shifts are often assumed to be fixed in advance by the union rules. Our approach allows managers to only specify the total duration of the breaks that an employee must receive without specifying which break duration comes in which position. Optimal break durations and break windows will be determined by the model. Standard break types can, however, be used to guarantee a fixed meal break length within a given time window.

To model the concept of fractionable breaks, a preprocessor can enumerate, for each shift that must receive a fractionable break, all the feasible sequences of subbreaks associated with the fractionable break. If a shift must in addition receive standard breaks (i.e., breaks that cannot be divided), all possible permutations of subbreaks and standard breaks can be enumerated. In the following, these sequences of breaks are called *break profiles*. Thus, the problem reduces to a shift scheduling

problem with multiple breaks. In order to determine the break window associated with a subbreak, one must consider the pre- and post-break work stretch duration restrictions as well as the pre-existing break windows of standard breaks. These pre-existing break windows can be simultaneously re-examined and updated for each shift. The sequence of break windows associated with the break profile of a shift is called a *window profile*.

The preprocessor can thus output a set J of all possible shifts. A shift $j \in J$ is defined for each combination of starting time, length, break profile and window profile. Recall that the break profile associated with a shift j , denoted in the following by BP_j , indicates the number and the sequence of break lengths that must be attributed to shift j . The window profile, WP_j , associated with shift j , specifies the time window in which each break can start. In the following, we make no distinction between subbreaks and standard breaks and refer to both types as breaks. The total number of breaks that must be attributed to a shift j is denoted by q_j (q_j is thus the number of elements in BP_j or WP_j). Let $BP_j(p)$ denote the length of the break that must be attributed to shift j in the p^{th} position and $WP_j(p)$ the break window associated with it. Let also TBP denote the set of all break profiles, i.e., $TBP = \cup_{j \in J} BP_j$. Finally, the day-part that maximizes the number of periods covered by shift j is denoted by dp_j .

4.3.2 Transportation problem substructure

The extended models proposed in the next section are based on the use of transportation problems to match shifts with admissible breaks, i.e., breaks that start within associated break windows and that are sufficiently separated with respect to the work stretch duration restrictions. We will show that only the

feasibility of these transportation problems is needed to ensure a correct match. As proved by ÇEZİK and GÜNLÜK (2004) and REKİK et al. (2004), under some assumptions defined in Section 4.3.2.2, the feasibility of some particular transportation problems can be ensured by a relatively small set of constraints: forward and backward constraints initially introduced by BECHTOLD and JACOBS (1990), and an equality constraint between the total supply and total demand.

This section defines the context in which forward and backward constraints are sufficient to guarantee the feasibility of a transportation problem and the parameters and sets needed to describe them.

Consider a transportation problem, $T(N_1, N_2)$, represented by a bipartite network $G = (N_1 \cup N_2, A)$ where N_1 and N_2 are the sets of supply and demand nodes, respectively. The supply of each node $i \in N_1$ is denoted by O_i and the demand of each node $j \in N_2$ is denoted by D_j . We first describe the classical flow model usually used to formulate transportation problems.

4.3.2.1 Flow model

The transportation problem described above can be modeled by the use of network flow variables Y_{ij} for each arc $(i, j) \in A$. If our objective is restricted to ensure the feasibility of such a problem, a classical flow model can be defined as follows:

$$\sum_{j:(i,j) \in A} Y_{ij} = O_i \quad \forall i \in N_1 \quad (4.1)$$

$$\sum_{i:(i,j) \in A} Y_{ij} = D_j \quad \forall j \in N_2 \quad (4.2)$$

$$Y_{ij} \geq 0 \quad \forall (i, j) \in A.$$

4.3.2.2 Forward and backward constraints

Assume that the structure of $T(N_1, N_2)$ is such that a total order relation \prec can be defined on set N_2 . Assume also that each supply node $i \in N_1$ is connected to a set of consecutive demand nodes $j \in N_2$, denoted by P_i . Finally, there exists no supply nodes i_1 and i_2 in N_1 such that $\min(P_{i_1}) \prec \min(P_{i_2})$ and $\max(P_{i_2}) \prec \max(P_{i_1})$. One can observe that the latter assumption generalizes the condition of the absence of extraordinary overlap initially defined for the shift scheduling problem. This condition is adapted here to a general transportation problem.

To describe forward and backward constraints, we define the following sets:

$$\begin{aligned}
 N_2^s &= \cup_{i \in N_1} \{\min(P_i)\} \\
 N_2^e &= \cup_{i \in N_1} \{\max(P_i)\} \\
 N_2^B(j) &= \{j' \in N_2 \mid j \preceq j'\} \quad \forall j \in N_2^s \\
 N_2^F(j) &= \{j' \in N_2 \mid j' \preceq j\} \quad \forall j \in N_2^e \\
 N_1^B(j) &= \{i \in N_1 \mid P_i \subseteq N_2^B(j)\} \quad \forall j \in N_2^s \\
 N_1^F(j) &= \{i \in N_1 \mid P_i \subseteq N_2^F(j)\} \quad \forall j \in N_2^e
 \end{aligned}$$

Finally, let $n_2^s = \min(N_2)$ and $n_2^e = \max(N_2)$. Forward and backward constraints are then given, respectively, by :

$$\sum_{j' \in N_2^F(j)} D_{j'} \geq \sum_{i \in N_1^F(j)} O_i \quad \forall j \in N_2^e \setminus \{n_2^e\}$$

and

$$\sum_{j' \in N_2^B(j)} D_{j'} \geq \sum_{i \in N_1^B(j)} O_i \quad \forall j \in N_2^s \setminus \{n_2^s\}.$$

Under the assumptions made in the beginning of this section, the feasibility of $T(N_1, N_2)$ is ensured by the use of forward and backward constraints and an equality constraint between the total supply and total demand (see ÇEZİK and GÜNLÜK, 2004 and REKİK et al. 2004):

$$\sum_{i \in N_1} O_i - \sum_{j \in N_2} D_j = 0.$$

We will see in Section 4.4.3 that forward and backward constraints as written above may yield a constraint matrix with a high density. We propose, hereafter, a reformulation of forward and backward constraints that reduces the density of the constraint matrix by adding slack variables. Recall that a total order relation \prec is defined on set N_2 . One can thus order the elements of sets N_2^e in ascending order and those of N_2^s in descending order. That is, $N_2^e = \{j_1^e, j_2^e, \dots, j_m^e\}$ where m represents the number of elements in N_2^e and $j_1^e \preceq j_2^e \preceq \dots \preceq j_m^e$. Similarly, $N_2^s = \{j_1^s, j_2^s, \dots, j_r^s\}$ where r is the number of elements in N_2^s and $j_r^s \preceq j_{(r-1)}^s \preceq \dots \preceq j_1^s$. Let j_l^e and $j_{(l+1)}^e$ be two successive elements of N_2^e such that $j_l^e \preceq j_{(l+1)}^e$. Forward constraints corresponding to j_l^e and $j_{(l+1)}^e$ can be rewritten as:

$$\sum_{j' \in N_2^F(j_l^e)} D_{j'} - \sum_{i \in N_1^F(j_l^e)} O_i = Z_{j_l^e} \quad (4.3)$$

and

$$\sum_{j' \in N_2^F(j_{(l+1)}^e)} D_{j'} - \sum_{i \in N_1^F(j_{(l+1)}^e)} O_i = Z_{j_{(l+1)}^e} \quad (4.4)$$

where $Z_{j_l^e} \geq 0$ and $Z_{j_{(l+1)}^e} \geq 0$ represent the slack variables associated with constraints (4.3) and (4.4), respectively. However, because of the structure of problem, $T(N_1, N_2)$, $N_2^F(j_l^e) \subseteq N_2^F(j_{(l+1)}^e)$ and $N_1^F(j_l^e) \subseteq N_1^F(j_{(l+1)}^e)$ as shown by

REKIK et al. (2004). Thus, equation (4.4) can be written as:

$$\sum_{j' \in N_2^F(j_{(l+1)}^e) \setminus N_2^F(j_l^e)} D_{j'} - \sum_{i \in N_1^F(j_{(l+1)}^e) \setminus N_1^F(j_l^e)} O_i + Z_{j_l^e} = Z_{j_{(l+1)}^e}.$$

Similarly, let j_l^s and $j_{(l+1)}^s$ be two successive elements of N_2^s such that $j_{(l+1)}^s \preceq j_l^s$.

Backward constraints associated with $j_{(l+1)}^s$ and j_l^s can be rewritten as:

$$\sum_{j' \in N_2^B(j_l^s)} D_{j'} - \sum_{i \in N_2^B(j_l^s)} O_i = Z_{j_l^s}$$

and

$$\sum_{j' \in N_2^B(j_{(l+1)}^s) \setminus N_2^B(j_l^s)} D_{j'} - \sum_{i \in N_1^B(j_{(l+1)}^s) \setminus N_1^B(j_l^s)} O_i + Z_{j_l^s} = Z_{j_{(l+1)}^s}$$

where $Z_{j_l^s} \geq 0$ and $Z_{j_{(l+1)}^s} \geq 0$.

4.3.3 Extension of the implicit formulation of Aykin

As mentioned in Section 4.2, AYKIN (1996) studied a shift scheduling problem with multiple breaks and multiple break windows. The implicit model that he proposed is essentially based on the use of break variables that are associated with shifts. The set of matching constraints between shifts and breaks has been shown to be very close to a transportation problem in which the break variables represent network flow variables and the shift variables represent the supply nodes (REKIK et al., 2004). Variables representing demand nodes are replaced in the demand constraints by the sum of flow variables of equation (4.2).

To extend AYKIN's approach to our context, a shift variable S_j is defined for each shift $j \in J$ and represents the number of employees working shift j . To determine the set K of all breaks (that represent demand nodes) one must consider the break profiles and the window profiles of all shifts. A break k is defined for each break profile $bp \in TBP$, each position p of a break in bp , each possible starting time t (with respect to the window profiles associated with break profile bp), and each admissible day-part dp of the operating day. A day-part dp is admissible for break k if break k can be matched with a shift j associated with dp (i.e., such that $dp_j = dp$). A break variable B_k is defined for each break k and represents the number of employees receiving break k . A network flow variable X_{jk} (representing the break variables in the original model of AYKIN (1996)) is defined for each shift j and each break k that can be matched with shift j . It represents the number of breaks k attributed to shift j . A break k is eligible for shift j if its break profile corresponds to the same break profile BP_j of shift j , if it has a position p_k and a starting time t_k such that $t_k \in WP_j(p_k)$, and, finally, if it is associated with the same day-part as shift j . In the following, the set of breaks k that can be matched with shift j in the position p , $p \in Q_j$, $Q_j = \{1, \dots, q_j\}$, is denoted by $K_{j(p)}$. Similarly, J_k will denote the set of shifts j that can receive a break k in the p_k^{th} position.

Let α_j and μ_j denote, respectively, the minimum and maximum number of periods that can be worked consecutively within shift $j \in J$. Recall that these parameters can be precomputed with respect to day-parts as explained in Section 4.3.1.1. The work stretch duration restrictions imply that two consecutive breaks attributed to shift j must be separated by at least α_j and at most μ_j periods. Consider the transportation problem $T(K_{j(p)}, K_{j(p+1)})$ as defined in Section 4.3.2 where $p \in Q_j^-$, $Q_j^- = \{1, \dots, q_j - 1\}$ and X_{jk} represents the supply/demand associated with nodes in $K_{j(p)}$ and $K_{j(p+1)}$. An arc exists between a node $k^1 \in K_{j(p)}$ and a node $k^2 \in K_{j(p+1)}$ if break k^2 starts at least α_j periods and at most μ_j periods after break k^1 ends.

Finally, a total order relation is defined on set $K_{j(p+1)}$ such that a break k is smaller than break k' if it starts before it. It can be verified that extraordinary overlap is absent and that the transportation problem $T(K_{j(p)}, K_{j(p+1)})$ possesses the same structure and satisfies all the conditions described in Section 4.3.2.2. Forward and backward constraints can thus be used to represent work stretch duration restrictions between consecutive breaks.

To describe the model, we also introduce the following parameters:

$\delta_{ij} = 1$ if shift $j \in J$ covers period $i \in I$; 0 otherwise;

$\rho_{ik} = 1$ if break k covers period $i \in I$; 0 otherwise.

The extended formulation of AYKIN (1996) is given by :

$$(P1) \quad \text{minimize} \quad \sum_{j \in J} S_j \quad (4.5)$$

subject to

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} \sum_{j \in J_k} X_{jk} \geq d_i \quad \forall i \in I \quad (4.6)$$

$$\sum_{k \in K_{j(p)}} X_{jk} - S_j = 0 \quad \forall j \in J, p \in Q_j \quad (4.7)$$

$$\sum_{k' \in K_{j(p+1)}^F(k)} X_{jk'} - \sum_{k' \in K_{j(p)}^F(k)} X_{jk'} \geq 0 \quad \forall j \in J, p \in Q_j^-, k \in K_{j(p+1)}^e \setminus \{k_{j(p+1)}^e\} \quad (4.8)$$

$$\sum_{k' \in K_{j(p+1)}^B(k)} X_{jk'} - \sum_{k' \in K_{j(p)}^B(k)} X_{jk'} \geq 0 \quad \forall j \in J, p \in Q_j^-, k \in K_{j(p+1)}^s \setminus \{k_{j(p+1)}^s\} \quad (4.9)$$

$$\sum_{k \in K_{j(p+1)}} X_{jk} - \sum_{k \in K_{j(p)}} X_{jk} = 0 \quad \forall j \in J, p \in Q_j^- \quad (4.10)$$

$$S_j \geq 0 \quad \text{and integer} \quad \forall j \in J \quad (4.11)$$

$$X_{jk} \geq 0 \quad \text{and integer} \quad \forall j \in J, k \in \cup_{p \in Q_j} K_{j(p)}. \quad (4.12)$$

The objective function (4.5) minimizes the total number of employees over the planning horizon. Demand constraints (4.6) ensure that the number of employees working in a certain period, i.e., those who are present and not on break, is at least equal to the demand for that period. Break window constraints (4.7) ensure that each shift j receives q_j breaks, one in each position $p \in Q_j$, such that the break in position p has a length compatible with the break profile of shift j and starts within $WP_j(p)$. Work stretch duration constraints (4.8)-(4.10) are forward, backward and equality constraints corresponding to transportation problems $T(K_{j(p)}, K_{j(p+1)}), j \in J, p = 1, \dots, q_j - 1$. It can be verified that equality constraints (4.10) are redundant in the presence of constraints (4.7) and can thus be removed.

4.3.4 Extension of the implicit formulation of Bechtold and Jacobs

BECHTOLD and JACOBS (1990) considered a discontinuous shift scheduling problem in which each shift must receive exactly one break within a prespecified break window and all breaks have the same duration. We propose in this section an extension of their model to a continuous environment with multiple breaks, including fractionable breaks, and work stretch duration restrictions. The proposed model ($P2$) uses shift variables S_j and break variables B_k as defined in Section 4.3.3.

Consider an extended model of ($P1$) in which variables B_k representing demand nodes are kept and constraint (4.6) is replaced with :

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} B_k \geq d_i \quad \forall i \in I \quad (4.13)$$

$$\sum_{j \in J_k} X_{jk} - B_k = 0 \quad \forall k \in K. \quad (4.14)$$

One can observe that in this extended model of (P1), the subproblem involving constraints (4.7), (4.12) and (4.14) can be separated into independent transportation problems $T(J_{(bp,dp)}, K_{(bp,dp,p)})$, one for each break profile $bp \in TBP$, each day-part $dp \in TDP$, and each possible position p of a break in bp , $p \in N_{bp}$ where $N_{bp} = \{1, \dots, n_{bp}\}$ and n_{bp} denotes the total number of breaks in bp . The set of supply nodes $J_{(bp,dp)}$ contains all shifts j associated with break profile bp and day-part dp , i.e., $J_{(bp,dp)} = \{j \in J | BP_j = bp \text{ and } dp_j = dp\}$. The supply associated with node j is S_j . The set of demand nodes $K_{(bp,dp,p)}$ contains all breaks k associated with day-part dp and occupying the p^{th} position in break profile bp , i.e., $K_{(bp,dp,p)} = \{k \in K | BP_k = bp \text{ and } dp_k = dp \text{ and } p_k = p\}$. The demand for node k is B_k . An arc (j, k) exists between a shift $j \in J_{(bp,dp)}$ and a break $k \in K_{(bp,dp,p)}$ if break k starts within the break window associated with the p^{th} position of the window profile WP_j of shift j . Breaks in $K_{(bp,dp,p)}$ are sorted in ascending order with respect to their starting time, which defines a total order relation on set $K_{(bp,dp,p)}$. We assume that there is no extraordinary overlap between breaks in $K_{(bp,dp,p)}$. Hence, for a given break profile bp , a given day-part dp and a given position p of a break in bp , the transportation problem $T(J_{(bp,dp)}, K_{(bp,dp,p)})$ can be replaced by a set of forward, backward and equality constraints as described in Section 4.3.2.2.

To model pre- and post-break work stretch duration restrictions, we consider a transportation problem $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$ for each break profile $bp \in TBP$, each day-part $dp \in TDP$ and each position p of a break in bp , $p \in N_{bp}^-$ where $N_{bp}^- = \{1, \dots, n_{bp} - 1\}$. The supply/demand of a node k is B_k . An arc (k^1, k^2) exists between a break $k^1 \in K_{(bp,dp,p)}$ and a break $k^2 \in K_{(bp,dp,p+1)}$ if break k^2 starts at least α_{dp} and at most μ_{dp} periods after break k^1 ends. Breaks in $K_{(bp,dp,p+1)}$ are sorted in ascending order with respect to their starting time. It can be verified that the condition of no extraordinary overlap, as defined in Section 4.3.2.2, is satisfied.

Thus, problems $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$, $bp \in TBP, dp \in TDP, p \in N_{bp}^-$, can be replaced by adapted forward, backward and equality constraints.

Model (P2) is thus written as follows:

$$(P2) \quad \text{minimize} \quad \sum_{j \in J} S_j \quad (4.15)$$

subject to

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} B_k \geq d_i \quad \forall i \in I \quad (4.16)$$

$$\begin{aligned} \sum_{k' \in K_{(bp,dp,p)}^F(k)} B_{k'} - \sum_{j \in J_{(bp,dp)}^F(k)} S_j &\geq 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp} \\ &\forall k \in K_{(bp,dp,p)}^e \setminus \{k_{(bp,dp,p)}^e\} \end{aligned} \quad (4.17)$$

$$\begin{aligned} \sum_{k' \in K_{(bp,dp,p)}^B(k)} B_{k'} - \sum_{j \in J_{(bp,dp)}^B(k)} S_j &\geq 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp} \\ &\forall k \in K_{(bp,dp,p)}^s \setminus \{k_{(bp,dp,p)}^s\} \end{aligned} \quad (4.18)$$

$$\sum_{k \in K_{(bp,dp,p)}} B_k - \sum_{j \in J_{(bp,dp)}} S_j = 0 \quad \forall dp \in TDP, bp \in TBP, p \in N_{bp} \quad (4.19)$$

$$\begin{aligned} \sum_{k' \in K_{(bp,dp,p+1)}^F(k)} B_{k'} - \sum_{k' \in K_{(bp,dp,p)}^F(k)} B_{k'} &\geq 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \\ &\forall k \in K_{(bp,dp,p+1)}^e \setminus \{k_{(bp,dp,p+1)}^e\} \end{aligned} \quad (4.20)$$

$$\begin{aligned} \sum_{k' \in K_{(bp,dp,p+1)}^B(k)} B_{k'} - \sum_{k' \in K_{(bp,dp,p)}^B(k)} B_{k'} &\geq 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \\ &\forall k \in K_{(bp,dp,p+1)}^s \setminus \{k_{(bp,dp,p+1)}^s\} \end{aligned} \quad (4.21)$$

$$\sum_{k \in K(bp,dp,p+1)} B_k - \sum_{k \in K(bp,dp,p)} B_k = 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \quad (4.22)$$

$$S_j \geq 0 \quad \text{and integer} \quad \forall j \in J \quad (4.23)$$

$$B_k \geq 0 \quad \text{and integer} \quad \forall k \in K. \quad (4.24)$$

Model ($P2$) considers the same objective function (4.15) and the same demand constraints (4.16) as model ($P1$). Constraints (4.17)-(4.18) are forward, backward and equality constraints corresponding to transportation problems $T(J_{(bp,dp)}, K_{(bp,dp,p)})$, $bp \in TBP, dp \in TDP, p = 1, \dots, n_{bp}$. They ensure that each shift receives the associated number of breaks and each attributed break is eligible with respect to break length (i.e., its position in the break profile) and break starting time (with respect to the associated break window). Pre- and post-break work stretch durations are modeled by forward, backward and equality constraints (4.20)-(4.22) corresponding to transportation problems $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$, $bp \in TBP, dp \in TDP, p = 1, \dots, n_{bp} - 1$. One can verify that equality constraints (4.22) are redundant in the presence of equality constraints (4.19) and can thus be removed.

