

UNIVERSITÉ DE MONTRÉAL

OPTIMISATION DES PAUSES DANS LE PROBLÈME DE FABRICATION
D'HORAIRES AVEC QUARTS DE TRAVAIL

MATHIEU BOUCHARD
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
JUN 2004

© Mathieu Bouchard, 2004.



Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97930-X

Our file Notre référence

ISBN: 0-612-97930-X

NOTICE:

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

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

AVIS:

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

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

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

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

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

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

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OPTIMISATION DES PAUSES DANS LE PROBLÈME DE FABRICATION
D'HORAIRES AVEC QUARTS DE TRAVAIL

présenté par : BOUCHARD Mathieu

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

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

M. SAVARD Gilles, Ph.D., président

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

M. SOUMIS François, Ph.D., membre

À mes parents

REMERCIEMENTS

J'aimerais tout spécialement remercier mon directeur M. Guy Desaulniers qui a été extraordinaire tout au long de ce mémoire. Sa disponibilité et ses conseils dans toutes les étapes de ce travail ont été plus qu'appréciés.

J'aimerais aussi remercier le groupe Gencol du GERAD pour son appui financier sans lequel il m'aurait été difficile de poursuivre mes études.

Durant ce mémoire, j'ai eu souvent à me référer à Eric Vatri, lequel s'est montré très disponible et patient pour répondre à mes questions. Je lui adresse mes plus sincères remerciements pour m'avoir accordé son temps.

Je tiens à dire merci à mes parents qui ont beaucoup contribué à l'épanouissement de mes intérêts dont ceux qui m'ont mené à entreprendre cette maîtrise. Ils ont toujours été et resteront toujours des modèles et une source d'inspiration.

Je remercie chaleureusement ma conjointe Hélène. Il m'est difficile d'imaginer comment j'aurais pu terminer ce mémoire sans son support, son amour et son aide.

RÉSUMÉ

Ce mémoire porte d'abord sur le problème d'assignation des activités à des quarts de travail dans un contexte multi-activités, l'objectif principal étant de satisfaire au maximum la demande pour les différentes activités tout en respectant un large éventail de contraintes et d'objectifs secondaires. Plus spécifiquement, nous adressons le problème de placer les pauses dans les quarts. Nous remplaçons une approche résolvant le problème dans le cas où les pauses contenues dans les quarts sont préalablement fixées, par un modèle et une méthodologie permettant de placer efficacement les pauses sans aucune information initiale à ce sujet. Ce modèle est du type multi-flots en nombres entiers avec contraintes additionnelles. La méthode de résolution utilisée consiste en une méthode heuristique de séparation et évaluation progressive intégrant une approche de génération de colonnes combinée à une stratégie de décomposition temporelle. Nous considérons plusieurs stratégies dans le but d'obtenir des solutions de qualité en des temps de résolution acceptables. Ces stratégies sont basées sur une méthode de résolution décomposant en deux phases la résolution des sous-problèmes de l'approche de génération de colonnes. La première phase génère un ensemble de pauses respectant les règles qui les régissent alors que la deuxième assigne les activités aux quarts en fixant les pauses générées lors de la première phase.

Nous considérons ensuite le problème plus général consistant à construire des quarts de travail tout en y assignant des activités et des pauses. L'approche proposée consiste à généraliser les stratégies que nous avons développées pour y inclure la construction des quarts. Cette généralisation est basée sur une approche existante qui modifie les réseaux sous-jacents aux problèmes d'assignation des activités afin de permettre la construction des quarts.

Nous comparons notre méthode à une approche existante qui décompose le problème

en trois étapes : construction des quarts de travail avec pauses, assignation des quarts aux employés et assignation des activités. Notons que notre approche intègre dans un même processus d'optimisation ces trois étapes. Les jeux de données utilisés pour effectuer ces comparaisons sont tirés d'instances réelles de problèmes de planification d'horaires pour des contrôleurs aériens sur un horizon variant d'une journée à un mois. Nos résultats démontrent que notre méthode permet des gains potentiels pouvant aller jusqu'à 24% en ce qui concerne la satisfaction de la demande pour les différentes activités. Les temps de résolution obtenus avec notre approche intégrée sont jugés acceptables.

ABSTRACT

This master's thesis considers first the problem of assigning activities to work shifts in a multi-activity context. The main objective is to maximize the satisfaction of demand for the activities while considering a large set of constraints and secondary objectives. In particular, we address for this problem the issue of positioning the breaks in the shifts. Starting from an approach that solves the problem in the case where the breaks are already fixed, we develop a model and a methodology that allows to position the breaks efficiently without any initial information about them. This problem is formulated as a multi-commodity network flow problem with side constraints. Integer solutions to the overall problem are derived through an heuristic branch-and-bound algorithm that embeds a column generation approach. A temporal decomposition strategy is also used to split the whole problem in smaller, easier-to-solve problems. To speed up the column generation approach, we study different strategies for solving the subproblems in two phases. The first phase generates admissible breaks while the second assigns activities to the work shifts using the breaks generated in the first phase.

The more general problem of simultaneously constructing work shifts and assigning activities and breaks in those shifts is then considered. We propose an approach that generalizes the strategies that we developed in the first part of this thesis to include the construction of the work shifts. This generalization is based on an existing approach that modifies the underlying networks of the activity assignment problem in a way that permits the construction of the work shifts.

Our approach is compared to a method that splits the problem in three steps: shift building, assignment of shifts to employees, and activity assignment. The approach

we propose integrates these three steps. The datasets used for the numerical experiments are real world instances of air traffic controller scheduling problems for horizons ranging from one day to one month. The results computed with our method show a potential improvement of up to 24% on the satisfaction of demand for the activities. Our approach exhibits processing times that are deemed acceptable.

TABLE DES MATIÈRES

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLE DES MATIÈRES	x
LISTE DES TABLEAUX	xv
LISTE DES FIGURES	xvii
LISTE DES ALGORITHMES	xviii
INTRODUCTION	1
CHAPITRE 1 : REVUE DE LITTÉRATURE	6
1.1 : Construction des cycles	7

1.2 : Construction des quarts	7
1.3 : Construction des cycles et des quarts	11
1.4 : Affectation d'activités à l'intérieur des quarts	13
1.5 : Fabrication d'horaires avec quarts de travail	13
1.6 : Contributions et justifications	14
 CHAPITRE 2 : LE PROBLÈME D'ATTRIBUTION DES ACTIVITÉS AVEC PAUSES FIXÉES	 16
2.1 : Version simplifiée du problème	16
2.2 : Modèle de base	17
2.2.1 : Formulation mathématique	17
2.2.2 : Les sous-problèmes	19
2.3 : Méthode de résolution	26
2.3.1 : Stratégie de décomposition temporelle	27
2.3.2 : Calcul d'une borne inférieure par génération de colonnes . .	28
2.3.3 : Recherche d'une solution entière	29
2.4 : Extensions	30
2.4.1 : Durée d'affectation maximale et minimale variable	31

2.4.2 : Contraintes sur les activités stressantes	31
2.4.3 : Traitement des activités fermées	31
2.4.4 : Respect de la durée de base	32
2.4.5 : Répartition de la sous-couverture entre les périodes	32
2.4.6 : Répartition de la sous-couverture entre les activités	33
2.4.7 : Respect des distributions cibles pour l'entretien des compétences	33
2.4.8 : Traitement de la formation des stagiaires	34
2.4.9 : Traitement de la réserve	34
2.4.10 : Traitement des quarts avec conditions initiales	35
2.4.11 : Traitement de la sur-couverture	35
CHAPITRE 3 : OPTIMISATION DES PAUSES	37
3.1 : Notation concernant les pauses	37
3.2 : Réoptimisation des pauses initiales	39
3.2.1 : Méthode de perturbation des pauses initiales	40
3.3 : Placement des pauses dans les quarts	44
3.3.1 : Le modèle de pauses exact	45

3.3.2 : Modélisation des sous-problèmes permettant la génération des pauses par agrégation des activités	48
3.3.3 : La méthode à deux phases	50
3.3.4 : Critères déterminant le potentiel d'un ensemble de pauses .	51
3.4 : Résultats numériques	54
3.4.1 : Génération continue des pauses par agrégation des activités	56
3.4.2 : Génération intermittente des pauses par agrégation des activités	56
3.4.3 : Génération intermittente des pauses par agrégation des acti- vités avec flexibilité	57
3.4.4 : Génération intermittente des pauses avec boucle utilisant le modèle exact	58
3.4.5 : Génération intermittente des pauses avec boucle utilisant le modèle exact avec flexibilité	59
3.5 : Conclusion	61
CHAPITRE 4 : INTÉGRATION À L'ISGAA	64
4.1 : Définitions et notation concernant les quarts	64
4.2 : Présentation du problème et de l'ISGAA	65
4.3 : Intégration des pauses dans le modèle de l'ISGAA	68