4.4 Computational Experiments

The focus of this section is to compare models ($P1$) and ($P2$) through computational experiments on a large set of instances. Two sets of data are considered: real-life instances arising from an air-traffic control agency and generated instances obtained by varying some scheduling parameters in the first set. Our objectives in developing the second set of instances are to compare model performance under various operating conditions and to measure the impact of considering fractionable breaks and work stretch duration restrictions on total workforce size.

4.4.1 Scheduling environments

The scheduling environments used for this analysis consist in a continuous 24-hour (cyclic) operating day divided into 96 periods of 15 minutes.

The first set of instances (SET1) includes three different operating environments. Each environment is characterized by a list of admissible shift types and work stretch parameters. A shift type is specified by a starting time, a length and the fractionable and/or unfractionable breaks it must receive. Instances provided by the company involve up to 21 shift types and almost all shift types include a unique fractionable break. For the first operating environment, all shift types include only fractionable breaks. For the two remaining operating environments, only one shift type incorporates a fixed break. In terms of work stretch parameters, all the instances consider three day-parts and thus three different values of minimum and maximum work stretch parameters during the day. Two demand patterns are specified for the first environment, three are specified for the second environment and, finally, two are specified for the third environment.

The second set of instances, SET2, includes eight operating environments. All the environments consider nine-hour shifts that must receive only one fractionable break. For six of these environments, break profiles associated with each shift type are restricted to those having exactly three subbreaks such that the length of the subbreak in the middle is strictly greater than the first and last ones. This restriction ensures realistic operating conditions. For the two remaining environments, a unique break profile is considered. For all operating environments, the minimum and maximum work stretch parameters, denoted hereafter by α and μ , respectively, are assumed to have the same value throughout the day. Table 4.1 describes the eight scheduling environments of SET2. In this table, P_{tot} , P_{min} , and P_{max} denote, respectively, the total fractionable break length and the minimum and maximum subbreak lengths. In the cases where only one break profile is considered (as for environments #6 and #7), the sequence of subbreak durations is specified. All values in Table 4.1 are given in minutes. For example, operating environment #1 includes nine-hour shifts starting every 30 minutes. Each shift must receive

a two-hour fractionable break with a minimum subbreak length of 15 minutes and a maximum subbreak length of 90 minutes. Finally, the minimum and maximum work stretch durations are set to 60 minutes and 180 minutes, respectively. For each of these operating environments, 11 demand patterns are considered. The first seven patterns correspond to the air-traffic control agency requirements of SET1. The last four are reported in BARD et al. (2003) and were obtained from the United State Postal Service in Oklahoma City. The 11 demand patterns used for experimental analysis are illustrated in the Appendix. Each operating environment is applied to the 11 demand patterns, yielding a total of 88 instances.

Table 4.1 Description of the scheduling environments of SET2

Scheduling environment	Shift starting time	$(P_{tot}, P_{min}, P_{max})$	Number of break profiles	(α, μ)
#1	30	(120, 15, 90)	6	(60, 180)
#2	15	(120, 15, 90)	6	(60, 180)
#3	60	(120, 15, 90)	6	(60, 180)
#4	30	(120, 15, 90)	6	(60, 240)
#5	30	(90, 15, 60)	3	(60, 180)
#6	30	(120, 15, 90)	1 : (15, 90, 15)	(60, 180)
#7	60	(120, 15, 90)	1 : (15, 90, 15)	(60, 180)
#8	15	(120, 15, 90)	6	(60, 240)

4.4.2 Problem size comparison of models

AYKIN (2000) has already compared his formulation to an extension of the model of BECHTOLD and JACOBS (1990) for a shift scheduling problem with multiple breaks and multiple break windows. He reported that, for the instances considered, although the formulation of BECHTOLD and JACOBS (1990) requires fewer variables, the model of AYKIN (1996) requires fewer constraints and yields a lower density of the constraint matrix. In the following, we show that for all the instances considered in

this paper, model ($P1$) keeps its advantage over model ($P2$) of a considerably lower constraint matrix density. We will also show that the reformulation of forward and backward constraints proposed in Section 4.3.2.2 considerably reduces the density of the constraint matrix of both models.

Let ($P3$), respectively ($P4$), denote the mixed-integer models associated with model ($P1$), respectively ($P2$), where all the forward and backward constraints (i.e., constraints (4.8) and (4.9) for ($P1$) and constraints (4.17), (4.18), (4.20) and (4.21) for ($P2$)) are reformulated with slack variables as described in Section 4.3.2.2. Tables 4.2 and 4.3 report the number of variables, number of constraints and the density of the constraint matrix (in percentage) of formulations ($P1$), ($P2$), ($P3$) and ($P4$) for instances in SET1 and SET2, respectively. One can see from these tables that model ($P2$) possesses a tremendous problem-size advantage over ($P1$) for the environments involving a high degree of flexibility. For example, for environment #8, model ($P1$) requires 23,040 variables and 15,648 constraints compared with 2520 variables and 5958 constraints for model ($P2$). But, for all the instances considered, ($P1$) keeps its primary advantage over ($P2$) of a lower density. In fact, for the three environments of SET1 and some environments of SET2 (#5, #6, and #7), the constraint matrix associated with ($P2$) is much denser reaching 24.79% for environment #7 of SET2 versus 1.81% for ($P1$). It is worth noting that these environments correspond to the less flexible ones. The reformulation of forward and backward constraints in models ($P3$) and ($P4$), although it increases the number of variables in comparison with models ($P1$) and ($P2$) (a continuous slack variable is added for each forward and each backward constraint), yields a constraint matrix with a significantly lower density. Moreover, the density of model ($P4$) becomes comparable with that of model ($P1$) and even slightly lower for all the operating environments of SET1 and for six environments over the eight considered in SET2 (75%). However, when we reformulate the forward and backward constraints of model ($P1$) with slack

variables, the density of the resulting model (*P3*) becomes once again substantially lower than that of model (*P4*) for almost all the scheduling environments. Although model (*P1*) initially has a relatively low density, we reformulate the associated forward and backward constraints to see the impact of such a reformulation on its performance. In fact, in the next section, we will see that the reformulation of forward and backward constraints with slack variables considerably improves the computational performance of both the AYKIN and BECHTOLD and JACOBS models. Thus, in the following, we use the more efficient formulations (*P3*) and (*P4*) to compute solution times for the extended models of AYKIN and BECHTOLD and JACOBS.

Table 4.2 Comparison of model size for SET1

Env.	Number of variables				Number of constraints				Constraint matrix density			
	(P1)	(P2)	(P3)	(P4)	(P1)	(P2)	(P3)	(P4)	(P1)	(P2)	(P3)	(P4)
#1	832	404	1377	1097	810	803	810	803	1.10	8.06	0.63	0.73
#2	1486	890	2428	2136	1295	1376	1295	1376	0.70	3.23	0.38	0.42
#3	1883	1101	3182	2959	1794	1994	1794	1994	0.54	2.96	0.29	0.28

Table 4.3 Comparison of model size for SET2

Env.	Number of variables				Number of constraints				Constraint matrix density			
	(P1)	(P2)	(P3)	(P4)	(P1)	(P2)	(P3)	(P4)	(P1)	(P2)	(P3)	(P4)
#1	9216	2166	16128	6258	7872	4206	7872	4206	0.17	4.46	0.05	0.14
#2	18432	2472	32256	8316	15648	5958	15648	5958	0.08	4.39	0.03	0.11
#3	4608	1986	8064	5166	3984	3294	3984	3294	0.33	4.73	0.12	0.16
#4	11520	2214	18432	6306	7872	4206	7872	4206	0.16	3.15	0.05	0.14
#5	4896	1089	8928	3147	4560	2163	4560	2163	0.31	8.72	0.09	0.26
#6	1536	361	2688	1043	1392	781	1392	781	0.97	24.01	0.35	0.75
#7	768	331	1344	861	744	629	744	629	1.81	24.79	0.66	0.86
#8	23040	2520	36864	8364	15648	5958	15648	5958	0.08	4.40	0.03	0.11

4.4.3 Solution time comparison of models

The branch-and-bound algorithm of CPLEX 8.1 was applied to both formulations (*P3*) and (*P4*) on a 1.8 GHz Pentium III PC with an upper time limit of 3600

seconds. In all experiments reported here, the built-in heuristic of CPLEX for finding a feasible solution was disabled. Indeed, we applied this heuristic to some instances and observed that it increased the average solution time for both formulations with a substantial increase for $(P4)$. For the same reasons, the automatic generation of cuts was disabled. BRUSCO and JACOBS (2000) reported that the lower bound provided by the objective function value may be used to add a cut on the integer objective function. They reported that this cut generally improved the branch-and-bound search process. In our case, however, adding such a cut deteriorated the performance for almost all instances. We also observed that the preprocessing (presolve) of CPLEX increased the solution times for model $(P4)$ and had no effect upon model $(P3)$. We thus decided not to use it.

As mentioned by AYKIN (2000), both BECHTOLD and JACOBS (1990) and AYKIN (1996) observed that the break variables in their formulations tend to take integer values when the shift variables are integer. Based on this observation, a branching strategy giving higher priority to shift variables S_j was considered. We performed a best-bound search with a minimum integer infeasibility criterion (variable closest to an integer) to select the branching variable.

Recall that AYKIN (2000) observed that his formulation and that of BECHTOLD and JACOBS (1990) yield the same LP objective value for all the instances considered. The equivalence of the linear relaxation of the two formulations was demonstrated in REKIK et al. (2004). Moreover, the linear relaxation of the extended formulations presented in this paper are also equivalent and both integer models, $(P3)$ and $(P4)$, will always result in the same integrality gap.

We first study the impact of reformulating forward and backward constraints with slack variables. Table 4.4 reports the LP and IP solution times required by models

(*P1*), (*P2*), (*P3*) and (*P4*) with the operating conditions of environment #1 of SET2. Clearly, reducing the number of nonzeros and thus the density of models (*P1*) and (*P2*) results, on average, in a tremendous time saving. When considering the AYKIN approach, the reformulation of forward and backward constraints in model (*P3*) yields an average solution time of 781.63 seconds compared to 1264.15 seconds for model (*P1*). Moreover, for the unique instance that remained unsolved by model (*P1*) within the time limit, model (*P3*) was able to find an optimal solution in 818.81 seconds. For the BECHTOLD and JACOBS approach, one can see from Table 4.4 that models (*P2*) and (*P4*) yielded an optimal solution within the time limit for ten instances out of the 11 considered. For the unique instance that was not solved to optimality by models (*P2*) and (*P4*), both models yield the same integer feasible solution. For the ten instances solved to optimality, model (*P2*) uses on average 666.34 seconds compared to only 219.40 seconds for model (*P4*). It is worth recalling that for the AYKIN approach, the number of nonzero elements of the constraint matrix decreases from 122,304 nonzeros for model (*P1*) to 63,480 nonzeros for model (*P3*). For the BECHTOLD and JACOBS approach, this reduction is much more considerable passing from 406,090 nonzeros for model (*P2*) to only 36,562 nonzeros for model (*P4*).

In the remainder of the experiments, we thus focus on models (*P3*) and (*P4*). The results obtained with these two models are reported in Tables 4.4.3 and 4.4.3. These tables display the average integrality gap as well as the average solution times (in seconds) needed by both formulations to find the LP and IP solutions for each operating environment of SET1 and SET2. Recall that, contrarily to SET2, not all the demand patterns were considered for SET1. Thus, we add a column in Table 4.4.3 to specify the number of instances considered for each environment. Two different averages are reported for the integer solution: the first average, Avg^1 , is computed for all the instances of a given operating environment even if no optimal

Table 4.4 Solution times for models (P1) – (P4) for environment #1 of SET2

Demand pattern	LP solution time (s.)				IP solution time (s.)			
	(P1)	(P2)	(P3)	(P4)	(P1)	(P2)	(P3)	(P4)
1	28.41	7.73	77.18	5.56	1926.10	1482.34	686.96	196.17
2	26.85	10.40	119.55	6.67	3600.00	1008.65	818.81	151.70
3	40.11	11.10	34.30	10.20	406.12	734.21	375.85	292.21
4	43.25	22.25	26.75	10.69	901.69	465.38	487.13	340.09
5	9.70	6.55	26.75	4.15	2110.36	3600.00	3314.98	3600.00
6	68.27	10.42	52.83	4.42	1255.68	1477.11	660.96	246.84
7	46.64	5.52	97.23	5.86	1007.49	519.48	794.42	243.18
8	36.50	48.28	103.21	6.78	538.79	225.52	215.49	242.53
9	93.21	5.39	59.40	4.22	1032.58	376.97	465.48	178.71
10	41.83	38.26	106.61	5.72	354.82	244.36	500.04	131.73
11	35.96	8.74	181.36	5.12	772.04	129.37	277.78	170.80
Average	42.79	15.88	84.07	6.32	1264.15	933.04	781.63	526.72

solution was found within the time limit. In this case, we consider a solution time equal to 3600 seconds, i.e., the time limit. The second average, Avg^2 , is computed only for the instances that were solved optimally within the time limit with both formulations. A dash ('-') in these columns indicates that no instance could be solved optimally by both models.

Table 4.5 Computational results for SET1

Env.	Number of instances	Gap	Average LP solution time (s.)		Average IP solution time (s.)			
			(P3)	(P4)	Avg^1		Avg^2	
			(P3)	(P4)	(P3)	(P4)	(P3)	(P4)
#1	2	1.67	0.24	0.23	0.35	0.34	0.35	0.34
#2	3	2.47	0.68	0.54	901.12	1845.87	0.51	968.81
#3	2	1.79	0.76	2.07	1.50	8.68	1.50	8.68

One can see from Table 4.4.3 that for SET1, formulations (P3) and (P4) are almost equivalent for operating environment #1. For operating environment #3, model (P4) needs on average 8.68 seconds compared to only 1.5 second for model (P3).

Table 4.6 Computational results for SET2

Env.	Gap	Average LP		Average IP solution time (s.)			
		solution time (s.)		Avg^1		Avg^2	
		(P3)	(P4)	(P3)	(P4)	(P3)	(P4)
#1	2.47	84.07	6.32	781.63	526.72	528.29	226.92
#2	1.93	523.67	12.62	3470.00	897.43	2169.96	1582.83
#3	2.81	9.29	5.86	446.64	488.44	131.30	177.29
#4	1.93	299.21	6.20	2073.66	332.69	2073.66	332.69
#5	1.16	9.85	1.95	138.98	42.15	138.98	42.15
#6	2.21	1.00	0.43	29.08	9.39	29.08	9.39
#7	2.27	0.26	0.24	3.79	2.33	3.79	2.33
#8	1.93	1339.80	15.23	3512.78	1097.01	—	—

Nevertheless, both solution times remain relatively small. A big difference between average solution times is observed for operating environment #2 where the average IP solution time, Avg^2 , is substantially smaller for (P3). In fact, there are two instances over three that were optimally solved by both formulations. For one of these instances the CPU time required by (P4) is slightly greater than that required by (P3). Thus, the large difference between average solution times comes from only one instance that was solved by (P3) in 0.54 seconds and required 1936.43 seconds with (P4).

For SET2, one can assert that, for the instances considered (88 in total), model (P4) is much more efficient than model (P3) in terms of the solution time and the number of instances solved to optimality. In fact, as reported in Table 4.4.3, for five operating environments over the eight considered, the average integer solution time required by (P4) is considerably smaller than that needed by model (P3) for both Avg^1 and Avg^2 . For the remaining environments (#3 and #7), the average solution times are similar with a slight advantage for (P3) with scheduling environment #3 and for (P4) with scheduling environment #7. When considering all instances, model (P4) requires 424.52 seconds on average compared to 1307.07 seconds for

model ($P3$). For the instances that were solved optimally by both formulations (65 in total), model ($P4$) needed on average 338.01 seconds compared to 725.01 seconds for model ($P3$). For 53 of these instances (82%), model ($P4$) required less time than model ($P3$). Moreover, over the 88 instances, only three instances were not solved to optimality by model ($P4$) within the time limit. This number reaches 21 instances with model ($P3$). For all the instances where model ($P4$) was not able to locate an optimal integer solution, a good feasible solution was found, i.e., a solution that is superior by one employee to the lower bound obtained by rounding up the value of the LP solution. Contrary to model ($P4$), model ($P3$) failed in locating any integer feasible solution for 20 instances out of the 21 that remained unsolved. One can observe that the higher the flexibility of the scheduling environment, the more efficient model ($P4$) becomes compared to model ($P3$).

It is worth recalling that the use of forward and backward constraints in model ($P4$) to place breaks within associated break windows assumes that break windows are composed of consecutive periods and that no extraordinary overlap exists between break windows. For the work stretch duration restrictions, one can in fact verify that extraordinary overlap (as defined in Section 4.3.2) cannot occur. The condition of continuous intervals must, however, remain for model ($P4$) and can be removed for model ($P3$) (see REKIK et al. (2004) for more details). These considerations give a little advantage to model ($P3$) since there is no need to impose any restriction on break windows. Nevertheless, ADDOU and SOUMIS (2004) have recently proposed an implicit model for a classical shift scheduling problem including extraordinary overlap. Their model is based on the use of forward and backward constraints with a relatively small number of additional constraints to support extraordinary overlap.

4.4.4 Impact of fractionable breaks on workforce size

To study the impact of fractionable breaks on workforce size, we consider environments #1 and #6 of SET2. Recall that for environment #1 a fractionable break of 120 minutes must be given to each shift with a minimum and a maximum subbreak length of 15 and 90 minutes, respectively. Break profiles are also restricted to those having exactly three subbreaks with a longer subbreak in the middle. One can verify that this results in six different break profiles. Environment #6 includes the same operating conditions except that a unique break profile is specified. A break profile composed of a 90-minute middle break preceded and followed by 15-minute breaks is considered.

We report in Table 4.7 the total number of employees required for environments #1 (i.e., with all the six break profiles) and #6 (i.e., with a unique break profile) for the 11 demand patterns described in Section 4.4.1. We also report the percentage savings in terms of the number of employees in environment #1 compared with environment #6. One can observe that a gain was obtained with demand patterns 2 and 5 while no gain was obtained for the other demand patterns. This results in an average saving of 1.46%. One may observe that for many of these patterns, the number of employees remains constant during a long period (see, e.g., pattern 1). We also noticed that for environment #6, where only the break profile (15, 90, 15) is used, there is no over-staffing in the optimal schedule for any of the nine demand patterns for which no saving was obtained. We thus tried to slightly perturb some of these demand patterns (five ones) in an attempt to obtain an over-staffing for environment #6 (i.e., with a unique profile). A bigger perturbation was also performed on demand pattern 1 to obtain a greater over-staffing for environment #6 and see the impact of fractionable breaks with such a perturbation. The new perturbed demand patterns are illustrated in the Appendix. Table 4.4.4 reports the results obtained with the five

perturbed demand patterns. One can notice that a significant gain is obtained for all the perturbed demand patterns resulting in an average saving of 5.89%. This saving reaches 8.7% for the more perturbed demand pattern 1. Moreover, we observed that whenever an overstaffing is obtained with the unique break profile (15, 90, 15), the use of all six break profiles reduces at maximum the overstaffing.

Table 4.7 Impact of fractionable breaks with original demand patterns

Demand pattern	Number of employees		Savings (%)
	All profiles	One profile	
1	23	23	—
2	25	26	4.00
3	15	15	—
4	18	18	—
5	16	17	6.25
6	22	22	—
7	23	23	—
8	73	73	—
9	28	28	—
10	97	97	—
11	107	107	—

Table 4.8 Impact of fractionable breaks with perturbed demand patterns

Perturbed demand pattern	Number of employees		Savings (%)
	All profiles	One profile	
1	23	25	8.70
3	15	16	6.67
4	18	19	5.56
6	24	25	4.17
7	23	24	4.35

4.4.5 Impact of work stretch duration restrictions

In this section, we first show that using the concept of work stretch duration restrictions consists in another approach to break window placement that not only

results in high-quality schedules but also yields a lower workforce size when compared to the ideal break start time approach. Second, we show that using adapted forward and backward constraints as in model ($P4$) represents the most efficient approach to model such an operating condition. In the following, one must distinguish between two terms: work stretch duration restrictions and work stretch constraints. By work stretch duration restrictions we mean the operating condition that must be considered in the scheduling environment. By work stretch constraints we mean the forward and backward constraints used in formulation ($P4$) to model the work stretch duration restrictions.

4.4.5.1 Work stretch duration restrictions approach versus ideal break start time approach

Consider operating environment #6 with the unique break profile (15, 90, 15) (the concept of fractionable break is not tested here). For this environment a minimum work stretch duration of 60 minutes and a maximum of 180 minutes must be respected for each shift. Consider also a new operating environment that incorporates the same operating conditions except the work stretch duration restrictions. The ideal break start time approach is used for this new environment to place break windows associated with the three breaks. This approach, as explained in AYKIN (1996), specifies an ideal starting time for each break. Break windows are then placed by setting the earliest break starting time half an hour earlier than the ideal break starting time. Finally, the break window length sets the latest time at which a break can occur. The ideal break starting time for the first break is commonly set at two hours after the start of the shift, the ideal start time for the second break is set at four hours of work plus the first break length, that is, in our context, four hours and 15 minutes after the start of the shift. Finally, the ideal

starting time for the last break is set at six hours plus the first and middle break lengths, that is, in our context, seven hours and 45 minutes after the start of the shift.

To compare the two approaches, we consider three combinations of break window lengths for the ideal break start time approach. The first and second combinations consider the same break window length for the three breaks, five break starting times for the first combination and seven break starting times for the second one. The third combination considers a break window length of five periods for the first and last break and a three-period break window for the middle break.

Table 4.4.5.1 reports the number of employees needed for the 11 demand patterns when considering both approaches as well as the percentage of savings in terms of the number of employees when considering the work stretch duration restriction approach versus the ideal break start time approach. In this table, a triplet is used to specify break window lengths associated with the three breaks. The length is given in terms of the number of periods at which a break can start.

One can see that the work stretch duration restriction approach results in an average gain of 10.05% for the break windows length combination (5, 3, 5) with a considerable gain of 13.33% for demand pattern 3. An average gain of 5.68% is obtained for the break windows length combination (5, 5, 5) with a maximum gain of 9.09% for demand pattern 6. The average savings fall to 2.07% for the last combination (7, 7, 7) but remain important for some demand patterns reaching 5.56% for demand pattern 4. In total, when compared to the ideal break start time approach, the work stretch duration restriction approach reduces the number of employees needed for 26 instances over the 33 considered in this study. For the remaining instances, both approaches yield the same workforce size.

Moreover, the use of work stretch duration restrictions guarantees schedules of high quality in terms of break placement. This is not necessarily the case with the ideal break start time approach, in particular when break windows are relatively large. It is worth mentioning that considering larger break windows for the ideal break start time approach would probably further reduce the number of employees needed. Besides, many studies (see e.g., AYKIN (1996), BRUSCO and JACOBS (2000)) concluded that the break window length plays a major role in reducing the total number of employees. But, they also reported that there is a break window length limit after which no improvement is obtained. Furthermore, the larger the break windows, the more likely it is that worse schedules will be obtained in terms of break placement.

Table 4.9 Number of employees with the work stretch duration restriction approach and the ideal break start time approach

Demand pattern	Work stretch duration restr. approach	Ideal break start time approach			Savings %		
		(5, 3, 5)	(5, 5, 5)	(7, 7, 7)	(5, 3, 5)	(5, 5, 5)	(7, 7, 7)
1	23	25	24	24	8.7	4.35	4.35
2	26	28	27	26	7.69	3.85	—
3	15	17	16	15	13.33	6.67	—
4	18	20	19	19	11.11	5.56	5.56
5	17	19	18	17	11.76	5.88	—
6	22	24	24	23	9.09	9.09	4.55
7	23	25	24	23	8.7	4.35	—
8	73	77	75	74	5.48	2.74	1.37
9	28	28	28	28	—	—	—
10	97	103	100	98	6.19	3.09	1.03
11	107	113	111	108	5.61	3.74	0.93
Average					10.05	5.68	2.07

4.4.5.2 Modeling work stretch duration restrictions

Consider again environment #6 for which all employees must work at least 60 minutes and no more than 180 minutes consecutively. Recall that each employee must also receive two 15-minute breaks and a 90-minute break in the middle. Two approaches for modeling work stretch duration restrictions are tested here. The first approach consists of using forward and backward constraints as in model (P4). The second approach enumerates all admissible break windows placement alternatives and defines a shift variable for each possible shift starting time and each admissible break window placement. A break window placement is admissible if each of the three breaks can start at any time within the associated break window without violating work stretch duration restrictions. Thus, the second approach considers the problem as a general shift scheduling problem with multiple breaks and multiple break windows that are placed as previously described. Each shift is thus replicated as many times as there exist admissible break window placements. Obviously, the number of break placement alternatives depends, among other things, on the position and the length of the break window associated with the first break. In our context (environment #6), there exist 45 admissible break window placements for each shift. In the following, we will refer to the first approach as the implicit approach and to the second one as the enumerative approach.

The shift scheduling problem considered by the enumerative approach being classical, it can be modeled either with the AYKIN formulation or with an extension of the BECHTOLD and JACOBS formulation as reported by AYKIN (2000). We consider here both models with a slight modification in BECHTOLD and JACOBS formulation. In fact we reformulate forward and backward constraints used in AYKIN (2000) with slack variables as described in Section 4.3.2.2. It is worth mentioning that the formulation reported by AYKIN (1996) corresponds to model

(*P3*) from which constraints (4.8)-(4.10) associated with the work stretch duration restrictions are removed. Similarly, the modified formulation of BECHTOLD and JACOBS corresponds to model (*P4*) without constraints (4.20)-(4.22). Table 4.4.5.2 displays the number of employees and the integer solution times associated with the implicit and the enumerative approaches. For the enumerative approach, these results are reported for both the AYKIN formulation and the modified formulation of BECHTOLD and JACOBS. The average integer solution time, Avg^1 , is also computed for each approach. A dash ('-') indicates that no integer feasible solution was identified within the time limit. This was the case for the AYKIN formulation with six instances. Thus, no average solution time was computed for this formulation in the enumerative approach. For the instances that were not solved to optimality, the solution time is set to the time limit (i.e., 3600 seconds).

Table 4.10 Number of employees and solution times for both approaches

Demand pattern	Number of employees			Solution time (s.)		
	Implicit approach	Enumerative approach		Implicit approach	Enumerative approach	
		Aykin	Bechtold and Jacobs		Aykin	Bechtold and Jacobs
1	23	24	24	27.95	3600.00	3600.00
2	26	26	26	2.36	976.29	13.05
3	15	16	15	2.87	3600.00	449.74
4	18	19	19	44.08	3600.00	3600.00
5	17	17	17	3.12	639.20	365.56
6	22	—	22	11.35	—	354.88
7	23	—	23	3.41	—	93.63
8	73	—	73	3.62	—	105.24
9	28	—	28	1.61	—	55.92
10	97	—	97	1.84	—	59.53
11	107	—	107	1.08	—	38.40
			Avg^1	9.39	—	794.18

As one can see from Table 4.4.5.2, the implicit approach has a significant advantage over the enumerative approach in terms of solution times. In fact, all the instances were solved to optimality in relatively short times with an average of 9.39 seconds for the implicit approach (model ($P4$)). However, for the enumerative approach, using the AYKIN formulation yielded an optimal solution for two instances over the 11. Moreover, for seven instances over the nine not solved to optimality, the formulation of AYKIN failed to find a feasible integer solution. When considering the modified formulation of BECHTOLD and JACOBS, only two instances were not solved to optimality within the time limit. However, for the instances solved to optimality, the modified formulation of BECHTOLD and JACOBS required much more time than the implicit approach, i.e., 170.66 seconds on average compared to 3.47 seconds for the implicit approach. This substantial difference in solution times comes from the fact that each shift variable present in model ($P4$) is replicated 45 times (the number of admissible break window placements) in both the AYKIN and BECHTOLD and JACOBS formulations. This yields 2160 shift variables compared to only 48 shift variables for the implicit approach.

The computational results reported in Table 4.4.5.2 for the enumerative approach also demonstrate that reformulating forward and backward constraints with slack variables results in a BECHTOLD and JACOBS formulation that is more efficient than the AYKIN formulation. Thus, conclusions made by AYKIN (2000) cannot be generalized to all instances. In fact, for the operating environment considered, the formulation of AYKIN required 27,840 variables and 6576 constraints with a constraint matrix density of 0.11% compared to 3069 variables and 699 constraints for the modified formulation of BECHTOLD and JACOBS with a density of 4.67%. Although the density of the AYKIN model remains substantially smaller than the density of the BECHTOLD and JACOBS model, the large size of the former model limits its performance when compared to the latter. Indeed, the number of nonzeros

in the constraint matrix of the AYKIN model reaches 199,200 in comparison with 100,081 for the BECHTOLD and JACOBS model.

In order to reduce problem size for the enumerative approach, we also studied the possibility of considering only some break window placement possibilities instead of all 45 possibilities. Different problems were considered. We report here detailed results for a problem, referred to as problem 1, where only one break-placement possibility is considered. For this problem, the first break must start at least 90 minutes and at most 150 minutes after the start of the shift, the second break must start at least three hours and 45 minutes and at most four hours and 45 minutes after the start of the shift and finally the last break must start at least seven hours and 15 minutes and at most seven hours and 45 minutes after the start of the shift. One can verify that the break windows as placed satisfy the work stretch duration restrictions of 60 and 180 minutes. Problem 1 was modeled with both the AYKIN formulation and the modified formulation of BECHTOLD and JACOBS. The two formulations always yield the same solution value. Table 4.4.5.2 reports the number of employees obtained by both formulations within the time limit and recalls the optimal solution obtained with model (*P4*). It also reports the percentage of employees saved with model (*P4*) versus the AYKIN formulation and the modified formulation of BECHTOLD and JACOBS for problem 1. IP solution times needed by each formulation are also specified as well as the average solution time, *Avg*¹.

As one can see from Table 4.4.5.2, considering only one break window placement alternative significantly reduces solution times when compared to the use of all the 45 possibilities. However, it results in a worse objective solution with a higher number of employees. When using model (*P4*), i.e., modeling the work stretch duration restrictions with forward and backward constraints, the smallest number of employees is obtained with reasonable solution times. On average, using the

Table 4.11 Number of employees and solution times for the implicit approach and problem 1

Demand pattern	Number of employees			Solution time (s.)		
	Implicit approach	Problem 1	Savings %	Implicit approach	Problem 1	
					Aykin	Bechtold and Jacobs
1	23	24	4.35	27.95	1.79	6.91
2	26	27	3.85	2.36	0.35	0.50
3	15	16	6.67	2.87	0.49	0.37
4	18	19	5.56	44.08	2.98	0.83
5	17	18	5.88	3.12	0.58	0.49
6	22	24	9.09	0.23	0.23	0.4
7	23	24	4.35	0.27	0.27	0.64
8	73	76	4.11	0.17	0.17	0.63
9	28	28	—	0.52	0.52	0.96
10	97	101	4.12	0.15	0.15	0.36
11	107	111	3.74	0.79	0.79	0.34
Average solution times (Avg^1)				9.39	0.76	1.13

implicit approach saves 4.7% in the number of employees when compared to the enumerative approach of problem 1. These savings were predictable since the feasible region associated with the implicit approach is larger than that relative to problem 1 of the enumerative approach when modeled with either the AYKIN formulation or the modified formulation of BECHTOLD and JACOBS. Figure 4.4.5.2 illustrates the feasible region, in terms of break starting times, associated with model ($P4$) and the model of AYKIN applied to problem 1. To simplify the figure, we consider a less flexible operating environment where the last 15-minute break no longer starts within a time interval, as in environment #6, but exactly seven hours and 30 minutes after the start of the shift. Two axis representing the starting times associated with the first and second breaks are considered.

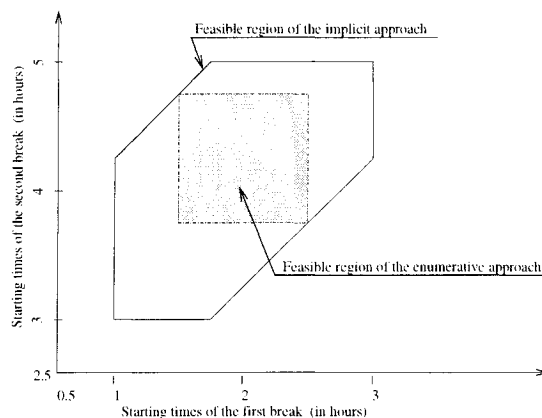


Figure 4.1 Feasible regions associated with the implicit and the enumerative approaches

As mentioned at the beginning of this section, different problems were considered for the enumerative approach with one or more break windows placement possibilities. For all these problems, the implicit approach was superior when considering solution values versus solution times. For problems with a large number of break window placement possibilities the solution values became equal to those obtained with the enumerative approach for almost all demand patterns but they required a considerably longer time. In fact, we observed that considering just five break window placement alternatives results in an average solution time that exceeds that required by the implicit approach.

4.5 Conclusion

This paper has considered a continuous shift scheduling problem that includes, to the best of our knowledge, the highest degree of flexibility ever considered in the shift scheduling literature. In addition to the classical shift scheduling flexibility alternatives (i.e., shift starting time and length, multiple breaks and break windows),

we have incorporated two other forms of flexibility: fractionable breaks and work stretch duration restrictions. This provides the possibility of fixing only the total duration of breaks that must be given within a shift without specifying which break length comes in which position. Optimal subbreak lengths and positions are determined by the proposed models. Standard break types with fixed break windows can, however, be used to ensure, for example, a fixed meal break length within a fixed time window. Moreover, pre- and post-break work stretch duration restrictions ensure high-quality shifts where periods of work and rest are well distributed. These parameters can take different values depending on the time interval of the operating day which is, in practice, more realistic since the workload is not the same all day.

We have studied the impact of considering these two operating conditions and showed that the use of fractionable breaks may yield, for some instances, a considerable workforce saving that reached 8.7% (in terms of the number of employees). We also demonstrated that the use of pre- and post-break work stretch duration restrictions is an efficient approach to place break windows within a shift. This approach not only ensures high-quality shifts but also results in a total workforce size smaller than that obtained with the ideal break start time approach. These labor savings reached 11.76% for some instances.

Considering the fractionable break concept adds a lot of complexity to the scheduling problem. However, we were able to model it in a simple and efficient way that reduces the problem to a shift scheduling problem with multiple standard breaks, break windows and work stretch duration restrictions. Two main approaches of implicit modeling were considered. The first approach extends the model of AYKIN (1996) to the context of work stretch duration restrictions. The latter is modeled with adapted forward and backward constraints. The second approach is inspired by the work of BECHTOLD and JACOBS (1990) and uses forward and backward constraints

to place breaks within associated break windows and to model work stretch duration restrictions. For both models, we proposed a reformulation of forward and backward constraints that reduces the density of the constraint matrix. This reformulation uses a continuous slack variable for each forward and each backward constraint without increasing the total number of constraints. We tested these models on a large set of instances and concluded that the proposed reformulation of forward and backward constraints results in a constraint matrix with a considerably reduced density in comparison with original models. This results in a significant improvement of solution times.

This gain in the constraint matrix density motivated us to reconsider the results reported by AYKIN (2000). In fact, AYKIN (2000) has compared his model to an extended formulation of BECHTOLD and JACOBS (1990) for a less flexible shift scheduling problem that did not include fractionable breaks and work stretch duration restrictions. He concluded that his model was computationally superior to that of BECHTOLD and JACOBS for almost all the instances considered and observed that this may be due to the substantial difference between model densities. In this paper, we have considered similar operating environments with more alternative shifts and shown that for these instances the model of BECHTOLD and JACOBS performs better than that of AYKIN although it is much denser. In this case, the low number of variables and constraints in the BECHTOLD and JACOBS model played a prominent role in determining model performance. In conclusion, it is difficult to predict which of the two formulations would be superior. However, we think that for highly flexible environments, given the reformulation of forward and backward constraints proposed here, the BECHTOLD and JACOBS model would perform better.

We have compared the AYKIN and the BECHTOLD and JACOBS models with reformulated forward and backward constraints through two sets of instances: a

real-life data set arising from an air-traffic control agency and a generated instance set in which we varied some scheduling parameters of the first set. For the first set, the two models showed a comparable performance except for one instance where the BECHTOLD and JACOBS model needed much more time than that of AYKIN. For the second set, the BECHTOLD and JACOBS model was, on average, computationally superior to the AYKIN model in terms of solution times and the number of instances solved to optimality.

Finally, we have considered two approaches of modeling work stretch duration restrictions: the enumerative approach uses conveniently placed break windows that respect the minimum and maximum pre- and post-break work stretch durations, while the implicit approach corresponds to the use of adapted forward and backward constraints. We concluded that modeling work stretch duration restrictions with the enumerative approach yields very large problems if all break window placement possibilities are considered. Otherwise, if only a smaller number of break window placement alternatives is considered, solution times are reduced but a larger workforce size is needed. Instead, modeling work stretch duration restrictions with forward and backward constraints yields an optimal workforce size in relatively small solution times.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant CRDPJ228083-99 and by AD OPT Technologies Inc. This support is gratefully acknowledged.

Bibliography

- I. ADDOU AND F. SOUMIS. “Bechtold-Jacobs generalized model for shift scheduling with extraordinary overlap.” Technical report, GERAD, HEC Montréal, 2004.
- T. AYKIN. “Optimal shift scheduling with multiple break windows.” *Management Science*, **42**:591–602 (1996).
- T. AYKIN. “A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows.” *Journal of the Operational Research Society*, **49**:603–615 (1998).
- T. AYKIN. “A comparative evaluation of modeling approaches to the labor shift scheduling problem.” *European Journal of Operational Research*, **125**:381–397 (2000).
- J. BAILEY. “Integrated days off and shift personnel scheduling.” *Computers and Industrial Engineering*, **9**:395–404 (1985).
- J. BAILEY AND J. FIELDS. “Personnel scheduling with flexshift models.” *Journal of Operations Management*, **5**:327–338 (1985).
- K.R. BAKER. “Workforce allocation in cyclic scheduling problems: A survey.” *Operational Research Quarterly*, **27**(1):155–167 (1976).
- J.F. BARD, C. BINICI AND A.H. DESILVA. “Staff scheduling at the United States Postal Service.” *Computers & Operations Research*, **30**:745–771 (2003).
- J.J. BARTHOLDI. “A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering.” *Operations Research*, **29**:501–510 (1981).