4.3.1 : Ajout des ressources de pause existantes	69
4.3.2 : Respect de la contrainte sur la durée séparant deux pauses consécutives	70
4.3.3 : Performance de la méthode de résolution de l'ISGAA	71
4.4 : Généralisation de la méthode à deux phases pour l'ISGAA	72
4.4.1 : Modélisation des sous-problèmes permettant la génération de quarts et de pauses par agrégation des activités	73
4.4.2 : Modélisation des sous-problèmes permettant de remplir d'ac- tivités les quarts de travail avec pauses fixées	73
4.4.3 : Stratégie utilisée avec la méthode à deux phases de l'ISGAA	74
4.4.4 : Vision à moyen terme	75
4.5 : Résultats numériques	77
4.6 : Conclusion	80
CONCLUSION	82
BIBLIOGRAPHIE	84

LISTE DES TABLEAUX

Tableau 2.1 : Fonctions de prolongation de la ressource $rNum$	24
Tableau 2.2 : Fonctions de prolongation de la ressource $rDuree$	25
Tableau 3.1 : Fonctions de prolongation de la ressource $rPau$	42
Tableau 3.2 : Consommation des ressources $rSansPauMin$ et $rSansPauMax$	44
Tableau 3.3 : Consommation de la ressource $rDurPauMax$	46
Tableau 3.4 : Consommation des ressources $rDurTotPauMin$ et $rDurTotPauMax$	47
Tableau 3.5 : Résultats pour la méthode de génération continue des pauses par agrégation des activités	55
Tableau 3.6 : Résultats pour la méthode de génération continue des pauses par agrégation des activités	57
Tableau 3.7 : Résultats pour la méthode de génération intermittente des pauses par agrégation des activités	58
Tableau 3.8 : Résultats pour la méthode de génération intermittente des pauses par agrégation des activités avec flexibilité	59

Tableau 3.9 : Résultats pour la méthode de générations intermittente des pauses avec boucle utilisant le modèle exact	60
Tableau 3.10 : Résultats pour la méthode de génération intermittente des pauses avec boucle utilisant le modèle exact avec flexibilité	61
Tableau 3.11 : Répartition des temps de résolution	63
Tableau 4.1 : Consommation des ressources $rMinPauTotDur$ et $rMaxPauTotDur$	68
Tableau 4.2 : Consommation des ressources $rMinPauTotDur$ et $rMaxPauTotDur$	70
Tableau 4.3 : Intrants des fonctions de prolongation des ressources $rSansPauMin$, $rSansPauMax$, $rCbQmin$ et $rCbQmax$	71
Tableau 4.4 : Résultats de l'ISGAA pour les mêmes fenêtres de temps . .	78
Tableau 4.5 : Résultats de l'ISGAA pour des temps de résolution de même ordre de grandeur	79
Tableau 4.6 : Répartition des temps de résolution pour l'ISGAA	81

LISTE DES FIGURES

Figure 2.1 : Un exemple de réseau $G^k = (V^k, E^k)$	36
Figure 3.1 : Une partie d'un réseau G'^k utilisé par la méthode de perturbation des pauses initiales	41
Figure 3.2 : Réseau agrégé utilisé lors de la première phase	49
Figure 4.1 : Réseau utilisé par l'ISGAA	67
Figure 4.2 : Fenêtres d'optimisation dans la vision à moyen terme	77

LISTE DES ALGORITHMES

Algorithme 3.1 : Algorithme pour la génération continue des pauses par agrégation des activités	51
Algorithme 3.2 : Algorithme pour la génération intermittente des pauses par agrégation des activités	52
Algorithme 3.3 : Algorithme pour la génération intermittente des pauses avec boucle utilisant le modèle exact	53
Algorithme 4.1 : Algorithme pour la génération intermittente des pauses et des quarts par agrégation des activités	75

INTRODUCTION

Dans tous les secteurs de l'économie faisant un usage intensif de ressources humaines, l'utilisation optimale de ces ressources est un enjeu majeur. Un emploi efficient de celles-ci n'est possible que si les horaires attribués aux membres du personnel sont efficaces. Le fait que le problème consistant à fabriquer ces horaires se formule généralement assez bien dans un langage mathématique et la progression rapide du secteur des technologies de l'information, ont contribué à l'essor de l'informatisation du processus de planification des horaires. En plus de permettre dans bien des cas d'améliorer l'allocation des ressources humaines, les logiciels de planification d'horaires ont aussi soulagé nombre d'administrateurs d'une tâche difficile et astreignante.

Le besoin pour ces logiciels est présent dans de nombreux secteurs, centres d'appels, banques, hôpitaux, service de transport, pour n'en nommer que quelques uns. C'est pourquoi la demande d'outils facilitant et améliorant la répartition de la charge de travail est aussi nombreuse que diversifiée. Néanmoins ces problèmes peuvent se regrouper en quelques catégories, notamment les problèmes d'horaires de membre d'équipage (Desrochers *et al.*, 1992 et Gamache *et al.* 1999) et les problèmes d'horaires avec quarts de travail (Brusco et Jacobs, 1998). Un équipage étant par définition mobile, l'offre de personnel en un temps et en un lieu donné dépend fortement des tâches auxquels les employés sont affectés, ce qui n'est pas le cas pour les horaires avec quarts de travail où les employés restent sur place. Ce facteur et d'autres sont suffisamment importants pour les considérer comme deux problèmes distincts.

Ce mémoire s'intéresse à un problème spécifique d'horaires avec quarts de travail concernant les contrôleurs aériens. Nous nommerons ce problème : le problème de fabrication des horaires avec quarts de travail (PFHQ). Une solution du PFHQ pour

une période donnée, contient, pour chaque employé, toutes les informations relatives à son emploi du temps : jours de travail, période et durée des quarts de travail, nature du travail à accomplir dans chaque quart. Les solutions ainsi fournies doivent être en accord avec la convention collective des employées, qui dans ce problème spécifique, comporte de nombreuses contraintes étant donnée la nature stressante du travail. La qualité d'une solution dépend de son coût total, de la couverture de la demande pour les différentes activités et de la satisfaction des préférences des employés.

Le problème est défini par plusieurs règles issues de la convention collective. Une partie de ces règles concerne la façon dont les jours de travail et de repos peuvent être affectés aux employés. D'autres déterminent comment les quarts de travail doivent être créés : temps de début, longueur, nature des pauses les composant. Pour un employé donné, des contraintes assurent un repos minimal entre deux quarts consécutifs et assurent à ce dernier un certain nombre d'heures de travail. Finalement, l'affectation d'une activité à un employé n'est possible que si celui-ci possède les compétences nécessaires.

Plusieurs logiciels fournissent une assistance au service des ressources humaines en les guidant dans la conception d'horaires respectant toutes les règles qui leur sont imposées. D'autres automatisent complètement le processus de création d'horaires en essayant autant que faire se peut de fournir l'horaire satisfaisant au maximum les objectifs établis. La deuxième classe de logiciels est largement favorisée lorsque la complexité du problème est tel qu'il est virtuellement impossible d'optimiser manuellement les horaires. Dans ce mémoire, nous travaillerons avec un logiciel appartenant à la deuxième catégorie.

Le logiciel ShiftLogic développé par la compagnie Technologies Ad Opt inc. a été conçu pour résoudre le PFHQ pour les contrôleurs aériens. Étant donnée la complexité des problèmes auxquels il doit faire face (horizon d'un mois, entre cinquante et cent

employés et une dizaine d'activités), ce logiciel fait appel à une heuristique permettant de décomposer le problème en quatre problèmes distincts plus faciles à résoudre. Les quatre modules suivent une décomposition naturelle du problème soit :

- Déterminer les jours de travail des employés.
- Créer des quarts de travail anonymes et les pauses qu'ils contiennent.
- Affecter les quarts aux employés.
- Remplir d'activités les quarts de travail.

Dans la première partie, le logiciel détermine quels jours un employé travaillera durant la période de planification. Les choix devront respecter la demande quotidienne en employés, tout en s'assurant que les employés affectés pour une date donnée pourront couvrir la demande pour les activités planifiées lors de cette journée. De plus, le module déterminera si l'employé effectuera un quart de jour ou de nuit. Pour ce faire, les règles suivantes doivent être respectées :

- Chaque employé doit travailler un certain nombre d'heures par semaine ou par mois.
- Le nombre de jours de travail et de repos d'un employé, de même que les patrons de quarts (quart de jour ou de nuit) doivent respecter une certaine configuration pré-établie.