- S.E. BECHTOLD AND L.W. JACOBS. "Implicit modeling of flexible break assignments in optimal shift scheduling." *Management Science*, **36**:1339–1351 (1990).
- S.E. BECHTOLD AND L.W. JACOBS. "Labor utilization effects of labor scheduling flexibility alternatives in a tour scheduling environment." *Decision Sciences*, **24**:148–166 (1993).
- S.E. BECHTOLD AND L.W. JACOBS. "The equivalence of general set-covering and implicit integer programming formulations for shift scheduling." *Naval Research Logistics*, **43**:233–249 (1996).
- M.J. BRUSCO AND L.W. JACOBS. "Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations." *European Journal of Operational Research*, **86**:249–261 (1995).
- M.J. BRUSCO AND L.W. JACOBS. "Optimal models for meal-break and start-time flexibility in continuous tour scheduling." *Management Science*, **46**:1630–1641 (2000).
- T. ÇEZİK AND O. GÜNLÜK. "Reformulating linear programs with transportation constraints- with applications to workforce scheduling." *Naval Research Logistics*, **51**:275–296 (2004).
- G.B. DANTZIG. "A comment on Edie's traffic delays at toll booths." *Operations Research*, **2**:339–341 (1954).
- A. GABALLA AND W. PEARCE. "Telephone sales manpower planning at Quantas." *Interfaces*, **9**:1–9 (1979).
- W.B. HENDERSON AND W.L. BERRY. "Heuristic methods for telephone operator shift scheduling: an experimental analysis." *Management Science*, **22**:1372–1380 (1976).

- E.G. KEITH. "Operator scheduling." *AIEE Transactions*, **11**:37–41 (1979).
- S.L. MOONDRA. "An L.P. model for workforce scheduling in banks." *Journal of Bank Research*, **6**:299–301 (1976).
- M. REKIK, J-F. CORDEAU AND F. SOUMIS. "Using Benders decomposition to implicitly model tour scheduling." *Annals of Operations Research*, **128**:111–133 (2004).
- M. SEGAL. "The operator scheduling problem: a network flow approach." *Operations Research*, **22**:808–824 (1974).
- G.M. THOMPSON. "Improved implicit optimal modeling of the labor shift scheduling problem." *Management Science*, **41**:595–607 (1995).

APPENDIX

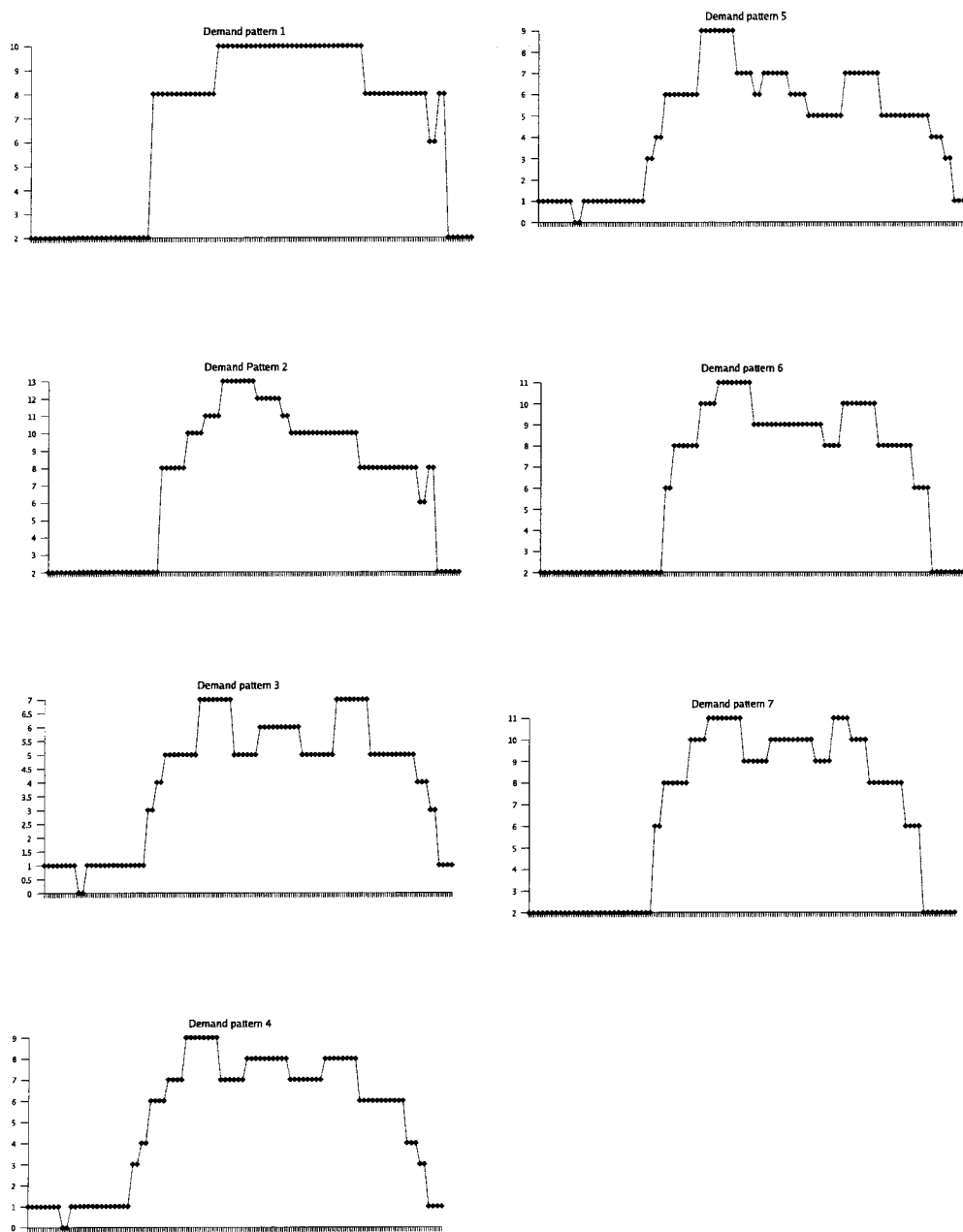


Figure 4.2 Demand patterns arising from the air-traffic control agency

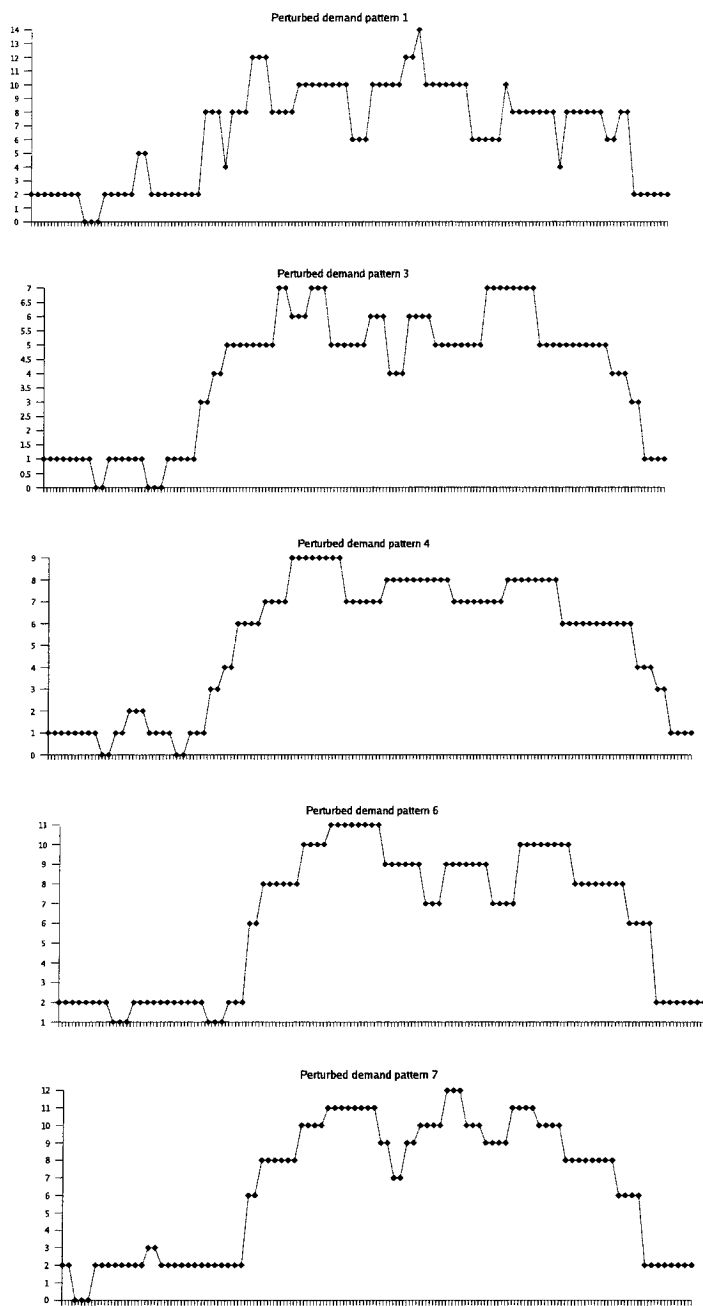


Figure 4.4 Perturbed demand patterns

CHAPITRE 5

SOLVING LARGE SHIFT SCHEDULING PROBLEMS

Monia Rekik, Jean-François Cordeau et François Soumis, soumis pour publication à *RAIRO*, décembre 2005.

L'approche de modélisation implicite a permis de réduire considérablement la taille des modèles mathématiques et d'utiliser ainsi des méthodes de résolution exactes plutôt que des heuristiques. Une telle approche a eu beaucoup de succès pour des problèmes de construction simultanée de cycles et de quarts de travail relativement simples ainsi que pour des problèmes de construction de quarts un peu plus complexes mais qui supposent un environnement d'opération discontinue. Pour les environnements continus, les méthodes exactes ont été généralement utilisées pour un horizon d'une seule journée d'opération.

Dans cet article, nous proposons deux méthodes de résolution pour un problème de construction de quarts de travail sur un horizon de planification s'étendant sur plusieurs jours. Nous commençons par définir l'environnement de travail considéré. Cet environnement incorpore en fait toutes les formes de flexibilité abordées dans le deuxième article, y compris les pauses décomposables et les restrictions sur la durée de travail consécutif minimum et maximum. Le problème de construction de quarts traité au deuxième article considère une seule journée d'opération. Dans cet article, nous considérons un environnement d'opération continue sur un horizon de planification beaucoup plus long. Nous décrivons ensuite les deux formulations implicites utilisées pour modéliser ce problème. Les modèles que nous proposons étendent ceux proposées dans le deuxième article à plusieurs jours d'opération. Ces

formulations utilisant des contraintes *forward* et *backward* adaptées, nous rappelons brièvement la forme de ces contraintes dans le cas général et les conditions de leur utilisation.

Nous proposons ensuite deux méthodes pour résoudre le problème. La première méthode est basée sur le concept de branchement local proposé récemment par FISCHETTI and LODI (2003) pour les problèmes en nombres entiers complexes. Nous rappelons d'abord les principes de base du branchement local ainsi que les quelques aspects heuristiques qui peuvent lui être imbriqués. Le branchement local repose principalement sur l'utilisation de coupes (*local branching cuts*) qui permettent d'explorer en priorité un sous-arbre de branchement a priori prometteur. Les coupes proposées et testées par FISCHETTI and LODI (2003) utilisent les variables binaires. Les modèles de construction de quarts proposés utilisant des variables entières, nous décrivons des coupes de branchement utilisant ce type de variables et qui sont spécifiques aux problèmes de construction de quarts. La forme de ces coupes s'inspire de l'extension proposée par FISCHETTI and LODI (2003). Selon les valeurs de certains paramètres, la méthode de branchement local peut être une méthode exacte ou une méthode heuristique.

La deuxième méthode de résolution proposée est purement heuristique. Elle est basée sur une décomposition temporelle du problème en plusieurs sous-problèmes, un pour chaque fenêtre de temps de l'horizon de planification. Chaque sous-problème correspond à un problème de construction de quarts restreint qui est résolu par séparation et évaluation progressive. Afin de tenir compte du caractère continu de l'environnement d'opération, les fenêtres de temps se chevauchent et sont définies dynamiquement au cours du processus de résolution. La méthode de décomposition par fenêtre de temps suppose qu'une solution explicite est fournie lors de la résolution de chaque sous-problème. Une des formulations implicites proposées ne donnant pas

des quarts explicites, nous proposons une méthode simple qui permet de convertir la solution implicite en une solution explicite.

Après avoir exposé ces deux approches de résolution dans un contexte général, nous étudions leur performance à travers des instances réelles fournies par une agence de contrôleurs aériens. Nous comparons chacune de ces approches à la méthode traditionnelle de séparation et d'évaluation progressive de CPLEX 9.0. Les résultats numériques obtenus démontrent que le branchement local que nous proposons est capable d'identifier des solutions meilleures que celles fournies par CPLEX pour la plupart des instances. La méthode de décomposition par fenêtres de temps est moins performante pour les instances relativement faciles. Cependant, pour les instances plus difficiles où aucune solution entière n'a été identifiée ni par le branchement local ni par la méthode de séparation et d'évaluation progressive de CPLEX, la méthode de décomposition donne de bonnes solutions en des temps relativement courts.

La principale contribution de cet article est de proposer et de comparer quelques méthodes de résolution heuristiques et exactes pour des problèmes de construction de quarts de travail complexes sur un horizon de plusieurs jours. Les résultats obtenus montrent les mérites du branchement local proposé par rapport au branchement standard pour le problème de construction de quarts. Ils confirment également l'utilité des méthodes de décomposition pour les problèmes très complexes.

Solving Large Shift Scheduling Problems

MONIA REKIK

École Polytechnique de Montréal

JEAN-FRANÇOIS CORDEAU

HEC Montréal

FRANÇOIS SOUMIS

École Polytechnique de Montréal

December 2005

Abstract

This paper considers large shift scheduling problems with different shift start times and lengths, fractionable breaks and work stretch duration restrictions. Two solution approaches are proposed to solve the problems over a multiple-day planning horizon. The first approach is based on a local branching strategy and the second one is based on a temporal decomposition of the problem. Local branching is very efficient in finding good feasible solutions when compared to a classical branch-and-bound procedure. However, the decomposition approach has the advantage of yielding feasible solutions in short computing times, even for difficult instances.

Keywords: shift scheduling, flexibility, fractionable breaks, work stretch restrictions, forward and backward constraints, local branching, heuristic.

5.1 Introduction

Shift scheduling problems have received a lot of attention in the last decades. Constructing shifts typically consists of specifying the daily work start and finish times as well as the number, the duration and the position of the breaks. To improve flexibility, organizations can use different shift start times and shift lengths as well as break windows, fractionable breaks, etc. A *break window* allows breaks to occur within a time window instead of at a fixed time. Unlike standard breaks, a *fractionable break* is not restricted to be taken as a whole. It can be separated into several subbreaks under some conditions.

Although scheduling flexibility adds a lot of complexity to the shift scheduling problem, it has been shown to help reduce labor costs. For example, BECHTOLD and JACOBS (1993) have evaluated the reduction in workforce size obtained by increasing the number of possible shift starts, by varying shift lengths, and by considering break windows. BRUSCO and JACOBS (2001) have demonstrated through a large experimental study that just a small subset of shift start times is enough to provide a minimal workforce size. However, the quality of the solution may rapidly deteriorate when the adequate subset is not considered. Moreover, finding the suitable subset is a difficult and time consuming task. AYKIN (1996) has studied the impact of break placement flexibility in a multiple break context. He concluded that using large break windows for all breaks may considerably decrease the number of employees required. Recently, TOPALUĞLU and OZKARAHAN (2002) have compared the workforce size needed for a scheduling environment using only eight-hour shifts to that required for an environment allowing three different shift lengths (eight-, ten- and twelve-hour shifts). An average reduction of 20% was obtained for the environment offering greater shift length variability. Furthermore, the large set of instances considered by TOPALUĞLU and OZKARAHAN (2002) confirm the positive

impact of using large break windows on the total workforce size. Finally, REKIK et al. (2005) have recently introduced the concept of fractionable breaks. They reported that the use of such “dividable” breaks may yield up to 8% savings in terms of the number of employees needed when compared to a context where only standard breaks (i.e., that cannot be divided) are used.

In this paper, we consider a continuous shift scheduling problem that includes a high degree of flexibility in terms of shift start time, shift length, break length and break placement. The problem also considers the fractionable break concept and the work stretch duration restrictions recently presented by REKIK et al. (2005). Work stretch duration restrictions impose a minimal and maximal period of work before and after each break within a shift. In their paper, REKIK et al. (2005) have focused on the modeling aspect of the problem and estimated the labor savings resulting from fractionable breaks and work stretch duration restrictions. They reported computational results for a 24-hour operating day. In this paper, we focus on solution methods for a continuous shift scheduling problem over longer planning horizons (i.e., one or several weeks). Two solution approaches are proposed. The first approach is based on the local branching method initially introduced by FISCHETTI and LODI (2003) to solve general integer programming problems. These authors intensively studied local branching cuts for integer problems containing a subset of binary variables. They also proposed some local branching extensions for problems involving only integer variables. The shift scheduling formulations considered in this paper belong to this category of pure integer programming models. Relying on the suggestions made by FISCHETTI and LODI (2003), we propose some local branching cuts that are specific to shift scheduling problems. The second solution approach, a windowing approach, is based on a temporal decomposition of the planning horizon into time windows. Each restricted shift scheduling problem associated to a time window is solved to optimality by a classical branch-and-bound procedure. Some

parts of the solution obtained for a time window are re-optimized in order to consider overlapping shifts arising in continuous operating environments. Both approaches are compared to the branch-and-bound procedure of CPLEX 9.0.

The computational experiments show that the proposed local branching yields better solutions than the branch-and-bound procedure of CPLEX 9.0 and the windowing approach for relatively small problems. For larger problems, the windowing approach is superior and yields good feasible solutions in relatively short times.

The remainder of this paper is organized as follows. In the next section we give an overview of the shift scheduling literature by focusing on solution methods. In Section 5.3, we define the operating environments and expose the implicit formulations used to model the problem as proposed by REKIK et al. (2005). Local branching and time windowing approaches are then described in Section 5.4. In Section 5.5, we illustrate model characteristics for a set of real-life instances from an air-traffic control agency as well as a set of generated instances. The computational performance of the proposed models is finally assessed and compared for each of the three solution approaches.

5.2 Literature Review

Two main approaches have been proposed to handle highly flexible and thus complex shift scheduling problems: *explicit* approaches and *implicit* approaches.

The explicit approach uses the classical set covering formulation initially introduced by DANTZIG (1954) and tries to handle the large size of the resulting models by developing efficient heuristics. In fact, a typical set covering model defines

a separate integer variable for each explicit work shift (i.e., daily work start and finish times with assigned rest periods). Clearly, a scheduling problem with a high degree of flexibility yields a large number of shift alternatives and thus large set covering models that are very difficult to solve with exact methods. For example, HENDERSON and BERRY (1976) have modeled a shift scheduling problem including up to 15,000 explicit work shifts. The problem is solved heuristically in two phases. A subset of explicit shifts is first selected with the so-called *maxdif* heuristic. The resulting shift scheduling problem with a restricted subset is then solved using some heuristic procedures based on the solution of the LP relaxation of the problem. The *maxdif* procedure adds shifts iteratively to the subset of selected shifts, one shift at a time. At each iteration, all remaining (i.e., not yet selected) shifts are compared to all shifts already selected. The shift that maximizes the number of periods that are not covered by the selected shifts is chosen. Three heuristics are proposed for the second phase to solve the restricted shift scheduling problems. Each of these heuristics converts the fractional LP solution into an integer one and then tries to improve it by local search. The local search consists in permuting employee shifts. In another context, THOMPSON (1996a) has considered the explicit approach for a shift scheduling problem where employees are not continuously available. A simulated annealing procedure is used to solve the problem.