Une fois établi, le nombre d'employés pour chaque période de la journée (jour et nuit) permet au deuxième module de générer un nombre de quarts correspondant au nombre d'employés. L'ensemble des quarts ainsi produits doit permettre de couvrir autant que possible la demande en employés par activité pour chaque unité de période de l'horizon de planification. Pour ce faire, le logiciel considère, pour chaque période, la demande totale toute activité confondue. Selon le type de quart généré, des contraintes sur la longueur et la période de début du quart s'appliquent. Autre

contrainte, un nombre donné de quarts de chaque type doit être généré. Il peut s'agir d'une quantité minimum ou maximum ou encore d'une proportion cible du nombre total de quarts. De plus, les périodes de pause sont choisies pour chaque quart selon certaines règles déterminant la durée totale de pause pour un quart, la durée individuelle des périodes de pause, l'intervalle de temps séparant deux pauses et, dans certains cas, l'intervalle de temps durant lequel la pause peut commencer dans le quart.

Le troisième module a pour but de déterminer quel employé travaillera à chacun des quarts créés. De toute évidence, un employé ne peut être assigné qu'à des quarts se déroulant sur une période de la journée où ce dernier travaille. Aussi, le module doit voir à ce que les deux quarts appartenant à un même employé soient séparés par une durée minimale de repos. De plus, les quarts doivent être attribués de manière à respecter la moyenne hebdomadaire d'heures de travail par employé. L'objectif lors de cette étape est de maximiser la satisfaction des employés quant à leur préférence pour les différents types de quart.

Le dernier module s'attaque à la composante multi-activités du problème. Ainsi, à l'intérieur d'un même quart, il est possible d'effectuer plus d'un type d'activité. Le module vise à remplir les quarts de travail avec ces différentes activités de sorte à satisfaire au mieux, pour chaque activité, la demande à chaque période de l'horizon de planification. Dans une moindre mesure, le module essaie aussi de garder le plus bas possible le nombre de transitions entre deux activités différentes. Plusieurs contraintes régissent l'affectation des activités. D'abord l'employé assigné à la couverture d'une activité doit avoir les qualifications nécessaires pour celle-ci. De plus, certaines règles imposent une limite sur la durée consécutive de certaines activités dans un quart (durée minimale et maximale). Étant donné le caractère stressant de certaines activités, une limite est aussi imposée sur la durée consécutive d'assignation à de telles activités.

La décomposition heuristique que l'on vient de présenter procure de nombreux avantages : temps de résolution accélérés, décomposition du problème selon un schéma administratif pratique, satisfaction adéquate des objectifs. Cependant, de par sa nature même, une telle décomposition mène généralement à des solutions sous-optimales. L'obtention de solutions de meilleure qualité serait possible si toutes ces étapes étaient exécutées dans un même processus d'optimisation. Une nouvelle approche de résolution allant dans ce sens a été proposée par Vatri (2001) dans le cadre de son mémoire de maîtrise. Cette approche fusionne les trois dernières étapes de la résolution heuristique du problème. Cependant, ce modèle ne tenant pas compte du placement des pauses dans les quarts, il est difficile de comparer ses performances à celles du logiciel ShiftLogic. Toutefois, les résultats obtenus avec son approche montrent qu'elle est prometteuse. Le présent mémoire a pour but d'inclure le placement des pauses dans l'approche de résolution proposée par Vatri (2001), ce qui permettra d'effectuer de véritables comparaisons avec le logiciel ShiftLogic. L'ajout des pauses dans une approche intégrée pour la résolution du PFHQ ajoute beaucoup de complexité au problème, c'est pourquoi l'approche retenue dans ce mémoire sera aussi une approche heuristique, en ce sens qu'elle ne prétend pas pouvoir fournir la solution satisfaisant au mieux les divers objectifs du problème.

Voici un bref aperçu du contenu de ce mémoire. Le premier chapitre présente une revue de la littérature concernant les divers problèmes de fabrication d'horaires avec une emphase particulière pour les problèmes d'horaires avec quarts de travail. Le deuxième chapitre présente un modèle mathématique simplifié du problème d'assignation des activités à des quarts de travail avec pauses fixées. Le troisième chapitre propose différentes stratégies permettant la résolution du problème plus général consistant à assigner des activités et des pauses à des quarts de travail existants. Le quatrième chapitre présente une façon d'intégrer les stratégies mises au point au chapitre précédent dans l'approche intégrée de résolution du PFQH. Ce mémoire se termine par une brève conclusion.

CHAPITRE 1 : REVUE DE LITTÉRATURE

Le problème auquel nous nous attardons, soit le PFHQ s'apparente à quatre autres types de problèmes rencontrés dans la littérature : la construction des cycles, la construction des quarts, la construction simultanée des cycles et des quarts et l'affectation d'activités. Le problème que nous abordons pourrait être défini comme un problème de fabrication de quart et d'affectation d'activités.

Ces problèmes peuvent prendre différentes formes selon les objectifs visés et le niveau de flexibilité permis. Par exemple, le problème de construction des quarts peut avoir pour objectif de minimiser le nombre d'employés pour satisfaire les besoins pour une période donnée, mais il peut aussi bien viser à satisfaire au maximum les besoins avec un nombre d'employés prédéterminé. De plus, certains de ces problèmes permettront une flexibilité sur la durée des quarts ou sur la période durant laquelle ils peuvent débiter. Le placement des pauses à l'intérieur des quarts peut être fixé ou permettre un éventail de configurations distinctes. Bailey et Fields (1985) ont étudié l'effet associé à l'utilisation de divers temps de début et différentes durées pour les quarts sur la diminution des coûts de main-d'oeuvre. Bechtold et Jacobs (1993) ont considéré un problème ayant une pause unique et ont déclaré que la flexibilité sur la longueur des quarts et sur le temps de début de la pause améliorerait systématiquement l'utilisation des ressources humaines pour toutes les courbes de demandes qu'ils ont étudiées. Étant donné les avantages potentiels d'un plus grand niveau de flexibilité, plusieurs recherches visent à inclure plus de flexibilité.

Un autre aspect important caractérisant ces problèmes est la nature de l'horizon de planification. Cet horizon peut être de deux types : un horizon continu où l'on trouve un temps d'opération de 24 heures par jour et un horizon non-continu où le temps

d'opération est de moins de 24 heures par jour. Dans le cas d'un horizon continu, les quarts peuvent chevaucher deux jours consécutifs. Brusco et Jacobs (1995) ont montré que les problèmes présentant un horizon continu sont difficiles à résoudre à l'optimalité. Certains de ces problèmes se sont révélés être NP-complets selon Bartholdi (1981).

1.1 Construction des cycles

La construction des cycles a pour but d'établir les jours de congé et de repos des employés pour un horizon de planification donné. On peut y trouver différents niveaux de souplesse, qui consiste généralement en un choix entre différents patrons de jour de travail. Quoiqu'un degré de souplesse plus élevé peut considérablement complexifier le problème, Mabert et Watts (1982) ont montré qu'il a souvent un impact positif sur l'utilisation du personnel. Deux approches parmi les méthodes proposées pour résoudre le problème de construction des cycles ont retenu notre attention. Burns et Carter (1985) ont proposé un algorithme itératif pouvant être utilisé sur un horizon continu. Emmons et Burns (1991) présente un algorithme glouton optimal fonctionnant sur un horizon continu et ayant la particularité d'inclure différents niveaux de qualifications.

1.2 Construction des quarts

La construction des quarts consiste à déterminer, pour chaque jour de travail des employés, le temps de début et de fin de leur quart de travail et possiblement à placer les pauses dans ces quarts. Différents niveaux de souplesse peuvent être présents, qui varieront selon les caractéristiques suivantes :

- la durée des quarts est fixe ou peut varier ;
- les temps auxquels peut débiter un quart ;
- le nombre de pauses par quart ;
- certaines pauses peuvent être décomposées ou non ;
- le début d’une pause est fixe ou il est compris à l’intérieur d’une fenêtre de temps (fenêtre de pause).

Plusieurs recherches se sont penchées sur le problème de construction des quarts. Dantzig (1954) a proposé une première formulation basée sur un modèle généralisé de recouvrement d’ensemble dans lequel chaque quart (un pour chaque combinaison de temps de début, temps de fin et de placement de pauses) est représenté explicitement par une variable de décision. Le nombre de quarts à énumérer avec ce modèle croît extrêmement rapidement lorsque la souplesse est augmentée : le problème de recouvrement devient alors très difficile à résoudre. Segal (1974), Keith (1979), Henderson et Berry (1976) ont étudié l’utilisation d’approches heuristiques pour contourner cette difficulté.

Bechtold et Jacobs (1990) ont proposé une approche différente qui consiste à réduire la taille du modèle en représentant implicitement le placement des pauses. L’algorithme proposé fonctionne sur un horizon non-continu où une pause unique non décomposable peut débiter dans une fenêtre de temps donnée. La durée de cette pause est fixe. Dans cette formulation, chaque quart (un pour chaque combinaison de temps de début et de durée de quart) est représenté par une variable indiquant le nombre de quarts de ce type devant être générés. Pour chaque période de l’horizon d’optimisation compris entre la première période et la dernière période où une pause peut débiter, le modèle associe une variable indiquant le nombre de pauses commençant à cette période. Les variables de quarts et de pauses sont liées par des contraintes de type *en avant* et *en arrière* demandant qu’il y ait un nombre suffisant de pauses avant et après un ensemble de temps précis. Lorsque la flexibilité

sur les pauses est assez grande, le modèle de Bechtold et Jacob permet de diminuer le nombre de variables requises par rapport au modèle généralisé de recouvrement d'ensemble présenté par Dantzig (1954). Le modèle mathématique est de type programmation linéaire en nombres entiers (PLNE) et est résolu à l'aide d'un algorithme de recherche de solutions entières avec plans de coupe. Leur algorithme permet de trouver, en 20 minutes, la solution optimale pour des instances ayant entre 5 et 45 employés, un horizon de 12 heures (la période de discrétisation est de 30 minutes) et 970 types de quarts. Bechtold et Jacobs (1996) ont plus tard prouvé que leur modèle est équivalent au modèle généralisé de recouvrement d'ensemble à la condition qu'il n'y ait pas de chevauchement extraordinaire (i.e. il n'y a pas deux quarts tels que la fenêtre de pause de l'un commence strictement avant et se termine strictement après la fenêtre de pause de l'autre).

Moondra (1976) a utilisé une modélisation où la flexibilité sur la longueur des quarts et les différents temps de début des quarts est représentée implicitement. Il a considéré des quarts de temps-plein de durée fixe contenant une pause pouvant débuter à l'intérieur d'une fenêtre de temps de deux périodes et des quarts de temps-partiel de longueur flexible ne contenant pas de pause. La flexibilité sur la durée des quarts de temps-partiel est modélisée à l'aide de variables définissant les temps de début et de fin de quart et un ensemble de contraintes qui force le respect des durées maximum et minimum de ces quarts. Afin de tenir compte de la flexibilité lors du placement des pauses, la moitié des employés à temps-plein sont forcés de prendre leur pause lors de la première période de la fenêtre de pause et les autres à la seconde période.

Thompson (1995) a présenté un modèle hybride reprenant la représentation implicite des quarts du modèle de Moondra (1976) et s'inspire du modèle de Bechtold et Jacobs (1990) pour représenter implicitement le placement des pauses dans le contexte d'une planification sur un horizon continu. Chaque quart reçoit au plus une pause sujette à certaines restrictions concernant les durées de travail survenant avant et après celle-ci. Les quarts ayant les mêmes coûts par période de travail, les mêmes possibilités

de durée, les mêmes durées de pause et les mêmes restrictions sur le placement de la pause sont regroupés dans un même type de quart. Pour chaque type de quart, des variables sont définies pour représenter le temps de début et de fin du quart et le temps de début de la pause. Un ensemble de contraintes permet de lier ces différentes variables afin d'assurer la conformité des quarts. Le modèle mathématique présenté par l'auteur est de type PLNE et est résolu par un algorithme exact de séparation et évaluation progressive. L'auteur a obtenu des solutions en 7 minutes sur les mêmes problèmes que Bechtold et Jacobs (1990).

Aykin (1996) a présenté une approche permettant de résoudre le problème de construction de quarts. Le modèle qu'il propose permet de planifier plusieurs types de pauses par quart, chacune pouvant débuter dans une fenêtre de temps pré-déterminée. Le placement des pauses est représenté en introduisant une variable pour chaque quart et chaque période où peut commencer une pause. Un quart étant défini comme la combinaison d'un temps de début, une durée, les types de pauses qu'il contient et les fenêtres qui leur sont associées. Contrairement à la modélisation présentée par Bechtold et Jacobs (1990), ce modèle permet plusieurs types de pauses et le chevauchement extraordinaire. Le pairing entre les quarts et les pauses pouvant lui être associées est obtenu par un ensemble de contraintes d'égalité, une pour chaque type de pause et chaque quart. Aykin (1998) propose un algorithme résolvant la formulation de Aykin (1996) à l'aide d'un algorithme de type séparation et évaluation progressive combiné à des plans de coupe. Aykin (2000) a montré que, pour une certaine famille de problèmes, sa formulation obtient plus rapidement des solutions optimales que la formulation de Bechtold et Jacobs (1990).

Rekik *et al.* (2003) ont présenté deux approches de modélisation implicite pour le problème de construction des quarts sur un horizon continu. Le problème qu'ils ont considéré comprend plusieurs pauses et des fenêtres de pauses avec restriction sur la durée des périodes de travail se trouvant avant et après les pauses. De plus, certaines

pauses peuvent être décomposables. La première approche qu'ils ont présentée est une extension de la formulation d'Aykin (1996) alors que la deuxième est une extension des travaux de Bechtold et Jacobs (1990). Dans leur première formulation, le modèle implicite proposé est essentiellement basé sur l'utilisation de variables de pauses associées aux quarts. Le lien entre les pauses et les quarts est formulé comme un problème de transport. Dans la deuxième formulation, ils ont étendu le modèle de Bechtold et Jacobs (1990) afin qu'il fonctionne sur les problèmes qu'ils ont considérés. Pour ce faire, ils ont modifié leur première formulation afin de pouvoir la séparer en un ensemble indépendant de problèmes de transport, lesquels sont remplacés par des contraintes *en avant* et *en arrière*. Baba-Hadji (2004) a fait une étude comparative des modèles de Dantzig (1954) et de ceux proposés par Rekik *et al.* (2003a). Il a, de plus, proposé une méthode de décomposition temporelle pour résoudre des problèmes sur un horizon d'une semaine.

1.3 Construction des cycles et des quarts

Comme son nom l'indique, la construction des cycles et des quarts de travail consiste à résoudre à la fois le problème de construction des quarts et de construction des cycles. Les possibilités de flexibilité présentes dans ces deux problèmes peuvent aussi se retrouver dans le problème de construction des cycles et des quarts.

Burns et Koop (1987) ont proposé un algorithme exact pour un horizon continu. Le modèle qu'ils ont présenté ne permet pas la planification des pauses et aucune flexibilité n'est possible quant aux heures de début et aux durées des quarts. Leur algorithme est simple mais son champ d'application est limité.

Easton et Rossin (1991) ont présenté un algorithme heuristique utilisant une approche par génération de colonnes. La méthode fonctionne sur un horizon non-continu, elle

ne permet pas la planification des pauses et ne considère pas les patrons de quarts (un patron de quarts est une séquence, de longueur variable, d'alternatives de types de quarts que les employés désirent travailler). Elle permet par contre une flexibilité sur les heures de début de quarts. Une heuristique réduit la taille du problème en choisissant un sous-ensemble d'horaires de travail parmi tous les horaires réalisables. Les résultats obtenus sont supérieurs à ceux obtenus avec d'autres méthodes utilisant un sous-ensemble prédéterminé d'horaires possibles.

Jarrah *et al.* (1994) ont présenté un algorithme heuristique dans le contexte d'un horizon non-continu. Le modèle ne considère pas les patrons de quarts. Cependant il prend en charge la planification d'une pause à l'intérieur d'une fenêtre de temps, il considère divers types de quarts et permet une flexibilité sur les heures de début et les durées des quarts. L'algorithme présenté permet d'obtenir des solutions de bonne qualité en des temps raisonnables.

Brusco et Jacobs (1998) ont étudié le cas où il y a des contraintes qui imposent un délai minimal entre les heures successives de début de quarts. Les auteurs ont divisé le problème en deux parties : la détermination des heures de début des quarts et l'assignation des cycles aux employés. Quoique l'algorithme présenté soit assez rapide, il ne trouve pas de solution pour certaines instances du problème considéré.

Haase (1999) a proposé un algorithme exact pour un problème sur un horizon continu. La méthode permet de prendre en considération la planification de plusieurs pauses et l'utilisation de fenêtres de temps pour les pauses ainsi que la flexibilité sur la durée et l'heure de début des quarts. L'auteur propose un modèle de recouvrement généralisé qui se résout par une méthode de génération de colonnes couplée à un algorithme exact de séparation et évaluation progressive.

Rekik *et al.* (2002) ont présenté une approche généralisant l'utilisation des contraintes *en avant* et *en arrière* introduite par Bechtold et Jacobs (1990) permettant ainsi de l'appliquer au problème de construction des cycles et des quarts.