The *implicit* approach focuses more on the modeling aspect and tries to reduce the problem size by implicitly representing some forms of flexibility. Exact branch-and-bound or branch-and-cut procedures are generally used to solve these implicit models. MOONDRA (1976) was the first to implicitly represent shift length. His model considers shift start time and shift finish time variables, and uses a set of constraints to impose limits on the shift duration. BECHTOLD and JACOBS (1990) implicitly model break placement flexibility for a discontinuous shift scheduling problem in a single break context. They define separate shift and break variables and

ensure a correct match between them by using the so-called *forward* and *backward* constraints. This set of constraints guarantees that an eligible break can be assigned to each shift in its associated break window, without specifying its actual start time. Complete shift schedules with appropriate break assignments are constructed *a posteriori* using a break allocation algorithm. The allocation procedure arranges shifts in a nondecreasing order with respect to the latest period in which a break may occur and assigns shifts to the earliest available breaks. In fact, using forward and backward constraints assumes the absence of *extraordinary overlap*. BECHTOLD and JACOBS (1990) define extraordinary overlap (E.O.) as the situation where there exist two shifts such that the break window for one shift begins strictly earlier and ends strictly later than the break window for the other shift. A branch-and-bound procedure is then used to solve the problem. BECHTOLD and JACOBS (1990) show, through computational experiments, that their model is superior to the traditional set covering formulation with respect to computing time and memory. Recently, ADDOU and SOUMIS (2004) have proved that extraordinary overlap may be handled by an extension of the formulation of BECHTOLD and JACOBS (1990) which incorporates a small set of additional constraints. One constraint is in fact added for each observed E.O.

THOMPSON (1995) has later combined the work of MOONDRA (1976) to implicitly represent shifts and the work of BECHTOLD and JACOBS (1990) to implicitly represent break placement. Each shift is assumed to receive at most one break with some restrictions on pre- and post-break work stretch durations. A branch-and-bound procedure followed by a post-processor were used to generate actual schedules. The post-processor constructs explicit shifts and assigns breaks to them by using a first-in-first-out (FIFO) procedure. THOMPSON (1995) has reported that the FIFO matching can alleviate the E.O. problem.

AYKIN (1996) has proposed another implicit approach to model break placement flexibility for a shift scheduling problem with multiple breaks and multiple break windows. He defines a break variable for each shift and each possible start time within the associated break window. Contrarily to BECHTOLD and JACOBS (1990), AYKIN (1996) does not impose any restriction on the definition of break windows. Moreover, AYKIN (1996) has considered a continuous operating environment where shifts are permitted to overlap from one day to the next. AYKIN (1996) used the demand constraints to compute upper bounds for shift variables. These upper bounds were included in the proposed formulation and were helpful in reducing the solution time in the branch-and-bound process. In a subsequent paper, AYKIN (1998) has proposed a more elaborate branch-and-cut algorithm that uses objective value cuts and dynamically computed upper bounds on shift variables. The linear relaxation of the implicit formulation proposed in AYKIN (1996) is first solved to obtain a lower bound. A rounding heuristic is then applied to the fractional solution to get a feasible schedule and thus an upper bound. Cuts are added and iteratively updated while using the rounding heuristic and a limited branch-and-bound search to find an optimal schedule. Good results were obtained for large instances. An optimal solution was found in several cases; in the remaining ones, the “best non-optimal” solution (i.e., superior by one to the lower bound) was identified.

Recently, REKIK et al. (2005) have introduced a new break flexibility alternative for shift scheduling environments: *fractionable breaks*. They define a fractionable break as a break that can be divided into several subbreaks under some conditions. The scheduling problem also considers flexibility in terms of shift start time and length as well as break placement. To ensure that periods of work and rest are well distributed, REKIK et al. (2005) impose a minimal and a maximal period of work before and after each break within a shift. Two implicit models were proposed. The first model extends the approach of AYKIN (1996) and the second one extends that

of BECHTOLD and JACOBS (1990). Adapted forward and backward constraints are used in both formulations to model the work stretch duration restrictions. REKIK et al. (2005) have proposed a reformulation of general forward and backward constraints and have shown that it considerably reduces model density and solution times.

5.3 Mathematical Models

This section extends to a continuous w -day planning horizon the formulations proposed by REKIK et al. (2005) for the shift scheduling problem with fractionable breaks, standard breaks, break windows and work stretch duration restrictions. REKIK et al. (2005) have presented two implicit formulations. The first formulation, denoted hereafter by ($P1$), is inspired by the work of AYKIN (1996) and the second one, denoted by ($P2$), is based on the work of BECHTOLD and JACOBS (1990). ÇEZİK and GÜNLÜK (2004) and REKIK et al. (2004) have generalized the concept of forward and backward constraints initially introduced by BECHTOLD and JACOBS (1990) to model break placement flexibility. They proved that the feasibility of the resulting transportation problems can be ensured by the use of adequate forward and backward constraints. In this context, REKIK et al. (2005) proved that satisfying work stretch duration restrictions also corresponds to ensuring the feasibility of some transportation problem, and that these transportation problems possess the substructure required to apply the generalized forward and backward constraints.

We first introduce the parameters and the sets needed to describe generalized forward and backward constraints.

5.3.1 Generalized forward and backward constraints

Consider a transportation problem, $T(N_1, N_2)$, represented by a bipartite network $G = (N_1 \cup N_2, A)$ where N_1 and N_2 are the sets of supply and demand nodes, respectively. The supply of each node $i \in N_1$ is denoted by O_i and the demand of each node $j \in N_2$ is denoted by D_j . Assume that the structure of $T(N_1, N_2)$ is such that a total order relation \prec can be defined on set N_2 . Assume also that each supply node $i \in N_1$ is connected to a set of consecutive demand nodes $j \in N_2$, denoted by P_i . Finally, there exist no supply nodes i_1 and i_2 in N_1 such that $\min(P_{i_1}) \prec \min(P_{i_2})$ and $\max(P_{i_2}) \prec \max(P_{i_1})$.

To describe forward and backward constraints, we define the following sets:

$$\begin{aligned}
 N_2^s &= \cup_{i \in N_1} \{\min(P_i)\} \\
 N_2^e &= \cup_{i \in N_1} \{\max(P_i)\} \\
 N_2^B(j) &= \{j' \in N_2 | j \preceq j'\} \quad \forall j \in N_2^s \\
 N_2^F(j) &= \{j' \in N_2 | j' \preceq j\} \quad \forall j \in N_2^e \\
 N_1^B(j) &= \{i \in N_1 | P_i \subseteq N_2^B(j)\} \quad \forall j \in N_2^s \\
 N_1^F(j) &= \{i \in N_1 | P_i \subseteq N_2^F(j)\} \quad \forall j \in N_2^e.
 \end{aligned}$$

Finally, let $n_2^s = \min(N_2)$ and $n_2^e = \max(N_2)$. Forward and backward constraints are then given, respectively, by:

$$\sum_{j' \in N_2^F(j)} D_{j'} \geq \sum_{i \in N_1^F(j)} O_i \quad \forall j \in N_2^e \setminus \{n_2^e\}$$

and

$$\sum_{j' \in N_2^B(j)} D_{j'} \geq \sum_{i \in N_1^B(j)} O_i \quad \forall j \in N_2^s \setminus \{n_2^s\}.$$

Under the assumptions made at the beginning of this section, the feasibility of $T(N_1, N_2)$ is ensured by the use of forward and backward constraints and an equality constraint between the total supply and total demand (see ÇEZİK and GÜNLÜK, 2004 or REKİK et al., 2004):

$$\sum_{i \in N_1} O_i - \sum_{j \in N_2} D_j = 0.$$

REKİK et al. (2005) proposed a reformulation of forward and backward constraints that uses additional slack variables without increasing the total number of constraints. They proved that such a reformulation considerably reduces the density of the constraint matrix and solution times. To simplify the presentation, we illustrate the models with a classical formulation of forward and backward constraints. However, in the computational experiments, we use the models with reformulated forward and backward constraints.

5.3.2 Problem definition

We consider a continuous planning horizon of w days and 24-hour work days. As is common in shift scheduling models, the planning horizon is divided into *periods* of equal lengths. The set of all periods is denoted by I . The number of employees required in each period $i \in I$ is assumed to be known in advance and is denoted by d_i .

REKIK et al. (2005) circumvented the complexity resulting from the fractionable break concept by using an enumerative pre-processor that generates the so-called *break profiles*. A break profile represents a sequence of break lengths that must be attributed to a given shift. Thus, if a shift must receive a fractionable break, all feasible sequences of subbreak lengths associated with a fractionable break are enumerated. If the shift must also receive some standard breaks (i.e., breaks that cannot be divided), all admissible permutations of subbreak and standard breaks are enumerated. The pre-processor also generates the so-called *window profile* for each shift. A window profile includes the break windows associated with the breaks of the break profile of the given shift. These break windows are determined by considering the pre- and post-break work stretch duration restrictions and the pre-existing break windows of standard breaks. The models proposed by REKIK et al. (2005) handle the case where work stretch duration parameters (i.e., minimum and maximum pre- and post-break work stretch durations) depend on the time interval (the so-called *day-part*) of the day. The minimum and maximum number of consecutive work periods associated with a day-part dp are denoted respectively by α_{dp} and μ_{dp} . The set of all day-parts is denoted by TDP (generally, there are three day-parts: day, evening and night). The work stretch parameters that apply to a given shift are those associated with the day-part covering the most periods in the shift.

A set J of all possible shifts is then generated by the pre-processor. A shift $j \in J$ is defined for each combination of start time, length, break profile and window profile. The break profile, respectively the window profile, associated with shift j is denoted by BP_j , respectively WP_j . The total number of breaks that must be attributed to a shift j is denoted by q_j (q_j is thus the number of elements in BP_j or WP_j). Parameter $BP_j(p)$ is used to denote the length of the break that must be attributed to shift j in the p^{th} position whereas $WP_j(p)$ denotes the break window associated with it. The set of all break profiles, i.e., $\cup_{j \in J} BP_j$, is denoted by TBP . Finally,

the day-part that maximizes the number of periods covered by shift j is denoted by dp_j .

To determine the set K of all breaks, the break profiles and the window profiles of all shifts $j \in J$ are considered. A break k is defined for each break profile $bp \in TBP$, each position p of a break in bp , each possible start time t (with respect to the window profiles associated with break profile bp), and each admissible day-part dp of the operating day. A day-part dp is admissible for break k if break k can be matched with a shift j associated with dp (i.e., such that $dp_j = dp$).

Extending the models proposed by REKIK et al. (2005) from a one-work-day planning horizon to a w -work-day horizon consists in enlarging the set of periods I , the set of shifts J , and the set of breaks K (K is actually derived from J).

5.3.3 Implicit formulation (P1)

REKIK et al. (2005) defined a network flow variable X_{jk} for each shift j and each break k that can be matched with shift j . This variable represents the number of breaks k attributed to shift j . A break k is eligible for shift j if its break profile corresponds to the same break profile BP_j of shift j , if it has a position p_k and a start time t_k such that $t_k \in WP_j(p_k)$, and, finally, if it is associated with the same day-part as shift j . In the following, the set of breaks k that can be matched with shift j in the position p , $p \in Q_j$, $Q_j = \{1, \dots, q_j\}$, is denoted by $K_{j(p)}$. Similarly, J_k will denote the set of shifts j that can receive a break k in position p_k .

The minimum and maximum number of periods that can be worked consecutively within shift $j \in J$ are denoted respectively by α_j and μ_j . The work stretch duration restrictions imply that two consecutive breaks attributed to shift j must be separated

by at least α_j and at most μ_j periods. REKIK et al. (2005) have shown that these constraints can be modeled with adapted forward and backward constraints derived from a particular transportation problem $T(K_{j(p)}, K_{j(p+1)})$ (as defined in Section 5.3.1) where $p \in Q_j^-$, $Q_j^- = \{1, \dots, q_j - 1\}$ and X_{jk} represents the supply/demand associated with nodes in $K_{j(p)}$ and $K_{j(p+1)}$. An arc exists between a node $k^1 \in K_{j(p)}$ and a node $k^2 \in K_{j(p+1)}$ if break k^2 starts at least α_j periods and at most μ_j periods after break k^1 ends.

To describe the model, the following parameters are introduced:

$\delta_{ij} = 1$ if shift $j \in J$ covers period $i \in I$; 0 otherwise;

$\rho_{ik} = 1$ if break k covers period $i \in I$; 0 otherwise.

Model (P1) is given by :

$$(P1) \quad \text{minimize} \quad \sum_{j \in J} S_j \quad (5.1)$$

subject to

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} \sum_{j \in J_k} X_{jk} \geq d_i \quad \forall i \in I \quad (5.2)$$

$$\sum_{k \in K_{j(p)}} X_{jk} - S_j = 0 \quad \forall j \in J, p \in Q_j \quad (5.3)$$

$$\sum_{k' \in K_{j(p+1)}^F(k)} X_{jk'} - \sum_{k' \in K_{j(p)}^F(k)} X_{jk'} \geq 0 \quad \forall j \in J, p \in Q_j^-, k \in K_{j(p+1)}^e \setminus \{k_{j(p+1)}^e\} \quad (5.4)$$

$$\sum_{k' \in K_{j(p+1)}^B(k)} X_{jk'} - \sum_{k' \in K_{j(p)}^B(k)} X_{jk'} \geq 0 \quad \forall j \in J, p \in Q_j^-, k \in K_{j(p+1)}^s \setminus \{k_{j(p+1)}^s\} \quad (5.5)$$

$$\sum_{k \in K_{j(p+1)}} X_{jk} - \sum_{k \in K_{j(p)}} X_{jk} = 0 \quad \forall j \in J, p \in Q_j^- \quad (5.6)$$

$$S_j \geq 0 \quad \text{and integer} \quad \forall j \in J \quad (5.7)$$

$$X_{jk} \geq 0 \quad \text{and integer} \quad \forall j \in J, k \in \cup_{p \in Q_j} K_{j(p)}. \quad (5.8)$$

The objective function (5.1) minimizes the total number of employees over the planning horizon. Demand constraints (5.2) ensure that the number of employees working in a certain period, i.e., those who are present and not on break, is at least equal to the demand for that period. Break window constraints (5.3) ensure that each shift j receives q_j breaks, one in each position $p \in Q_j$, such that the break in position p has a length compatible with the break profile of shift j and starts within $WP_j(p)$. Work stretch duration constraints (5.4)-(5.6) are forward, backward and equality constraints corresponding to transportation problems $T(K_{j(p)}, K_{j(p+1)})$, $j \in J, p = 1, \dots, q_j - 1$. It can be verified that equality constraints (5.6) are redundant in the presence of constraints (5.3) and can thus be removed.

5.3.4 Implicit formulation (P2)

Model (P2) uses shift variables S_j as defined for model (P1). A break variable B_k is defined for each break $k \in K$ and represents the number of employees receiving break k .

REKIK et al. (2005) have proved that each shift will receive its breaks within the associated break windows if the feasibility of some transportation problems is ensured. In fact, they defined a transportation problems $T(J_{(bp,dp)}, K_{(bp,dp,p)})$ for each break profile $bp \in TBP$, each day-part $dp \in TDP$, and each possible position p of a break in bp , $p \in N_{bp}$, where $N_{bp} = \{1, \dots, n_{bp}\}$ and n_{bp} denotes the total number of breaks in bp . The set of supply nodes $J_{(bp,dp)}$ contains all shifts j associated with break profile bp and day-part dp , i.e., $J_{(bp,dp)} = \{j \in J | BP_j = bp \text{ and } dp_j = dp\}$. The supply associated with node j is S_j . The set of demand nodes $K_{(bp,dp,p)}$ contains all breaks k associated with day-part dp and occupying the p^{th} position in break profile

bp , i.e., $K_{(bp,dp,p)} = \{k \in K \mid BP_k = bp \text{ and } dp_k = dp \text{ and } p_k = p\}$. The demand for node k is B_k . An arc (j, k) exists between a shift $j \in J_{(bp,dp)}$ and a break $k \in K_{(bp,dp,p)}$ if break k starts within the break window associated with the p^{th} position of the window profile WP_j of shift j . Breaks in $K_{(bp,dp,p)}$ are sorted in ascending order with respect to their start time, which defines a total order relation on set $K_{(bp,dp,p)}$. The authors assume that there is no extraordinary overlap between breaks in $K_{(bp,dp,p)}$. Hence, for a given break profile bp , a given day-part dp and a given position p of a break in bp , the transportation problem $T(J_{(bp,dp)}, K_{(bp,dp,p)})$ can be replaced by a set of forward, backward and equality constraints as described in Section 5.3.1.

To model pre- and post-break work stretch duration restrictions, REKIK et al. (2005) consider a transportation problem $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$ for each break profile $bp \in TBP$, each day-part $dp \in TDP$ and each position p of a break in bp , $p \in N_{bp}^-$ where $N_{bp}^- = \{1, \dots, n_{bp} - 1\}$. The supply/demand of a node k is B_k . An arc (k^1, k^2) exists between a break $k^1 \in K_{(bp,dp,p)}$ and a break $k^2 \in K_{(bp,dp,p+1)}$ if break k^2 starts at least α_{dp} and at most μ_{dp} periods after break k^1 ends. Breaks in $K_{(bp,dp,p+1)}$ are sorted in ascending order with respect to their start time. It can be verified that the condition of no extraordinary overlap, as defined in Section 5.3.1, is satisfied. Thus, problems $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$, $bp \in TBP, dp \in TDP, p \in N_{bp}^-$, can be replaced by adapted forward, backward and equality constraints.

Model (P2) is thus written as follows:

$$(P2) \quad \text{minimize } \sum_{j \in J} S_j \quad (5.9)$$

subject to

$$\sum_{j \in J} \delta_{ij} S_j - \sum_{k \in K} \rho_{ik} B_k \geq d_i \quad \forall i \in I \quad (5.10)$$

$$\begin{aligned} \sum_{k' \in K_{(bp, dp, p)}^F(k)} B_{k'} - \sum_{j \in J_{(bp, dp)}^F(k)} S_j &\geq 0 & \forall bp \in TBP, dp \in TDP, p \in N_{bp} \\ & & \forall k \in K_{(bp, dp, p)}^e \setminus \{k_{(bp, dp, p)}^e\} \end{aligned} \quad (5.11)$$

$$\begin{aligned} \sum_{k' \in K_{(bp, dp, p)}^B(k)} B_{k'} - \sum_{j \in J_{(bp, dp)}^B(k)} S_j &\geq 0 & \forall bp \in TBP, dp \in TDP, p \in N_{bp} \\ & & \forall k \in K_{(bp, dp, p)}^s \setminus \{k_{(bp, dp, p)}^s\} \end{aligned} \quad (5.12)$$

$$\sum_{k \in K_{(bp, dp, p)}} B_k - \sum_{j \in J_{(bp, dp)}} S_j = 0 \quad \forall dp \in TDP, bp \in TBP, p \in N_{bp} \quad (5.13)$$

$$\begin{aligned} \sum_{k' \in K_{(bp, dp, p+1)}^F(k)} B_{k'} - \sum_{k' \in K_{(bp, dp, p)}^F(k)} B_{k'} &\geq 0 & \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \\ & & \forall k \in K_{(bp, dp, p+1)}^e \setminus \{k_{(bp, dp, p+1)}^e\} \end{aligned} \quad (5.14)$$

$$\begin{aligned} \sum_{k' \in K_{(bp, dp, p+1)}^B(k)} B_{k'} - \sum_{k' \in K_{(bp, dp, p)}^B(k)} B_{k'} &\geq 0 & \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \\ & & \forall k \in K_{(bp, dp, p+1)}^s \setminus \{k_{(bp, dp, p+1)}^s\} \end{aligned} \quad (5.15)$$

$$\sum_{k \in K_{(bp, dp, p+1)}} B_k - \sum_{k \in K_{(bp, dp, p)}} B_k = 0 \quad \forall bp \in TBP, dp \in TDP, p \in N_{bp}^- \quad (5.16)$$

$$S_j \geq 0 \quad \text{and integer} \quad \forall j \in J \quad (5.17)$$

$$B_k \geq 0 \quad \text{and integer} \quad \forall k \in K. \quad (5.18)$$

Model (P2) considers the same objective function (5.9) and the same demand constraints (5.10) as model (P1). Constraints (5.11)-(5.12) are forward, backward and

equality constraints corresponding to transportation problems $T(J_{(bp,dp)}, K_{(bp,dp,p)})$, $bp \in TBP, dp \in TDP, p = 1, \dots, n_{bp}$. They ensure that each shift receives the associated number of breaks and each attributed break is eligible with respect to break length (i.e., its position in the break profile) and break start time (with respect to the associated break window). Pre- and post-break work stretch durations are modeled by forward, backward and equality constraints (5.14)-(5.16) corresponding to transportation problems $T(K_{(bp,dp,p)}, K_{(bp,dp,p+1)})$, $bp \in TBP, dp \in TDP, p = 1, \dots, n_{bp} - 1$. One can verify that equality constraints (5.16) are redundant in the presence of equality constraints (5.13) and can thus be removed.