1.4 Affectation d'activités à l'intérieur des quarts

Omari (2002) a proposé un algorithme heuristique permettant de résoudre le problème d'affectation d'activités à l'intérieur des quarts sur un horizon continu pour les contrôleurs aériens. Il s'agit ici d'un contexte multi-activités. Partant de quarts de travail construits et déjà affectés aux employés, la méthode présentée par l'auteur remplit les quarts d'affectation à des activités. Le modèle tient compte des compétences des employés lors de l'affectation des activités. L'objectif du problème considéré vise à minimiser la somme des sous-couvertures des demandes pour chaque activité et chaque période de l'horizon de planification. Les problèmes doivent respecter de nombreuses règles issues de la convention collective des employés. La technique de résolution est basée sur une méthode de génération de colonnes combinée à un algorithme heuristique de séparation et évaluation progressive. Les jeux de données utilisés comprennent un horizon d'un mois, 85 employés, une vingtaine d'activités ainsi qu'une discrétisation de l'horizon de planification en période de 15 minutes. Les résultats obtenus par l'auteur présente une diminution de la somme des sous-couvertures par rapport aux solutions obtenues à l'aide des méthodes précédemment utilisées pour ce problème spécifique.

1.5 Fabrication d'horaires avec quarts de travail

La fabrication d'horaires avec quarts de travail consiste à combiner le problème de fabrication de quarts et le problème d'affectation des activités à l'intérieur des quarts. Les problèmes traités dans le présent mémoire sont de ce type. Vatri (2001) a présenté une extension des travaux de Omari (2002) permettant la résolution du PFHQ. En effet, il reprend le modèle de Omari (2002) pour l'affectation des activités et y ajoute des composantes de construction de quarts et d'affectation des quarts aux employés.

Les problèmes étudiés sont les mêmes que ceux de Omari (2002) et se situent donc dans un contexte multi-activités sur un horizon continu. Le modèle présenté considère les types de quarts et permet une flexibilité sur la durée et le début des quarts. De plus, des contraintes permettent de respecter des règles régissant le nombre de quarts de chaque type à générer. Les jours de travail et les patrons de quarts des employés sont pré-assignés. Le modèle présenté ne permet toutefois pas la planification des pauses à l'intérieur des quarts de travail. L'auteur utilise une méthode heuristique de séparation et évaluation progressive intégrant une approche de génération de colonnes combinée à une stratégie de décomposition temporelle. Les résultats présentés montrent que la méthode permet d'obtenir des solutions pour PFHQ légèrement meilleures que celles obtenues avec les méthodes précédentes utilisant des quarts pré-établis.

1.6 Contributions et justifications

La principale contribution de ce mémoire est le développement de stratégies d'accélération dans le contexte de la génération de colonnes afin de permettre la génération des pauses à l'intérieur des quarts. La deuxième est de démontrer que l'intégration des trois étapes (construction des quarts de travail avec pauses, assignation des quarts aux employés et assignation des activités) permet d'obtenir des solutions de meilleure qualité. Finalement, l'analyse des résultats indique que l'approche proposée peut résoudre ces problèmes en des temps acceptables. Les modèles présentés ont été conçus dans l'optique de les utiliser avec une technique de résolution intégrant une méthode de génération de colonnes combinée à une stratégie de décomposition temporelle pour deux raisons. La première raison est que la taille de l'ensemble des quarts réalisables est si grand que le problème de couverture d'ensemble généralisé serait presque impossible à résoudre. La deuxième raison est que certaines contraintes

comme la qualification pour les activités et la prise en charge d'activités stressantes, ne peuvent être modélisées implicitement. On ne peut donc pas utiliser de modèle implicite pour réduire la taille du problème.

CHAPITRE 2 : LE PROBLÈME D'ATTRIBUTION DES ACTIVITÉS AVEC PAUSES FIXÉES

Cette section présente un survol du modèle et de la méthode utilisés par Omari (2002) pour la résolution du problème d'attribution des activités pour les contrôleurs aériens. Dans ce contexte, un ensemble de pauses respectant les contraintes auxquelles elles sont assujetties est donné. Ce problème accepte telles quelles les pauses reçues et ne cherche donc pas à les optimiser. Nous l'appellerons donc : *problème d'attribution des activités avec pauses fixées*. Afin de ne pas alourdir le texte, nous présenterons, en premier lieu, un modèle de base associé à une version simplifiée de ce problème ainsi que les techniques de résolution qui lui sont associées. Nous présenterons ensuite brièvement les extensions qui doivent être ajoutées pour satisfaire le problème en entier.

2.1 Version simplifiée du problème

Le problème est défini pour un horizon de planification donné, par exemple un mois. Cet horizon est discrétisé en périodes de même durée, typiquement 15 minutes. Un ensemble d'activités est donné et, pour chacune d'entre elles, une demande est définie pour chaque période de l'horizon de planification. Un ensemble de quarts est fourni et pour chaque quart de cet ensemble on connaît son heure de début et de fin, les heures de début et de fin des pauses qui s'y trouvent et l'employé qui y travaille. Le problème consiste à déterminer, pour chaque quart et pour chaque période du quart où il n'y a pas de pause, l'activité à affectuer lors de cette période. L'affectation d'une activité

à un employé doit respecter deux contraintes : l'employé doit avoir les qualifications requises et la durée consécutive d'affectation à l'activité doit respecter une valeur minimale et maximale. On définit une sous-couverture comme étant l'absence d'un employé pour couvrir une activité sur une durée d'une période. On appelle transition le passage pour un même employé d'une activité (ou pause) à une autre activité (ou pause). L'objectif principal du problème est de couvrir au mieux la demande (i.e. minimiser le nombre de sous-couvertures). Un objectif secondaire est de minimiser le nombre de transitions.

2.2 Modèle de base

Le modèle de base proposé dans cette section est valide pour la version simplifiée du problème d'attribution des activités avec pauses fixées présentée à la sous-section précédente. En se restreignant à ces objectifs et contraintes, il est possible de formuler le problème d'attribution des activités avec pauses fixées à l'aide d'un modèle pouvant être résolu par une approche de génération de colonnes. La notation nécessaire pour ce modèle et le modèle lui-même sont présentés dans la première sous-section. Comme l'approche de génération de colonnes fait appel à des sous-problèmes, nous présentons leur formulation à la deuxième sous-section.

2.2.1 Formulation mathématique

Pour bien comprendre la formulation proposée, il faut faire l'hypothèse que nous possédons une énumération de tous les horaires réalisables pour chaque quart de travail. Le problème consiste à trouver parmi ces horaires réalisables, la combinaison d'horaires ayant le plus petit coût total. Définissons d'abord la notation suivante qui sera utile à l'élaboration de la formulation mathématique :

- H , l'intervalle de temps pour lequel on cherche à réaliser un horaire de travail. On nomme cet intervalle de temps l'*horizon d'optimisation*.
- K , l'ensemble des quarts de travail ayant une intersection avec l'horizon d'optimisation H .
- P , l'ensemble des périodes de l'horizon d'optimisation H ; $p \in P$ est une période. Les éléments de cet ensemble sont numérotés de 1 à $|P|$.
- δ , la durée d'une période (les périodes ont toutes la même durée).
- Ω^k , l'ensemble des horaires pour le quart $k \in K$.
- c_w^k , le coût de l'horaire $w \in \Omega^k$.
- n_{ap} , la demande en employés pour l'activité $a \in A$ pendant la période $p \in P$.
- $P_a = \{p \in P \mid n_{ap} > 0\}$, l'ensemble des périodes où la demande de l'activité $a \in A$ est non nulle.
- c_{ap} , le coût correspondant à la pénalité pour une unité de sous-couverture de l'activité $a \in A$ durant la période $p \in P_a$.
- $g_{ap,w}^k$, le coefficient binaire qui vaut 1 si l'horaire $w \in \Omega^k$ couvre l'activité $a \in A$ durant la période $p \in P_a$, et 0 sinon.

Le modèle fait appel à deux groupes de variables. Le premier groupe contient les variables binaires Θ_w^k , $k \in K$, $w \in \Omega^k$, qui valent 1 si l'horaire w est retenu pour le quart k , et 0 sinon. Les variables d'écart Y_{ap} du deuxième groupe permettent de violer les contraintes de couverture moyennant une pénalité. La formulation mathématique proposée est la suivante :

$$\min \sum_{k \in K} \sum_{w \in \Omega^k} c_w^k \Theta_w^k + \sum_{a \in A} \sum_{p \in P_a} c_{ap} Y_{ap} \quad (2.1)$$

sujet à :

$$\sum_{k \in K} \sum_{w \in \Omega^k} g_{ap,w} \Theta_w^k + Y_{ap} \geq n_{ap}, \quad \forall a \in A, p \in P_a \quad (2.2)$$

$$\sum_{w \in \Omega^k} \Theta_w^k = 1, \quad \forall k \in K \quad (2.3)$$

$$\Theta_w^k \in \{0, 1\}, \quad \forall k \in K, w \in \Omega^k. \quad (2.4)$$

L'expression (2.1) est la fonction objectif qui cherche à minimiser la somme des coûts des horaires choisis et les pénalités associées aux variables d'écart Y_{ap} utilisées dans les contraintes (2.2). Ces dernières favorisent la couverture des demandes en employés pour l'ensemble des activités $a \in A$ pour chacune des période $p \in P_a$. Les variables d'écart Y_{ap} permettent de sous-couvrir la demande pour l'activité $a \in A$ à la période $p \in P_a$. Cependant, dans ce cas, la fonction objectif se voit pénalisée d'une quantité c_{ap} pour chaque unité de sous-couverture. Les contraintes (2.3) conjointement avec les contraintes d'intégrité (2.4) assurent qu'un seul horaire réalisable soit choisi dans la solution finale, pour chaque quart $k \in K$.

2.2.2 Les sous-problèmes

Tel que présenté à la section 2.3, le modèle (2.1)-(2.4) se résout à l'aide d'une approche de génération de colonnes imbriquée dans une méthode de séparation et d'évaluation progressive. La technique de génération de colonnes divise un programme mathématique en deux composantes distinctes, soit un problème maître, dans ce cas-ci la relaxation linéaire de (2.1)-(2.4), et un ensemble de sous-problèmes. Dans le cas du problème d'attribution des activités avec pauses fixées, il y a un sous-problème pour chaque quart de travail $k \in K$ chevauchant l'horizon d'optimisation. Ce sous-problème est un problème de plus court chemin avec contraintes de ressources défini sur un réseau espace-temps G^k .

Cette sous-section présente la formulation d'un tel sous-problème. Nous verrons

d'abord la structure du réseau G^k , pour ensuite présenter les ressources requises par ce problème et, finalement, donner la formulation du sous-problème.

Définition du réseau

La notation suivante sera nécessaire pour la définition du réseau :

- P^k , l'ensemble des périodes du quart $k \in K$ qui sont comprises dans l'horizon d'optimisation H .
- T , l'ensemble des temps qui délimitent les périodes comprises dans l'horizon d'optimisation H . Les éléments de cet ensemble sont numérotés de 1 à $|T|$.
- T^k , l'ensemble des temps qui délimitent les périodes de l'ensemble P^k .
- A , l'ensemble des activités. Les activités sont numérotées de 1 à $|A|$.
- A^k , l'ensemble des activités pour lesquelles l'employé associé au quart $k \in K$ est qualifié.
- B^k , l'ensemble des pauses fixées pour le quart $k \in K$ dont les débuts sont compris dans l'horizon d'optimisation H .
- $d_a^{k,min}$, la durée minimale d'affectation à l'activité $a \in A^k$ pour le quart $k \in K$.
- $Pause_b^k$, l'intervalle $[t, t']$ où $t \in T^k$ et $t' \in T^k$ durant lequel la pause $b \in B^k$ est fixée.
- $Deb_a^k = \{t \in T^k \mid \forall b \in B^k, Pause_b^k \cap [t, t + d_a^{k,min}] = \emptyset\}$, l'ensemble des temps valides de début d'affectation à l'activité $a \in A^k$ pour le quart $k \in K$.
- c_{ij}^k , le coût d'utilisation de l'arc $(i, j) \in E^k$.

Le sous-problème associé au quart $k \in K$ se définit sur un réseau espace-temps $G^k = (V^k, E^k)$ (voir Figure 2.1). L'ensemble V^k comprend cinq types de noeuds : $nOri$, $nDes$, $nAct$, $nPau$ et $nTrs$. Ces types permettent de définir le contenu de V^k comme suit :

- Un noeud de type $nOri$, noté $o(k)$, qui correspond au début du quart k .
- Un noeud de type $nDes$, noté $d(k)$, qui correspond à la fin du quart k .
- Pour chaque activité $a \in A^k$ et chaque temps $t \in T^k$, un noeud de type $nAct$, noté $(nAct, a, t)$, représentant l'activité a au temps t .
- Pour chaque temps $t \in T^k$, un noeud de type $nPau$, noté $(nPau, t)$, représentant une pause au temps t .
- Pour chaque temps $t \in T^k$, un noeud de type $nTrs$, noté $(nTrs, t)$, représentant une transition, au temps $t \in T^k$, entre deux activités ou entre une activité et une pause.

Les éléments de E^k se répartissent selon six types d'arcs : $aDeb$, $aFin$, $aDebA$, $aAct$, $aPau$ et $aTrs$. On peut définir l'ensemble E^k de la façon suivante :

- Un arc de type $aDeb$ relie le noeud $o(k)$ au premier noeud de type $nTrs$ pour représenter le début du quart de travail k dans l'horizon d'optimisation H .
- Un arc de type $aFin$ relie le dernier noeud $nTrs$ au noeud $d(k)$ pour représenter la fin du quart de travail k dans l'horizon d'optimisation H .
- Pour chaque activité $a \in A^k$ et chaque temps $t \in Deb_a^k$, un arc de type $aDebA$ relie le noeud $(nTrs, t)$ et le noeud $(nAct, a, t + d_a^{k,min})$. Cet arc représente l'affectation à l'activité a au temps t pour une durée $d_a^{k,min}$, assurant ainsi le respect de la contrainte de durée minimale d'affectation.
- Pour chaque activité $a \in A^k$ et chaque temps $t \in T^K$: si $\forall b \in B^k, [t, t + \delta] \cap Pause_b^k = \emptyset$, alors un arc de type $aAct$ relie $(nAct, a, t)$ et $(nAct, a, t + \delta)$. Cet arc représente la prolongation de l'activité a au temps t pour une durée δ si aucune pause n'est prévue dans cet intervalle de temps.
- Pour chaque pause $b \in B^k$, un arc de type $aPau$ relie le noeud $(nTrs, t)$ et le noeud $(nPau, t')$, où t est le début de l'intervalle $Pause_b^k$ et t' est la fin de cet intervalle. On fixe ainsi la pause car cet arc est le seul à faire le passage entre le temps t et le temps t' dans le réseau G^k .

- Pour chaque noeud de type $nAct$ ou $nPau$, il y a un arc de type $aTrs$ qui part de ce noeud pour se rendre au noeud $nTrs$ ayant le même temps. Ces arcs permettent d'effectuer une transition d'une activité à une autre ou d'une activité à une pause.

Tous les arcs ont un coût nul sauf les arcs $aTrs$ qui ont un coût unitaire de γ^t correspondant à la pénalité associée à une transition. Ce réseau permet d'énumérer tous les horaires réalisables pour le quart $k \in K$. Toutefois, il permet aussi d'énumérer des horaires non-réalisables. L'utilisation de ressources permet de restreindre l'énumération à l'ensemble des horaires réalisables.

Ressources

Certaines contraintes s'appliquant individuellement à chaque quart de travail peuvent être modélisées à l'aide de ressources qui sont gérées au niveau des réseaux (voir Desrosiers *et al.*, 1995). Il s'agit d'une valeur scalaire qui varie selon le chemin utilisé pour passer de $o(k)$ à $d(k)$. Chaque arc consomme ou produit, selon l'état du réseau avant de traverser l'arc, une certaine quantité de la ressource, cette quantité étant définie par la fonction de prolongation. Chaque ressource doit satisfaire un ensemble de conditions, lesquelles sont définies en général à l'aide de fenêtres de ressource à chacun des noeuds. Plusieurs des contraintes que nous aurons à traiter seront prises en charge à l'aide de telles ressources.