5.4 Solution Approaches

This section presents two approaches for solving complex shift scheduling problems over long planning horizons. The first approach is based on the local branching strategy initially introduced by FISCHETTI and LODI (2003) to solve difficult MIP problems. The local branching strategy presented here uses local branching cuts that are adapted for shift scheduling problems. The second approach uses a temporal decomposition of the problem into different sub-problems, one for each time window of the planning horizon. This latter approach is referred to as *time windowing*.

5.4.1 Adapted local branching strategy

The local branching strategy aims to improve the heuristic behavior of Mixed Integer Programming solvers. It provides good feasible solutions at early stages of the solution process and updates incumbent solutions frequently, thus accelerating the solution process.

Local branching strategies use two branching criteria. The first branching criterion is external and tends to direct the search into a promising search sub-space. The second branching criterion corresponds to the branching performed within the MIP solver. It is used to explore the sub-space resulting from the first branching. As noticed by FISCHETTI and LODI (2003), local branching is inspired by local neighborhood search methods (see, e.g., MLADENOVIC and HANSEN, 1997) in the sense that the search sub-space represents in fact a neighborhood of a given feasible solution to the problem. This neighborhood is defined through the so-called *local branching constraints*.

More specifically, consider a MIP problem and let \mathcal{B} denote the set of binary variable indices. For a given solution \bar{x} of the problem, let $\bar{\mathcal{B}}$ denote the binary support of \bar{x} , i.e., $\bar{\mathcal{B}} = \{j \in \mathcal{B} : \bar{x}_j = 1\}$. For a given positive integer parameter k , FISCHETTI and LODI (2003) define the k -OPT neighborhood $\mathcal{N}(\bar{x}, k)$ of \bar{x} as the set of feasible solutions of the problem satisfying the following local branching constraints:

$$\Delta(x, \bar{x}) \leq k,$$

$$\text{where } \Delta(x, \bar{x}) = \sum_{j \in \bar{\mathcal{B}}} (1 - x_j) + \sum_{j \in \mathcal{B} \setminus \bar{\mathcal{B}}} x_j.$$

As observed by FISCHETTI and LODI (2003), for a given neighbor x of \bar{x} , the *distance function* $\Delta(x, \bar{x})$ represents the number of binary variables x_j flipping their value, with respect to \bar{x} , either from 1 to 0 or from 0 to 1.

A basic local branching scheme is illustrated in Figure 5.1 (we use a figure similar to that presented by FISCHETTI and LODI (2003)). In this figure, a triangle marked

by the word “solve” corresponds to the branching subtrees to be explored with the standard branching criterion of the exact MIP solver at hand. As one can see from Figure 5.1, the local branching method starts with an initial incumbent solution \bar{x}^1 at the root node. This solution can be obtained heuristically or by the MIP solver. Two branches are then created to partition the initial feasible region, \mathcal{N} , into two disjoint sub-regions. The left branch corresponds to the local branching constraints and defines a neighborhood of \bar{x}^1 , $\mathcal{N}(\bar{x}^1, k)$. The right branch defines all feasible solutions except those belonging to $\mathcal{N}(\bar{x}^1, k)$. The left branch node is explored with the MIP solver until an optimal solution, \bar{x}^2 , is obtained (with respect to the neighborhood). The same scheme is then re-applied on the smallest feasible region $\mathcal{N} \setminus \mathcal{N}(\bar{x}^1, k)$ with the new incumbent solution \bar{x}^2 . Obviously, updating the incumbent solution assumes that the optimal solution found in a given neighborhood improves the current incumbent one. In the case where no improved solution is obtained (such as for node 6), the right branch node (corresponding in the figure to sub-tree $\mathcal{N} \setminus (\mathcal{N}(\bar{x}^1, k) \cup \mathcal{N}(\bar{x}^2, k) \cup \mathcal{N}(\bar{x}^3, k))$) is totally explored with the MIP solver.

FISCHETTI and LODI (2003) reported that the local branching method they propose may be used either as an exact solution method or as a heuristic method that yields good feasible solutions that are not necessarily optimal. With the basic scheme described above, local branching explores all the branching tree while giving the priority to some promising search sub-spaces. It acts then as an exact solution method. For the heuristic aspect, FISCHETTI and LODI (2003) impose a time limit for the left branch computation to handle the cases where exploring all the left subtree is very time consuming. If the time limit is reached with no improved solution (with respect to the incumbent), a backtrack to the father node is done and a new left branch node is created by reducing the size of the neighborhood (i.e., reducing the right hand side of the local branching cut). If an improved solution is

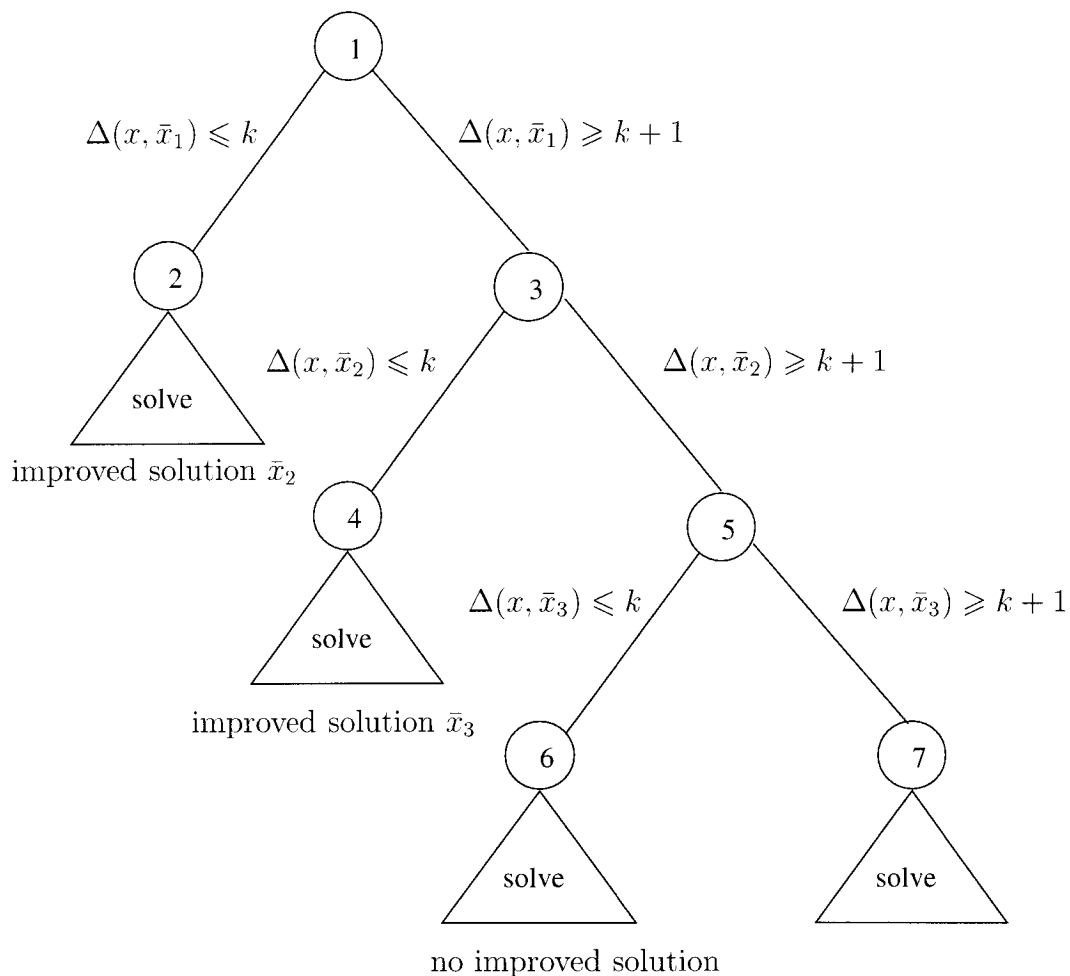


Figure 5.1 Basic local branching strategy

obtained, it becomes the new incumbent, a backtrack to the father node is done, and a new left branch node associated with the new incumbent solution is created.

Furthermore, FISCHETTI and LODI (2003) used some diversification mechanisms to handle the case where the exploration of the left branch node terminates with no improved solution. Two types of diversifications are applied: the so-called *soft* and *strong* diversifications. A soft diversification consists in enlarging the current neighborhood. A strong diversification not only considers larger neighborhoods but also accepts solutions that are worse than the incumbent. This worse solution is

then used as the new reference solution and the method is re-applied in an attempt to improve it and eventually improve the incumbent.

It is worth mentioning that local branching constraints were initially defined with respect to binary variables only. This was done intentionally since, according to FISCHETTI and LODI (2003), the difficulty of MIP problems comes essentially from these variables. The authors however suggest, as an extension of their paper, some local branching constraints including general variables that can be used for MIPs that incorporate only general integer variables.

The shift scheduling problem discussed in this paper is in fact a MIP with only general integer variables. Inspired by the suggestions of FISCHETTI and LODI (2003), we propose hereafter some local branching constraints that are specific to shift scheduling problems.

Let (\bar{S}, \bar{X}) (respectively (\bar{S}, \bar{B})), denote a feasible *reference* solution of a shift scheduling problem modeled with model (P1) (respectively model (P2)). The distance function of the local branching cut we propose here is defined with respect only to the shift variables of the reference solution. Moreover, only non zero variables are considered. Let $\bar{J}(\bar{S})$ denotes the set of these variable indices. Thus $\bar{J}(\bar{S}) = \{j \in J : \bar{S}_j \neq 0\}$.

To describe the local branching constraints, one has to introduce two additional integer variables, S_j^- and S_j^+ for each $j \in \bar{J}(\bar{S})$. The value of these variables measures the distance between the value of a current shift variable S_j and the correspond reference value \bar{S}_j . Thus:

$$S_j = \bar{S}_j + S_j^+ - S_j^- \quad \forall j \in \bar{J}(\bar{S}).$$

Thus, for a given reference solution (\bar{S}, \bar{X}) of (P1) or (\bar{S}, \bar{B}) of (P2), the local branching cut is defined as:

$$\sum_{j \in \bar{J}(\bar{S})} (S_j^+ + S_j^-) \leq k,$$

where k is a fixed integer parameter.

Except for the definition of the local branching constraints, the local branching strategy scheme used is the same as that presented by FISCHETTI and LODI (2003) with the enhanced heuristic solution scheme described above.

5.4.2 Time windowing approach

As its name suggests, the time windowing approach is based on a temporal decomposition of the shift scheduling problem into several sub-problems, one for each time window of the planning horizon. Each shift scheduling sub-problem is then solved to optimality over the associated time window. The final solution for the overall planning horizon is obtained by combining all the partial solutions of the sub-problems. Thus, time windowing appears as a pure heuristic approach since there is no guarantee that the final solution is optimal.

The definition of the time windows has a major impact on the performance of the time windowing approach in terms of solution quality and computing times. In fact, time windows must be sufficiently small to yield shift scheduling subproblems that are easy to solve within reasonable times, but also still large enough to yield a global solution that is not too far from the optimal one. Furthermore, one can decompose the planning horizon into disjoint time windows so that each period

belongs to only one time window. However such a decomposition does not consider overlapping shifts that arise in continuous scheduling problems. In fact, it assumes a discontinuous formulation of the problem, i.e., a formulation where each shift is entirely contained in a single time window. As mentioned by several researchers (see, e.g., BRUSCO and JACOBS, 1995), this assumption may result in considerable labor losses when compared to a continuous formulation of the problem.

The windowing procedure proposed in this paper considers overlapping time windows thus addressing a continuous formulation of the shift scheduling problem. A shift scheduling subproblem associated with a given time window thus incorporates all shifts that start within the time window and the corresponding breaks. The demand constraints considered are those associated with the periods belonging to the corresponding time window.

Let T denote the set of time windows resulting from a temporal decomposition of the planning horizon. One can easily define a total order relation \prec on set T . In this order a time window t_l is smaller than a time window t_m if t_l starts before t_m . One can easily index the elements of T with respect to this total order relation so that $T = \{t_1, t_2, \dots, t_n\}$ and $t_1 \prec t_2 \prec \dots \prec t_n$, where n denotes the number of generated time windows. It is worth mentioning that $t_1 \cup t_2 \dots \cup t_n = I$, the set of all periods of the planning horizon. In order to consider overlapping shifts present in continuous scheduling problems, time window t_{l+1} overlaps with its predecessor t_l (if this predecessor exists, i.e., for $l \geq 2$) on all the periods that are covered by all shifts that overlap from t_l to t_{l+1} . All these overlapping shifts are then re-considered in the shift scheduling sub-problem associated with time window t_{l+1} . Obviously, the demand constraints associated with periods in $t_l \cap t_{l+1}$ were satisfied when solving the shift scheduling problem over time window t_l . Since these periods are reconsidered in the shift scheduling sub-problem associated with t_{l+1} , one must update the right

hand side of the corresponding demand constraints. Indeed, this update consists in subtracting from the original value of each of these demands the sum of all shifts that start and finish within t_l and for which the demand period is a work period (not a break). For cyclic problems, the last time window t_n not only overlaps with its predecessor but also with the first time window t_1 . The periods covered in common by time windows t_n and t_1 are those covered by all shifts that overlap from t_n to t_1 . Thus, for the shift scheduling subproblem associated with time window t_n , the right hand sides of demand constraints associated with periods in $t_n \cap t_1$ also need to be updated. This update consists in subtracting from the original value of each of these demands the sum of all shifts that begin and end within t_1 and for which the demand period is a work period.

The final solution is obtained by combining all shift and break variables values yielded by the solution of all sub-problems. In fact, each shift and each break variable associated with shifts overlapping from one time window, t_l , to the next (t_{l+1}) will have two values. The value considered in the final solution is that obtained for time window t_{l+1} . In a cyclic case, the values considered for the variables associated with shifts that overlap from t_n to t_1 are those obtained for t_n .

The time windowing method presented above is general enough to be applied to any shift scheduling problem. The shift scheduling subproblems involve the same types of variables and constraints as the original one, but on a smaller planning horizon. The only pre-processing needed for a given subproblem consists of updating the right-hand side of some demand constraints when overlapping time windows are considered. Such a pre-processing assumes that explicit shifts are obtained after the solution of the previous sub-problem. Hence, the difficulty of the pre-processing task depends on the formulation used to model the scheduling problem. In the case

where only implicit solutions are yielded, one needs to define some procedures to obtain explicit solutions.

The shift scheduling problem addressed in this paper considers a multiple-break context and restrictions on the minimum and maximum work stretch duration before and after each break. The solution obtained by formulation $(P1)$ yields explicit shifts with specified start and finish times and break assignments. Recall that formulation $(P1)$ uses shift variables S_j for each shift $j \in J$ and break variables X_{jk} for each shift j and each break $k \in K$ that can be assigned to shift j . Thus the update of demand constraints is straightforward with formulation $(P1)$. However, formulation $(P2)$ does not use such an explicit assignment variable to link shifts with associated breaks. In fact, it incorporates separate shift and break variables $S_j, j \in J$ and $B_k, k \in K$. The correct match between shifts and breaks is ensured by the forward and backward constraints. The optimal implicit solution, (S^*, B^*) determined by $(P2)$ guarantees that an explicit schedule with break assignments can always be obtained from it. In fact, the optimal non-null breaks $B_k^*, k \in K^*$, determined by $(P2)$ can be assigned to the optimal non-null shifts $S_j^*, j \in J^*$, in different ways depending on the scheduling problem. For a single break context, this can be done using a simple allocation procedure as that described in BECHTOLD and JACOBS (1990), for example. In a multiple break context, although such a procedure assigns a break to a shift within the associated break window, it does not guarantee that these breaks are conveniently placed one with respect to another when work stretch duration restrictions must be satisfied. To handle the complexity of work stretch duration constraints, the assignment of optimal breaks to optimal shifts is done by solving an integer programming model denoted hereafter by $(P3)$. Model $(P3)$ incorporates integer variables X_{jk} , similar to those of model $(P1)$, for each shift j of a non-null optimal variable $S_j^*, j \in J^*$, and each break k of a non-null optimal variable $B_k^*, k \in K^*$, that is admissible for j . X_{jk} represents the number of breaks

$k \in K^*$ attributed to shift $j \in J^*$. The objective function of model (P3) has a constant value since the only purpose of considering the model is to determine a feasible solution to the break assignment problem. Model (P3) incorporates the same set of constraints (4.7)-(4.12) of model (P1) but only for shifts $j \in J^*$ and breaks in $k \in K^*$; they will also consider only the break and shift variables of these sets. Furthermore, shift variables S_j of constraints (4.7) will have a constant value equal to S_j^* . Finally, the following set of constraints is added to (P1) to ensure that the number of assignments of a break k does not exceed the total number of available breaks k :

$$\sum_{j \in J_k \cup J^*} X_{jk} = B_k^* \quad \forall k \in K^*.$$

Model (P3) is thus given by:

$$(P3) \quad \text{minimize} \quad 0$$

subject to

$$\begin{aligned} \sum_{k \in K_{j(p)} \cap K^*} X_{jk} &= S_j^* & \forall j \in J^*, p \in Q_j \\ \sum_{j \in J_k \cap J^*} X_{jk} &= B_k^* & \forall k \in K^* \\ \sum_{k' \in K_{j(p+1)}^f(k) \cap K^*} X_{jk'} - \sum_{k' \in K_{j(p)}^f(k) \cap K^*} X_{jk'} &\geq 0 & \forall j \in J^*, p \in Q_j^-, k \in (K_{j(p+1)}^e \setminus \{k_{j(p+1)}^e\}) \cap K^* \\ \sum_{k' \in K_{j(p+1)}^b(k) \cap K^*} X_{jk'} - \sum_{k' \in K_{j(p)}^b(k) \cap K^*} X_{jk'} &\geq 0 & \forall j \in J^*, p \in Q_j^-, k \in (K_{j(p+1)}^s \setminus \{k_{j(p+1)}^s\}) \cap K^* \\ \sum_{k \in K_{j(p+1)} \cap K^*} X_{jk} - \sum_{k \in K_{j(p)} \cap K^*} X_{jk} &= 0 & \forall j \in J, p \in Q_j^- \\ X_{jk} &\geq 0 & \text{and integer} \quad \forall j \in J^*, k \in (\cup_{p \in Q_j} K_{j(p)}) \cap K^*. \end{aligned}$$

5.5 Computational Experiments

The purpose of this section is to first compare different solution approaches for real-life shift scheduling problems arising from an air-traffic control agency. Three solution approaches are considered: the adapted local branching method described in Section 5.4.1, the windowing heuristic described in Section 5.4.2, and the branch-and-bound procedure of CPLEX 9.0. A second set of instances is then considered to show the performance of each of these procedures for generated shift scheduling problems traditionally considered in the shift scheduling literature.

5.5.1 Scheduling environments and problem size

The scheduling environments considered in the analysis operate 24 hours a day and seven days a week. The demand profiles are cyclic with a cycle length equal to one week. Thus, one need only determine the schedules on a one-week period.