Soit R l'ensemble des ressources utilisées. La valeur prise par la ressource $r \in R$ au noeud $i \in N^k$, $k \in K$, est exprimée par la variable Φ_i^{kr} , la fonction de prolongation de r le long de l'arc $(i, j) \in E^k$ par $f_{ij}^r(\Phi_i^{kr})$ et la fenêtre de cette ressource au noeud $i \in N^k$ par $[l_i^r, u_i^r]$. De plus, on définit, pour chaque ressource $r \in R$ et chaque arc $(i, j) \in E^k$, la quantité φ_{ij}^{kr} qui est un intrant de la fonction de prolongation $f_{ij}^r(\Phi_i^{kr})$. Le lien entre la valeur de la ressource à un noeud $i \in N^k$ donné et la fenêtre de

ressource à ce même noeud est exprimé par la relation $l_i^r \leq \Phi_i^{kr} \leq u_i^r$. Pour respecter cette relation, le traitement suivant est effectué après le passage sur l'arc $(i, j) \in E^k$:

$$T_j^{kr} = \begin{cases} f_{ij}^r(\Phi_i^{kr}), & \text{si } l_i^r \leq f_{ij}^r(\Phi_i^{kr}) \leq u_i^r \\ l_i^r, & \text{si } f_{ij}^r(\Phi_i^{kr}) < l_i^r \\ \emptyset, & \text{si } f_{ij}^r(\Phi_i^{kr}) > u_i^r \text{ (la visite du noeud n'est pas admissible).} \end{cases}$$

Une notion importante pour l'utilisation des ressources est celle de dominance. Lors de la résolution du problème de plus court chemin utilisant des ressources, chaque noeud déjà parcouru possède un ensemble d'étiquettes décrivant les caractéristiques des chemins partant du noeud source et allant à ce noeud. Ces étiquettes contiennent le coût du chemin, et les valeurs Φ_i^{kr} des différentes ressources. Il y aura dominance entre deux étiquette en un noeud quand le chemin associé à l'une d'elle est en tous points inférieur à celui associé à l'autre (i.e. son coût et la valeur de chaque ressource est plus élevé). Les étiquettes dominées sont enlevé de l'ensemble d'étiquettes du noeud.

Deux ressources, $rDuree$ et $rNum$, sont nécessaires pour exprimer le modèle de base. Elles servent toutes deux à assurer le respect de la contrainte sur la durée maximale d'affectation à une activité. La première cumule la durée d'affectation consécutive à une activité tandis que la deuxième assure que chaque transition soit faite entre des activités distinctes.

Les conditions imposées à la ressource $rNum$ ne sont pas de type fenêtres de ressource sur les noeuds. Elles sont plutôt définies comme un droit de regard sur le passage sur un arc. Plus précisément, on peut emprunter l'arc $(i, j) \in E^k$ si on respecte la condition suivante :

$$(\Phi_i^{k,rNum} = \varphi_{ij}^{k,rNum}) \vee (\Phi_i^{k,rNum} = 0) \vee (\varphi_{ij}^{k,rNum} = 0). \quad (2.5)$$

La valeur $\Phi_i^{k,rNum}$ correspond au numéro de l'activité $a \in A$ en cours avant d'avoir visité le noeud $i \in V^k$, ce numéro est noté num_a et est strictement supérieur à 0.

Tableau 2.1 – Fonctions de prolongation de la ressource $rNum$.

Type d'arc	$\varphi_{ij}^{k,rNum}$	$f_{ij}^r(\Phi_i^{k,rNum})$
$aDeb$	0	0
$aFin$	0	$\Phi_i^{k,rNum}$
$aDebA$	num_a	$\begin{cases} \varphi_{ij}^{k,rNum} & \text{si } \Phi_i^{k,rNum} \neq \varphi_{ij}^{k,rNum} \\ \emptyset & \text{sinon} \end{cases}$
$aAct$	0	$\Phi_i^{k,rNum}$
$aPau$	0	0
$aTrs$	0	$\Phi_i^{k,rNum}$

$\Phi_i^{k,rNum}$ prend cependant la valeur 0 si on est au début du quart $k \in K$ ou si on revient d'une pause. Les valeurs de φ_{ij}^{kr} et les fonctions de prolongation $f_{ij}^{k,rNum}$ sont données au tableau 2.1.

Par contre, les conditions imposées à la ressource $rDuree$ sont exprimées comme des fenêtres de ressource sur certains noeuds du réseau. En fait, ces conditions ne sont imposées que sur les noeuds de type $nAct$ associés à une activité $a \in A$ (i.e., pour les autres noeuds, la fenêtre est $[0, +\infty]$). Pour un noeud $(nAct, a, t)$ donné, la valeur de cet intervalle est $[0, d_a^{k,max}]$ où $d_a^{k,max}$ est la durée maximale d'affectation à l'activité a pour le quart k . Les valeurs de φ_{ij}^{kr} et les fonctions de prolongation $f_{ij}^{k,rNum}$ sont données au tableau 2.2.

Formulation des sous-problèmes

Tel que mentionné plus haut, un sous-problème correspondant à un problème de plus court chemin dans le réseau G^k est associé à chaque quart $k \in K$. Tel que nous le verrons à la section 2.3, l'objectif de ce sous-problème consiste à trouver, pour le quart k , l'horaire réalisable $w \in \Omega^k$ qui a le plus petit coût réduit étant donné

Tableau 2.2 – Fonctions de prolongation de la ressource $rDuree$.

Type d'arc	$\varphi_{ij}^{k,rDuree}$	$f_{ij}^r(\Phi_i^{k,rDuree})$
$aDeb$	0	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$
$aFin$	0	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$
$aDebA$	d_{ij}	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$
$aAct$	d_{ij}	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$
$aPau$	0	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$
$aTrs$	$\begin{cases} -d_a^{k,max} & \text{si } i \text{ est de type nAct} \\ 0 & \text{sinon} \end{cases}$	$\Phi_i^{k,rDuree} + \varphi_{ij}^{k,rDuree}$

une solution duale du problème (2.1)-(2.4). La notation suivante sera utile pour la formulation :

- $g_{ap,ij}^k$, le coefficient binaire qui vaut 1 si le passage sur l'arc $(i,j) \in E^k$ correspond à une assignation à l'activité $a \in A$ durant la période $p \in P_a$, et 0 sinon.
- α_{ap} , la variable duale associée à la contrainte (2.2) pour l'activité $a \in A$ et la période $p \in P_a$.
- μ_k , la variable duale associée à la contrainte (2.3) pour le quart $k \in K$.

Le problème de plus court chemin s'exprime à l'aide de variables de décision binaires X_{ij}^k indiquant le flot sur les arcs $(i,j) \in E^k$, $k \in K$. Les variables $T_i^{k,r}$ comptabilisent l'état de la ressource $r \in R$ au noeud $i \in N^k$, $k \in K$. La formulation du sous-problème associé au quart $k \in K$ est :

$$\min \sum_{(i,j) \in E^k} (c_{ij}^k - \sum_{p \in P} \sum_{a \in A_p} g_{ap,ij}^k \alpha_{ap}) X_{ij}^k - \mu^k \quad (2.6)$$

sujet à :

$$\sum_{j:(o(k),j) \in E^k} X_{o(k),j}^k = 1 \quad (2.7)$$

$$\sum_{j:(i,j) \in E^k} X_{ij}^k - \sum_{j:(j,i) \in E^k} X_{ji}^k = 0, \quad \forall i \in N^k \quad (2.8)$$

$$\sum_{j:(j,d(k)) \in E^k} X_{j,d(k)}^k = 1 \quad (2.9)$$

$$X_{ij}^k (f_{ij}^{kr}(T_i^{kr}) - T_j^{kr}) \leq 0, \quad \forall r \in R, (i, j) \in E^k \quad (2.10)$$

$$l_i^r \leq T_i^{kr} \leq u_i^r, \quad \forall r \in R, i \in V^k \quad (2.11)$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in E^k. \quad (2.12)$$

Les contraintes (2.7)-(2.9) imposent que la solution du sous-problème corresponde à un chemin entre $o(k)$ et $d(k)$ dans le réseau G^k . Les contraintes (2.10) assurent que la consommation de chaque ressource $r \in R$ sur l'arc $(i, j) \in A^k$ soit définie par la fonction de prolongation $f_{ij}^{k,r}$ telle que précisée plus haut. Les contraintes (2.11) imposent à la ressource $r \in R$, au noeud $i \in V^k$, la fenêtre de temps $[l_i^r, u_i^r]$. Les contraintes (2.12) gardent entiers les flots sur les arcs du réseau G^k . La fonction objectif (2.6) permet de trouver le chemin réalisable entre $o(k)$ et $d(k)$ de plus petit coût réduit par rapport à la solution duale donnée.

2.3 Méthode de résolution

Cette section décrit l'approche de résolution retenue par Omari (2002) pour résoudre le problème d'attribution des activités avec pauses fixées. Les instances de ce problème rencontrées en pratique sont en général de très grande taille (200 quarts, 50 activités, ≈ 3000 périodes de 15 minutes pour un horizon d'un mois). Pour diminuer la taille des problèmes, une stratégie de décomposition temporelle a été utilisée pour diviser l'horizon d'optimisation en plusieurs tranches de temps se chevauchant. On obtient

ainsi un problème d’attribution des activités avec pauses fixées de taille réduite pour chaque tranche de temps. Chacun de ces problèmes est résolu à l’aide d’une technique de séparation et évaluation progressive (mieux connue sous le nom de “branch-and-bound”) utilisant une approche par génération de colonnes pour calculer la borne inférieure associée à chaque noeud de l’arbre de branchement (voir Desaulniers *et al.*, 1998). Cette section présentera d’abord la stratégie de décomposition temporelle. Elle aborde ensuite la recherche de bornes inférieures par génération de colonnes avant de discuter de la technique de séparation et évaluation progressive.