The first set of instances includes three different operating environments, denoted hereafter by *ENV1*, *ENV2* and *ENV3*, respectively. Each environment is characterized by a list of admissible shift types and work stretch parameters. A shift type is specified by a start time, a length and the fractionable and/or unfractionable breaks it must receive. Instances provided by the company involve up to 21 shift types and almost all shift types include a unique fractionable break. For *ENV1*, all shift types include only fractionable breaks. For *ENV2* and *ENV3*, only one shift type incorporates a fixed break. In terms of work stretch parameters, all the instances consider three day-parts and thus three different values of minimum and maximum work stretch parameters during the day. Six demand patterns are

specified for the first and third environments and two are specified for the second environment, resulting in 14 real-life instances.

The second set of generated instances includes two operating environments, denoted hereafter by *ENV4* and *ENV5*, respectively. Environment *ENV4* considers nine-hour shifts starting every hour. Each shift must receive two 15-minute relief breaks and a 90-minute meal break. A break window is defined for each break and the minimum and maximum work stretch duration restriction parameters are set to 60 minutes and 180 minutes, respectively. Except the pre- and post-break work stretch duration constraints, the first hypothetical environment is a typical shift scheduling environment with multiple standard breaks and is often considered in the literature on shift scheduling problems (e.g., AYKIN, 2000). The second hypothetical environment, *ENV5*, offers more flexibility in the definition of shifts and breaks. It considers nine-hour shifts starting every 30 minutes. Each shift must receive a 90-minute fractionable break with a minimum and a maximum subbreak length of 15 minutes and 60 minutes, respectively. In order to ensure realistic operating conditions, the break profiles associated with shifts are restricted to those having exactly three subbreaks such that the length of the subbreak in the middle is strictly greater than the first and last ones, thus resulting in three different break profiles (15, 60, 15), (30, 45, 15) and (15, 45, 30). Finally, the minimum and maximum work stretch duration restrictions are set to 60 minutes and 180 minutes, respectively. For each of these operating environments, the six demand patterns of the air-traffic control agency are considered.

It is worth recalling that the implicit models used in this experimental analysis correspond to models (*P1*) and (*P2*) described in Sections 4.3.3 and 4.3.4 for which all forward and backward constraints are reformulated with slack variables as proposed by REKIK et al. (2005). In fact, the authors demonstrated, through

computational experiments, that such a reformulation considerably reduces the density of the constraint matrix and solution times. In the following, models (*P1*) and (*P2*) correspond to these models with reformulated forward and backward constraints. Table 5.5.1 reports the number of variables and constraints as well as the density of the constraint matrix of models (*P1*) and (*P2*) for the five scheduling environments.

Table 5.1 Model size for (*P1*) and (*P2*)

ENV.	Number of variables		Number of constraints		Constraint matrix density (%)	
	(P1)	(P2)	(P1)	(P2)	(P1)	(P2)
1	22,274	21,481	12,558	14,486	0.04	0.04
2	16,996	15,624	9,065	10,100	0.05	0.06
3	9,639	7,943	5,670	5,801	0.09	0.10
4	9,408	5,901	5,208	4,373	0.09	0.12
5	62,496	21,291	31,920	14,835	0.01	0.04

As one can see from Table 5.5.1, the experimental study proposed here includes small (environments 3 and 4), medium (environment 2) as well as large scheduling problems (environments 1 and 5). Except for environment 5, models (*P1*) and (*P2*) are almost equivalent in terms of the number of variables and constraints, and the matrix density. It is worth noticing that the density of both models is very low (no more than 0.12%). For environment 5, model (*P1*) has up to 62,496 variables and 31,920 constraints compared to only 21,291 variables and 14,835 constraints for model (*P2*).

5.5.2 Comparison of the solution approaches

The three solution approaches were applied to both formulations (*P1*) and (*P2*) on a 2.66 GHz Xeon PC with a time limit of 3600 seconds. The local branching strategy

and the windowing approach use CPLEX 9.0 with most parameters set at their default values. A branching strategy giving higher priority to shift variables S_j was considered for the three approaches. After a series of tests, we concluded that, on average, better solutions were obtained, for the three methods, with the “emphasis” parameter of CPLEX 9.0 set to “feasibility” (i.e., when the search concentrates on finding feasible solutions rather than optimal ones).

For the local branching strategy, the size of the neighborhood (k) is set to 10. In fact, after a series of tests with different values of k , we found that the value yielding better solutions on average was $k = 10$. The time allowed to explore a local branching node is set to 300 seconds (this parameter correspond in fact to the *node-time-limit* parameter defined by FISCHETTI and LODI (2003)). Finally, no more than five strong diversifications are permitted (this parameter corresponds to the *dv-max* parameter described in FISCHETTI and LODI (2003)). Recall that a strong diversification consists in restarting the local branching process from a new reference solution. The initial reference solution corresponds to the first feasible one given by CPLEX 9.0.

For the windowing approach, the seven-day planning horizon was divided into three time windows of two days each and a time window of one day. In fact, depending on the solution obtained for its predecessor, the current time window was enlarged as described in Section 5.4.2 to consider overlapping shifts. Each of these subproblems was solved within a time limit of 900 seconds.

The results obtained with the four approaches are reported in Tables 5.5.2 and 5.5.2. These tables display the LP and IP solution values obtained with each formulation for each instance as well as the time (in seconds) needed to find the IP solutions. It is worth recalling that the local branching method and the branch-and-bound

procedure of CPLEX 9.0 are applied for the whole shift scheduling problem over the week. An optimal solution was thus very difficult to obtain in 3600 seconds. Hence, we report in the IP value column the best feasible solution obtained and mark it with an asterisk when it is proven to be optimal. The IP time columns of the local branching strategy and the branch-and-bound procedure of CPLEX 9.0 display the time needed to find the best feasible solution during the search process. A dash ('-') in a column indicates that no feasible solution was identified within the time limit. For the windowing approach, the reported solution values and times are those obtained at the end of the whole process. If no optimal solution was found over a given time window, a solution time equal to 900 seconds, i.e., the time limit, is considered for the corresponding sub-problem. We also add in Tables 5.5.2 and 5.5.2 a column, denoted by "LB", which displays the best lower bound deduced from the classical branch-and-bound search of CPLEX (i.e., by rounding up the "best node" value).

When compared to the branch-and-bound procedure of CPLEX, the local branching strategy applied to model (*P1*) yielded a better feasible solution for 15 instances over the 22 (68.18%) for which an initial feasible solution was obtained within the time limit. This ratio reached 57.89% for model (*P2*) (11 instances over 19). The difference between objective values reached 8.57% (see demand pattern 4 for ENV4) for model (*P1*) and 5.65% for model (*P2*) (see demand pattern 3 for ENV4). For the seven remaining instances formulated with model (*P1*), both procedures resulted in the same objective value. The branch-and-bound procedure of CPLEX was, however, computationally superior since it consumes an average time of 49.58 seconds to find these solutions compared to 112.78 seconds on average for the local branching method. Moreover, the branch-and-bound procedure was able to prove the optimality of four of the seven equal solutions. When considering model (*P2*), the proposed local branching procedure yielded the same solution as the branch-

Table 5.2 Computational results for model (*P1*)

ENV	Demand pattern	LP	LB	Local Branching		CPLEX 9.0		Windowing	
				IP value	IP time	IP value	IP time	IP value	IP time
1	1	192.5	193	196	173.47	197	56.41	199	2701.70
	2	192.5	193	196	11.71	196	41.37	199	1807.90
	3	141.61	147	147	124.02	147*	40.66	153	36.80
	4	125.25	132	133	635.01	133	178.69	139	198.90
	5	183.38	186	189	21.47	190	368.47	192	2702.20
	6	199.91	202	204	2141.59	206	52.88	207	945.40
2	3	136.25	139	140	372.38	141	23.12	143	906.10
	4	119.00	122	126	4.84	126	65.05	131	2700.60
3	1	185.50	186	189	4.02	190	2.07	189	1803.00
	2	186.50	188	189	1973.24	190	15.62	189	3.30
	3	141.25	146	146	3.12	146*	10.88	146	3.30
	4	124.25	133	133	2.57	133*	1.51	133	2.60
	5	185.50	189	189	4.00	190	2.11	189	2.00
	6	200.57	203	203	8.24	203*	8.88	203	4.00
4	1	151.95	152	161	614.21	166	197.55	164	2553.00
	2	153.92	154	163	2321.72	167	446.25	162	1614.20
	3	117.32	118	124	42.00	129	105.83	124	36.20
	4	102.31	103	105	1775.72	114	26.83	109	1139.00
	5	163.66	164	170	551.86	180	14.81	169	1911.10
	6	177.85	178	183	121.49	193	54.72	185	2700.80
5	1	144.84	145	154	604.17	158	2101.28	149	203.20
	2	145.48	146	153	3106.53	160	3107.79	148	1202.70
	3	112.00	112	-	-	-	-	115	1403.76
	4	98.00	98	-	-	-	-	108	3391.50
	5	148.40	149	-	-	-	-	156	3600.00
	6	161.66	162	-	-	-	-	171	3600.00

Table 5.3 Computational results for model (*P2*)

ENV	Demand pattern	LP	LB	Local Branching		CPLEX 9.0		Windowing	
				IP value	IP time	IP value	IP time	IP value	IP time
1	1	192.50	193	197	1164.90	199	374.77	196	2704.40
	2	192.50	193	198	735.45	199	541.58	200	2705.00
	3	141.61	147	150	3431.28	147	1963.87	151	224.90
	4	125.25	132	135	2103.41	135	639.36	134	287.10
	5	183.36	186	190	1358.99	189	1978.77	189	2704.50
	6	199.91	202	205	1131.13	209	757.97	204	963.60
2	3	136.25	139	140	227.45	140	457.97	140	2748.50
	4	119.00	122	126	318.97	126	265.10	128	2701.60
3	1	185.50	186	189	96.50	190	104.25	196	17.10
	2	186.50	188	189	96.50	191	35.14	199	13.10
	3	141.25	146	146	1092.46	146	39.82	146	3.30
	4	124.25	133	133	155.53	133	53.29	135	1804.00
	5	185.50	189	189	241.99	189*	116.39	189	17.90
	6	200.57	203	203	115.99	204	116.39	204	14.00
4	1	151.95	152	159	1527.26	166	123.43	167	2095.90
	2	153.92	154	162	1995.79	165	313.30	168	1822.60
	3	117.32	118	124	1321.70	131	178.08	129	92.3
	4	102.31	103	107	3265.78	111	777.79	112	104.00
	5	163.66	164	168	1480.97	175	175.55	177	3600.00
	6	177.85	178	-	-	-	-	194	1108.60
5	1	144.84	145	-	-	-	-	154	1171.30
	2	145.48	146	-	-	-	-	156	2541.80
	3	112.00	112	-	-	-	-	115	1770.30
	4	98.00	98	-	-	-	-	104	1586.60
	5	148.40	149	-	-	-	-	162	2714.40
	6	161.658	162	-	-	-	-	177	2714.70

and-bound of CPLEX 9.0 for six instances (over 19) and worse solutions only for two instances; 2.04% worse for demand pattern 3 of ENV1, and 0.5% worse for demand pattern 5 of ENV1. The branch-and-bound procedure of CPLEX 9.0 is also computationally superior to the local branching strategy. In fact, it needed 261.98 seconds on average to find the six equal solutions compared to 689.96 seconds for the local branching strategy.

The results obtained with the windowing approach show, once again, that decomposing a continuous shift scheduling problem may considerably deteriorate the quality of the solution. This heuristic may appear inefficient for some easy instances that can easily be solved by local branching or traditional branch-and-bound procedures. However, as reported in Tables 5.5.2 and 5.5.2, except some instances (four in total for $(P1)$ and four for $(P2)$), the windowing method results in feasible solutions that are comparable to those obtained with the first two solution methods. For more complex instances, for which a first feasible solution is hard to find, the windowing approach is able to yield better solutions. For example, for demand pattern 2 of ENV5 formulated with $(P1)$, the windowing approach yielded a solution with 148 employees compared to 153 employees for the local branching method and to 157 employees for the branch-and-bound procedure of CPLEX. Moreover, the local branching and the classical branch-and-bound procedures were not able to find any feasible solution for four instances modeled with $(P1)$ and for seven instances formulated with $(P2)$. On those instances, the windowing approach gave good feasible solutions.

It is worth mentioning that a recent release of CPLEX (version 9.1) implements a local branching scheme. A new parameter controls whether this scheme is invoked by CPLEX or not. This parameter is off by default. We considered the branch-and-bound procedure of CPLEX 9.1 with this parameter turned on and found that

the solutions obtained were either as good as or worse than those obtained with our own local branching procedure.

5.6 Conclusion

This paper has considered a continuous shift scheduling problem that includes a high degree of flexibility in defining shifts and breaks. It extends the shift scheduling problem studied by REKIK et al. (2005) from one operating day to a w -day planning horizon. It thus incorporates the concepts of fractionable (or dividable) breaks and the pre- and post-break work stretch duration restrictions.

Two implicit formulations have been presented. These formulations extend those proposed by REKIK et al. (2005). They incorporate generalized forward and backward constraints reformulated with slack variables to reduce the matrix density.

The complexity of the continuous scheduling problem studied was essentially due to the presence of shifts that overlap from one day to the next. Two solution approaches have been proposed: a local branching and a windowing approach. The local branching method considers the problem as a whole over all the planning horizon. It is based on the local branching strategy initially introduced by FISCHETTI and LODI (2003) to solve complex MIP problems. We proposed particular local branching cuts that are specific to shift scheduling problems and used the same solution scheme as proposed by FISCHETTI and LODI (2003). The windowing approach decomposes the problem into subproblems, one for each time window of the planning horizon. These time windows are enlarged during the process to handle the overlapping shifts from one time window to the next. In fact, when comparing the two proposed solution approaches, the local branching approach has the advantage of considering

the overall problem and may thus yield better objective values than the windowing approach. Moreover, it may act as an exact solution method and yield optimal solutions for some easy instances. However, a local branching strategy starts from an initial feasible solution that may be very difficult to obtain for some instances. The windowing approach has the advantage of always yielding feasible solutions in reasonable time, even for complex shift scheduling problems. The solution quality and computing time needed will depend on the way time windows are defined.

We have compared the proposed approaches to the classical branch-and-bound procedure of CPLEX 9.0 for real-life instances obtained from an air-traffic control agency requirements and for some generated instances. Both local branching and windowing methods used CPLEX 9.0 to solve corresponding MIP sub-problems. The local branching method proposed yielded, on average, better objective values than the two other approaches. The results obtained with the windowing approach are a little worse especially for relatively easy instances. This was predictable since decomposing a continuous problem is likely to deteriorate the quality of the solution. However, on average, the solutions obtained with the windowing approach remain comparable to those obtained with the branch-and-bound procedure of CPLEX 9.0. For more difficult instances, whereas the local branching and the branch-and-bound procedure of CPLEX 9.0 were not able to identify any feasible solution, the windowing approach yielded good feasible solutions.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant CRDPJ228083-99 and by AD OPT Technologies Inc. This support is gratefully acknowledged.

Bibliography

- I. ADDOU AND F. SOUMIS. “Bechtold-Jacobs generalized model for shift scheduling with extraordinary overlap.” Technical report, GERAD, HEC Montréal, 2004.
- T. AYKIN. “Optimal shift scheduling with multiple break windows.” *Management Science*, **42**:591–602 (1996).
- T. AYKIN. “A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows.” *Journal of the Operational Research Society*, **49**:603–615 (1998).
- T. AYKIN. “A comparative evaluation of modeling approaches to the labor shift scheduling problem.” *European Journal of Operational Research*, **125**:381–397 (2000).
- S.E. BECHTOLD AND L.W. JACOBS. “Implicit modeling of flexible break assignments in optimal shift scheduling.” *Management Science*, **36**:1339–1351 (1990).
- S.E. BECHTOLD AND L.W. JACOBS. “Labor utilization effects of labor scheduling flexibility alternatives in a tour scheduling environment.” *Decision Sciences*, **24**:148–166 (1993).
- M.J. BRUSCO AND L.W. JACOBS. “Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations.” *European Journal of Operational Research*, **86**:249–261 (1995).
- M.J. BRUSCO AND L.W. JACOBS. “Starting-time decisions in labor tour scheduling: an experimental analysis and case study.” *European Journal of Operational Research*, **131**:459–475 (2001).

- T. ÇEZİK AND O. GÜNLÜK. “Reformulating linear programs with transportation constraints- with applications to workforce scheduling.” *Naval Research Logistics*, **51**:275–296 (2004).
- G.B. DANTZIG. “A comment on Edie’s traffic delays at toll booths.” *Operations Research*, **2**:339–341 (1954).
- M. FISCHETTI AND A. LODI. “Local branching.” *Mathematical Programming*, **B98**:23–47 (2003).
- W.B. HENDERSON AND W.L. BERRY. “Heuristic methods for telephone operator shift scheduling: an experimental analysis.” *Management Science*, **22**:1372–1380 (1976).
- N. MLADENOVIC AND P. HANSEN. “Variable neighborhood search.” *Computers and Operations Research*, **24**:1097–1100 (1997).
- S.L. MOONDRA. “An L.P. model for workforce scheduling in banks.” *Journal of Bank Research*, **6**:299–301 (1976).
- M. REKIK, J-F. CORDEAU AND F. SOUMIS. “Using Benders decomposition to implicitly model tour scheduling.” *Annals of Operations Research*, **128**:111–133 (2004).
- M. REKIK, J-F. CORDEAU AND F. SOUMIS. “Implicit shift scheduling with multiple breaks and work stretch duration restrictions.” Technical report, GERAD-2005-15, 2005.
- G.M. THOMPSON. “Improved implicit optimal modeling of the labor shift scheduling problem.” *Management Science*, **41**:595–607 (1995).
- G.M. THOMPSON. “A simulated annealing heuristic for shift scheduling using non-continuously available employees.” *Computers and Operations Research*, **32**:275–288 (1996a).

S. TOPALOGLUA AND I. OZKARAHAN. “Implicit optimal tour scheduling with flexible break assignments.” *Computers & Industrial Engineering*, **44**:75–89 (2002).

DISCUSSION GÉNÉRALE ET CONCLUSION

Les problèmes de construction d'horaires de travail ont fait l'objet de plusieurs études depuis les dernières décennies tant au niveau de la modélisation qu'au niveau des méthodes de résolution. Au fur et à mesure que la recherche avançait, les problèmes considérés tenaient de plus en plus compte des contraintes de la vie réelle incorporant ainsi plusieurs formes de flexibilité dans la définition des quarts, des pauses et des cycles. Deux principales approches de modélisation ont été adoptées. La première approche, qualifiée d'explicite, utilise un modèle de recouvrement généralisé dans lequel chaque horaire acceptable est représenté par une variable de décision. La deuxième approche, qualifiée d'implicite, essaie de réduire la taille de l'information fournie au modèle en représentant implicitement certaines des données. Des méthodes exactes basées sur une procédure de séparation et d'évaluation progressive avec ou sans coupes ainsi que des heuristiques adaptées ou basées sur la recherche locale ont été proposées pour résoudre les problèmes de construction d'horaires.

Le premier axe de recherche de cette thèse s'est orienté vers l'aspect modélisation du problème. Nous nous sommes particulièrement intéressés à l'approche implicite devenue de plus en plus populaire ces dernières années. Un des principaux apports dans ce domaine a été de représenter implicitement le placement des pauses au sein des quarts par le biais d'un ensemble restreint de contraintes connues sous le nom de contraintes "*forward*" et "*backward*". L'utilisation de ce nombre réduit de contraintes suppose néanmoins que les fenêtres de temps à l'intérieur desquelles les pauses sont placées sont formées de périodes continues et que l'extra-chevauchement est absent. La première contribution de cette thèse a été alors de généraliser le

concept des contraintes *forward* et *backward*, étendant ainsi leur application à plusieurs autres aspects du problème. Nous avons d'abord montré que les contraintes *forward* et *backward* représentent des coupes de réalisabilité de Benders résultant de la reformulation de Benders d'un modèle simple et général basé sur un problème de transport. Nous avons ensuite démontré que ces coupes sont suffisantes lorsque le problème de transport correspondant possède une structure particulière. Dès lors, la réalisabilité de n'importe quel problème de transport ayant cette structure particulière est assurée par un ensemble relativement restreint de contraintes qui remplaceraient le modèle d'affectation traditionnellement utilisé.

Le modèle basé sur le problème de transport a été en fait proposé pour un problème de construction simultanée de cycles et de quarts de travail incorporant plusieurs formes de flexibilité. La simplicité et la généralité de ce modèle permettent de traiter une panoplie de problèmes d'horaires dans lesquels aucune, ou peu de restrictions sont imposées. La preuve fournie pour les contraintes *forward* et *backward* donne en fait la forme générale des coupes de réalisabilité de Benders nécessaires dans le cas où les problèmes de transport utilisés sont quelconques. Nous offrons ainsi un modèle implicite, certes de grande taille, mais aussi et surtout très général se prêtant facilement à une décomposition de Benders pour laquelle nous avons défini toutes les coupes nécessaires.

La généralisation des contraintes *forward* et *backward* a permis de représenter implicitement des formes de flexibilité autres que le placement des pauses dans les fenêtres de temps correspondantes. Nous avons appliqué ce résultat pour le concept de bandes d'heures de début propre au problème de construction simultanée de cycles et de quarts ainsi que pour le concept de travail consécutif minimum et maximum rencontré dans les problèmes de construction de quarts de travail.

Les bandes d'heures de début limitent le champ d'heures de début de travail possibles à travers les jours du cycle. Nous avons montré que ce concept pouvait être modélisé par un ensemble de problèmes de transport et que sous certaines hypothèses (absence d'extra-chevauchement et contiguité des bandes), chacun de ces problèmes de transport peut être remplacé par des contraintes *forward* et *backward* appropriées. En comparant cette approche à l'approche classique de modélisation des bandes d'heures de début, une réduction importante du nombre de variables est obtenue avec l'approche utilisant les contraintes *forward* et *backward* contre une légère augmentation du nombre de contraintes. Nous avons également montré que les restrictions sur la durée de travail consécutif à l'intérieur du quart est modélisable avec un ensemble de problèmes de transport répondant aux conditions d'utilisation des contraintes *forward* et *backward*. Nous avons comparé notre nouvelle approche implicite à une approche explicite où toutes les possibilités de placement des fenêtres de pauses sont considérées. Nous avons une fois de plus prouvé la supériorité de l'approche implicite du point de vue de la taille du modèle et du temps de calcul.

Une autre contribution de cette thèse a été de comparer les performances de deux approches de modélisation implicite les plus utilisées dans la littérature, à savoir : l'approche d'AYKIN et celle de BECHTOLD et JACOBS. Nous avons d'abord démontré que les relaxations linéaires des deux modèles sont équivalentes. Ce résultat stipule que les modèles en nombres entiers correspondants induisent le même saut d'intégrité, éliminant ainsi un critère de choix possible de l'une ou l'autre de ces approches. Nous avons ensuite montré, à travers une analyse expérimentale, que la matrice des contraintes du modèle d'AYKIN est nettement moins dense que celle de BECHTOLD et JACOBS. Dans une étude comparative ultérieure, AYKIN avait déjà abouti à ces mêmes conclusions. Il s'est alors demandé si la supériorité de son modèle, observée pour les instances considérées, n'était pas due à cette différence

notable entre les densités. En fait, les instances difficiles de notre étude expérimentale ont donné presque toujours avantage au modèle de BECHTOLD et JACOBS du point de vue du nombre de variables et de contraintes et à celui d'AYKIN du point de vue de la densité de la matrice. La différence entre les densités est en fait due aux contraintes *forward* et *backward* présentes dans le modèle de BECHTOLD et JACOBS. Une autre contribution de cette thèse a été de proposer une reformulation des contraintes *forward* et *backward* qui réduit la densité de la matrice des contraintes. La reformulation proposée convertit convenablement les contraintes d'inégalité correspondant aux contraintes *forward* et *backward* en des contraintes d'égalité en définissant des variables d'écart continues. Cette reformulation permet de garder le même nombre de contraintes que la formulation originale et d'éviter aux variables additionnelles qu'elle définit d'augmenter la complexité du processus de résolution. Nous avons montré, à travers des résultats numériques, que la reformulation proposée des contraintes *forward* et *backward* réduit considérablement la densité et donc les temps de résolution par rapport à la formulation classique. Avec cette reformulation et pour les instances considérées, le modèle de BECHTOLD et JACOBS est en moyenne supérieur à celui d'AYKIN en termes de temps de calcul et de nombre d'instances résolues à l'optimalité.

Une autre contribution importante de cette thèse a été d'introduire une nouvelle forme de flexibilité dans la définition des pauses : les *pauses décomposables*. Contrairement à ce qui a été traditionnellement considéré dans la littérature, nous avons traité un problème de construction de quarts de travail dans lequel la pause peut se décomposer en plusieurs sous-pauses à l'intérieur du quart. Avec ce nouveau concept, nous offrons aux organisations la possibilité de préciser uniquement la durée totale des pauses à accorder au quart sans se soucier du nombre, de la durée, ni de la position de chaque sous-pause au sein du quart. Ces valeurs sont optimisées

par le modèle. Le concept de pauses décomposables, apparemment complexe, a été modélisé de façon simple et efficace. Les modèles ainsi obtenus ont été résolus en des temps relativement courts.

L'impact des formes de flexibilité sur le coût de la main-d'œuvre a fait l'objet de plusieurs études. Ces études ont pour la plupart rapporté les gains importants induits par l'introduction de l'une ou l'autre de ces formes. Le concept de pauses décomposables étant nouveau, nous nous sommes intéressés à son impact sur la taille de la main-d'œuvre. Les résultats obtenus montrent que l'utilisation des pauses décomposables pouvait réduire considérablement le nombre total d'employés requis par rapport aux environnements d'opération où seules les pauses standards (non décomposables) sont utilisées. D'autre part, les problèmes de construction de quarts avec plusieurs pauses considérés dans la littérature utilisent des fenêtres de pauses qui sont fixées d'avance. Ces fenêtres sont en général suffisamment espacées pour éviter que deux pauses successives soient très proches. Le placement de ces fenêtres de pauses est décidé arbitrairement ou selon certaines approches telles que l'approche basée sur le temps idéal d'heures de début de pauses. Le concept de restrictions sur la durée de travail consécutif minimum et maximum constitue, en fait, une autre approche de placement des fenêtres de pauses. Une autre contribution de cette thèse a été de prouver que cette nouvelle approche définit des quarts résultant en un nombre moins élevé d'employés que celui requis par l'approche classique avec fenêtres pré-fixées. Plus encore, la présence de restrictions sur la durée de travail consécutif minimum et maximum assure des quarts de bonne qualité où les périodes de travail et de repos sont convenablement distribuées.

Cette thèse s'est aussi intéressée à l'aspect résolution du problème. Des instances de la vie réelle fournies par une agence de contrôleurs aériens ainsi que des instances

hypothétiques générées par la variation de certains paramètres du problème ont été considérées. Tous les environnements étudiés opèrent de façon continue, 24 heures sur 24 et sept jours sur sept. Ces environnements incorporent, à notre connaissance, toutes les formes de flexibilité considérées dans la littérature jusqu'à en plus des pauses décomposables et des restrictions sur la durée de travail consécutif minimum et maximum. Nous avons proposé deux approches de résolution du problème. La première approche est basée sur la méthode de branchement local récemment proposée pour la résolution des problèmes en nombres entiers relativement complexes. S'inspirant de cette méthode, nous avons proposé des coupes de branchement local spécifiques aux problèmes de construction de quarts. Nous avons montré que, dans les limites de temps de calcul fixées, le branchement local proposé donne une solution meilleure que celle fournie par la méthode classique de séparation et d'évaluation progressive de CPLEX 9.0. La deuxième approche de résolution proposée est basée sur une décomposition temporelle du problème en différents sous-problèmes, un pour chaque fenêtre de temps. Les fenêtres de temps sont allongées au courant du processus de résolution pour tenir compte de l'aspect "chevauchement des quarts entre les fenêtres de temps", cause principale de la complexité des problèmes d'horaires continus. Cette deuxième approche s'est révélée efficace pour les problèmes difficiles. En effet, alors qu'aucune solution réalisable n'a été identifiée par le branchement local ni par la méthode de séparation et d'évaluation progressive de CPLEX, l'approche de décomposition par fenêtres de temps a produit une solution de bonne qualité en un temps relativement court.

Les problèmes de construction d'horaires abordés dans cette thèse avaient tous pour but de construire des horaires anonymes où le côté affectation aux employés a été ignoré. Plusieurs des concepts utilisés restent cependant assez généraux et pourraient être appliqués aux problèmes de construction d'horaires individuels. Un

défi serait d'aborder le problème au complet, c'est-à-dire, de construire les cycles et les quarts de travail simultanément et de les attribuer aux employés. Ceci passerait probablement par une représentation implicite des données du problème et par une décomposition du problème global sur un ou plusieurs aspects.

BIBLIOGRAPHIE

- ADDOU, I. ET SOUMIS, F. Bechtold-Jacobs generalized model for shift scheduling with extraordinary overlap. Rapport technique, GERAD, HEC Montréal, (2004).
- ALFARES, H.K., (1998). An efficient two-phase algorithm for cyclic days-off scheduling. *Computers & Operations Research*, 25(11) :913–923.
- AYKIN, T. Reformulating the tour scheduling problem to improve solvability, (1995). GSM Working paper 95-29 , Graduate School of Management, Rutgers University, Newark, NJ.
- AYKIN, T., (1996). Optimal shift scheduling with multiple break windows. *Management Science*, 42 :591–602.
- AYKIN, T., (1998). A composite branch and cut algorithm for optimal shift scheduling with multiple breaks and break windows. *Journal of the Operational Research Society*, 49 :603–615.
- AYKIN, T., (2000). A comparative evaluation of modeling approaches to the labor shift scheduling problem. *European Journal of Operational Research*, 125 :381–397.
- BAILEY, J., (1985). Integrated days off and shift personnel scheduling. *Computers and Industrial Engineering*, 9 :395–404.
- BAILEY, J. ET FIELDS, J., (1985). Personnel scheduling with flexshift models. *Journal of Operations Management*, 5 :327–338.
- BAKER, K. ET MAGAZINE, M.J., (1977). Workforce scheduling with cyclic demands and day-off constraints. *Management Science*, 24 :161–167.
- BAKER, K.R., (1974). Scheduling a full-time workforce to meet cyclic staffing requirements. *Management Science*, 20 :1561–1568.

- BAKER, K.R., (1976). Workforce allocation in cyclic scheduling problems : A survey. *Operational Research Quarterly*, 27 :155.
- BAKER, K.R., BURNS, R.N. ET CARTER, M.W., (1979). Staff scheduling with day-off and workstretch constraints. *AIIE Transactions*, 11 :286–292.
- BARD, J.F., BINICI, C. ET DESILVA, A.H., (2003). Staff scheduling at the United States Postal Service. *Computers & Operations Research*, 30 :745–771.
- BARTHOLDI, J.J., (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, 29 :501–510.
- BARTHOLDI, J.J., ORLIN, J.B. ET RATLIFF, H.D., (1980). Cyclic scheduling via integer programs with circular ones. *Operations Research*, 28 :1074–1085.
- BARTHOLDI, J.J. ET RATLIFF, H.D., (1978). Unnetworks, with applications to idle time scheduling. *Management Science*, 24 :850–858.
- BECHTOLD, S.E. ET BRUSCO, M.J., (1994). Working set generation methods for labor tour scheduling. *European Journal of Operational Research*, 74 :540–551.
- BECHTOLD, S.E., BRUSCO, M.J. ET SHOWALTER, M.J., (1991). A comparative evaluation of labor tour scheduling methods. *Decision Sciences*, 22 :683–699.
- BECHTOLD, S.E. ET JACOBS, L.W., (1990). Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science*, 36 :1339–1351.
- BECHTOLD, S.E. ET JACOBS, L.W., (1993). Labor utilization effects of labor scheduling flexibility alternatives in a tour scheduling environment. *Decision Sciences*, 24 :148–166.
- BECHTOLD, S.E. ET JACOBS, L.W., (1996). The equivalence of general set-covering and implicit integer programming formulations for shift scheduling. *Naval Research Logistics*, 43 :233–249.

- BECHTOLD, S.E. ET SHOWALTER, M.J., (1987). A methodology for labor scheduling in a service delivery system. *Decision Science*, 18 :89–107.
- BLAIS, J.Y., LAMONT, J. ET ROUSSEAU, J.M., (1990). The hastus vehicle and manpower scheduling at the Société de Transport de la Communauté Urbaine de Montréal. *Interfaces*, 20 :26–42.
- BROWNWELL, W.S. ET LOWERRE, J.W., (1976). Scheduling of work forces required in continuous operations under alternative labor policies. *Management Science*, 22 :597–605.
- BRUSCO, M.J. ET JACOBS, L.W., (1993). A simulated annealing approach to the cyclic staff-scheduling problem. *Naval Research Logistics*, 40 :69–84.
- BRUSCO, M.J. ET JACOBS, L.W., (1995). Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations. *European Journal of Operational Research*, 86 :249–261.
- BRUSCO, M.J. ET JACOBS, L.W., (2000). Optimal models for meal-break and start-time flexibility in continuous tour scheduling. *Management Science*, 46 :1630–1641.
- BRUSCO, M.J. ET JACOBS, L.W., (2001). Starting-time decisions in labor tour scheduling : an experimental analysis and case study. *European Journal of Operational Research*, 131 :459–475.
- BRUSCO, M.J., JACOBS, L.W., BONGIORNO, R.J., LYONS, D. V. ET TANG, B., (1995). Improving personnel scheduling at airline stations. *Operations Research*, 43 :741–751.
- BUFFA, E.S., COSGROVE, M.S. ET LUCE, B.J., (1976). An integrated work shift scheduling system. *Decision Science*, 7 :620–630.

- BURNS, R.N., (1978). Manpower scheduling with variable demands and alternate week-ends off. *INFOR*, 16 :101–111.
- BURNS, R.N. ET CARTER, M.W., (1985). Work force size and single shift schedules with variables demands. *Management Science*, 31 :599–607.
- BYRNE, J.L. ET POTTS, R.B., (1973). Scheduling of toll collectors. *Transportation Science*, 7 :224–245.
- ÇEZİK, T. ET GÜNLÜK, O., (2004). Reformulating linear programs with transportation constraints- with applications to workforce scheduling. *Naval Research Logistics*, 51 :275–296.
- ÇEZİK, T., GÜNLÜK, O. ET LUSS, H., (2001). An integer programming model for the weekly tour scheduling problem. *Naval Research Logistics*, 48 :608–624.
- DANTZIG, G.B., (1954). A comment on Edie's traffic delays at toll booths. *Operations Research*, 2 :339–341.
- EASTON, F.F. ET ROSSIN, D.F., (1991). Sufficient working subsets for the tour scheduling problem. *Management Science*, 37 :1441–1451.
- EMMONS, H., (1985). Workforce scheduling with cyclic requirements and constraints on days off, weekends off, and workstretch. *IIE Transacrction*, 17 :8–16.
- EMMONS, H. ET BURNS, R.N., (1991). Off-day scheduling with hierarchical worker categories. *Operations Research*, 39 :484–495.
- ERNST, A.T., JIANG, H., KRISHNAMOORTHY, M. ET SIER, D., (2003). Staff scheduling and rostering : A review of applications, methods and models. *European Journal of operational Research*.
- FISCHETTI, M. ET LODI, A., (2003). Local branching. *Mathematical Programming*, B98 :23–47.

- GABALLA, A. ET PEARCE, W., (1979). Telephone sales manpower planning at Quantas. *Interfaces*, 9 :1-9.
- GAMACHE, M., SOUMIS, F., MARQUIS, G. ET DEROSIERS, J., (1999). A column generation approach for large scale aircrew rostering problems. *Operation research*, 47(2) :247-262.
- HENDERSON, W.B. ET BERRY, W.L., (1976). Heuristic methods for telephone operator shift scheduling : an experimental analysis. *Management Science*, 22 :1372-1380.
- HOEY, J.M. ET LOVE, R.R., (1990). Management Science improves fast-food operations. *Interfaces*, 20 :21-29.
- HOFFMAN, K.L. ET PADBERG, M., (1993). Solving airline crew scheduling problems by branch-and-cut . *Management Science*, 39 :657-682.
- HUETER, J. ET SWART, W., (1998). An integrated labor-management system for Taco-Bell. *Interfaces*, 28 :75-91.
- JACOBS, L.W. ET BRUSCO, M.J., (1996). Overlapping start-time bands in implicit tour scheduling. *Management Science*, 42 :1247-1259.
- JACOBS, L.W. ET BRUSCO, M.J., (1998). Personnel tour scheduling when starting time restrictions are present. *Management Science*, 44 :534-547.
- JARRAH, A.I.Z, BARD, J.F ET DESILVA, H., (1994). Solving large-scale tour scheduling problems. *Management Science*, 40 :1124-1144.
- KEITH, E.G., (1979). Operator scheduling. *AIIE Transactions*, 11 :37-41.
- MABERT, V. ET SHOWALTER, M., (1988). An evaluation of full-/part-time tour scheduling methodology. *International Journal of Operations and Production Management*, 9 :54-71.

- MABERT, V.A. ET SHOWALTER, M.J., (1990). Measuring the impact of part-time workers in service organizations. *Journal of Operational Management*, 9 :209–229.
- MABERT, V.A. ET WATTS, C.A., (1982). A simulation analysis of tour-shift scheduling construction procedures. *Management Science*, 28 :520–532.
- MCGINNIS, L.F., CULVER, W.D. ET DEANE, R.H., (1978). One- and two-Phase heuristics for workforce scheduling. *Computers and Industrial Engineering*, 2 :7–15.
- MLADENOVIC, N. ET HANSEN, P., (1997). Variable neighborhood search. *Computers and Operations Research*, 24 :1097–1100.
- MOONDRA, S.L., (1976). An L.P. model for workforce scheduling in banks. *Journal of Bank Research*, 6 :299–301.
- MORRIS, J.G. ET SHOWALTER, M.J., (1983). Simple approaches to shift, days-off and tour scheduling problems. *Management Science*, 29 :942–950.
- NEMHAUSER, G. ET WOLSEY, L. *Integer and combinatorial optimization*. Wiley, New York, (1988).
- REKIK, M., CORDEAU, J-F. ET SOUMIS, F., (2004). Using Benders decomposition to implicitly model tour scheduling. *Annals of Operations Research*, 128 :111–133.
- REKIK, M., CORDEAU, J-F. ET SOUMIS, F. Implicit shift scheduling with multiple breaks and work stretch duration restrictions. Rapport technique, GERAD-2005-15, (2005).
- SCHINDLER, S. ET SEMMEL, T., (1993). Station staffing at Pan American World Airways. *Interfaces*, 23 :91–98.
- SEGAL, M., (1974). The operator scheduling problem : a network flow approach. *Operations Research*, 22 :808–824.

- THOMPSON, G.M., (1995). Improved implicit optimal modeling of the labor shift scheduling problem. *Management Science*, 41 :595–607.
- THOMPSON, G.M., (1996a). A simulated annealing heuristic for shift scheduling using non-continuously available employees. *Computers and Operational Research*, 32 :275–288.
- TOPALOGLUA, S. ET OZKARAHAN, I., (2002). Implicit optimal tour scheduling with flexible break assignments. *Computers & Industrial Engineering*, 44 :75–89.
- VANCE, P.H., BARNHART, C., JOHNSON, E.L. ET NEMHAUSER, G.L., (1997). Airline crew scheduling : a new formulation and decomposition algorithm. *Operations Research*, 45 :188–200.