2.3.1 Stratégie de décomposition temporelle

Le but de la stratégie de décomposition temporelle est de diviser l’horizon d’optimisation en plusieurs tranches de temps non disjointes. Soit n le nombre de tranches de temps. Pour tout $i = 2, \dots, n - 1$, la tranche de temps d’indice i chevauche les tranches de temps $i - 1$ et $i + 1$. Une décomposition temporelle est donc complètement définie à l’aide de deux paramètres : l , la longueur d’une tranche de temps et c , la durée de chevauchement entre deux tranches de temps consécutives. Par exemple, une décomposition temporelle ayant les paramètres $l = 480$ et $c = 120$ appliquée à un horizon d’optimisation s’étendant sur une journée (1440 minutes) définit les tranches de temps $[0, 480]$, $[360, 840]$, $[720, 1200]$ et $[1080, 1440]$.

À la fin de la résolution du problème restreint à une tranche de temps d’indice $i \in \{2 \dots n\}$, la solution du problème est combinée avec la solution cumulative obtenue à la fin de l’itération $i - 1$ dans le but de construire la solution cumulative de l’itération i . Pour que la solution cumulative soit réalisable, la résolution du problème associé à une tranche de temps commençant au temps $t \in T$ doit prendre en considération les conditions initiales des quarts qu’elle chevauche et dont le début est antérieur à t . Pour ce faire, le réseau modélisant le problème associé à cette tranche de temps

est élargi pour y inclure un chemin qui correspond aux conditions initiales. Cette extension est décrite à la section 2.4.10.

La stratégie de décomposition temporelle permet de réduire considérablement les temps de résolution en considérant une série de problèmes de taille raisonnable. Le chevauchement entre deux tranches de temps consécutives permet d'améliorer la qualité de la solution en reconsidérant la fin de la première tranche de temps à la lumière de ce qui s'en vient dans la deuxième tranche de temps. Il est à noter que ces problèmes peuvent aussi se formuler à l'aide du modèle (2.1)-(2.4).

2.3.2 Calcul d'une borne inférieure par génération de colonnes

Lors de la présentation de la formulation mathématique à la section 2.2.1, on a supposé qu'on connaissait Ω^k l'ensemble des horaires réalisables pour le quart k . En pratique, il est très coûteux, voir impossible d'avoir cet ensemble dans son entier. La technique de génération de colonnes permet néanmoins de résoudre une relaxation du problème (2.1)-(2.4), appelée problème maître. Il s'agit d'une méthode itérative qui fait appel, à chaque itération, à un sous-ensemble $\bar{\Omega}^k$ de Ω^k pour chaque quart $k \in K$. Ces sous-ensembles donnent naissance à une version restreinte du problème maître. Chaque itération comporte deux étapes. La première consiste à résoudre le problème maître restreint et de construire la fonction objectif des sous-problèmes à l'aide du vecteur dual associé à la solution optimale. La deuxième étape consiste à résoudre les sous-problèmes à l'aide d'un algorithme de programmation dynamique (Desrochers et Soumis, 1988). Pour chaque quart $k \in K$, la solution du sous-problème est l'horaire $w \in \Omega^k$ de plus petit coût réduit. Dans le cas où les valeurs optimales de tous les sous-problèmes sont supérieures ou égales à zéro, la solution du problème maître restreint est aussi optimale pour le problème maître. Dans le cas contraire,

pour chaque quart $k \in K$ dont la valeur optimale du sous-problème correspondant est strictement négative, on ajoute l'horaire w associé à la solution optimale à l'ensemble $\bar{\Omega}^k$ pour obtenir un nouveau problème maître restreint. On recommence alors avec une nouvelle itération.

En pratique, cette technique peut mettre beaucoup de temps avant de converger, c'est pourquoi on ajoute souvent d'autres critères d'arrêt. Par exemple, on peut mettre fin à la procédure lorsque la fonction objectif du problème maître restreint n'a pas diminué d'une certaine valeur au cours d'un nombre défini d'itérations consécutives. Cependant, ces nouveaux critères d'arrêt ne permettent pas de résoudre le problème jusqu'à l'optimalité. Pour chaque quart $k \in K$, on peut ajouter à l'ensemble $\bar{\Omega}^k$, en plus de l'horaire ayant le plus petit coût réduit, un certain nombre d'horaires de coûts réduits négatifs. Enfin, notons aussi qu'il n'est pas nécessaire de résoudre tous les sous-problèmes si on a déjà généré un nombre considéré suffisant de nouveaux horaires à ajouter à l'ensemble $\bar{\Omega}^k$. Nous avons appliqué conjointement ces différentes modifications à la méthode de génération de colonnes ce qui a permis de réduire substantiellement les temps de résolution des problèmes de test.

2.3.3 Recherche d'une solution entière

Pour obtenir une solution entière, on a recours à une méthode d'évaluation et de séparation progressive utilisant une stratégie de branchement relativement simple. Étant donné une solution fractionnaire pour le problème maître restreint, elle consiste à fixer à 1 un certain nombre de variables Θ_w^k qui ont une valeur de flot fractionnaire supérieure à s , un seuil prédéterminé. Si aucune variable ne prend une valeur supérieure à s , alors la variable Θ_w^k ayant la plus grande valeur est fixée à 1. Dans le cas contraire, on définit l'ensemble C des variables Θ_w^k qui ont une valeur de flot fractionnaire supérieure à s . On choisit le plus grand sous-ensemble $C' \subseteq C$ tel que

$\sum_{\Theta_w^k \in C'} (1 - \Theta_w^k) \leq m$ et C' contient les $|C'|$ éléments de C ayant la plus grande valeur fractionnaire ; m est un paramètre prédéfini. On fixe à 1 toutes les variables appartenant à C' . Les valeurs de s et m utilisées lors des tests étaient respectivement de 0.75 et de 0.25. Il est à noter que cette stratégie de branchement ne fixe aucune variable à 0. On explore l'arbre de branchement selon la technique profondeur d'abord sans retour arrière.

Ces types de décision sur les variables Θ_w^k ne changent pas la nature de la formulation du problème maître, ni celles des sous-problèmes. Par conséquent, la technique de génération de colonnes peut s'appliquer à chaque noeud de l'arbre de branchement.

2.4 Extensions

Le modèle de base qui a été présenté ne permet de traiter qu'une petite partie de l'ensemble des règles associées au problème présenté par Omari (2002). Nous présentons ici les extensions à ajouter au modèle de base pour traiter le problème en entier. Ces extensions sont de quatre types distincts :

- ajout de ressources 2.4.1, 2.4.2 ;
- ajout de pénalité dans le problème maître (2.4.3, 2.4.5, 2.4.6) ;
- modification du réseau (2.4.4, 2.4.7, 2.4.10, 2.4.9) ;
- heuristique (2.4.8).

Il est à noter que ces extensions restent dans le cadre du modèle de base et ne modifient pas la structure du problème mathématique à résoudre. En revanche, l'ajout de ces extensions alourdit passablement la résolution du problème.

La présentation des extensions se divise en autant de parties qu'il reste de contraintes non couvertes. Chaque partie explique brièvement la nature d'une extension et l'im-

part de celle-ci sur la résolution du problème.

2.4.1 Durée d'affectation maximale et minimale variable

Dans le modèle de base, on a considéré que la durée maximale et la durée minimale d'affectation à une activité sont constantes durant la journée. Dans le cas où ces durées varient selon l'heure de la journée, on doit, pour assurer correctement le respect des contraintes, ajouter deux ressources permettant de déterminer la valeur de ces durées limites pour une période $p \in P$ donnée. L'utilisation d'arcs de type $aDebA$ d'une longueur d_{min} ne suffit plus à respecter la durée d'affectation minimale à une activité. On doit donc ajouter une ressource semblable à celle utilisée pour le respect de la durée maximale. Cependant, une seule ressource de type $rNum$ suffit pour les contraintes de durée maximale et minimale. L'ajout de ces trois ressources augmente le temps de résolution des sous-problèmes.

2.4.2 Contraintes sur les activités stressantes

On impose le respect de la durée maximale d'affectation consécutive à une activité stressante à l'aide de deux ressources. L'une cumule le temps consécutif assigné à une activité stressante. L'autre permet de déterminer la durée maximale d'affectation à une activité stressante à une période donnée (considérant que cette durée peut varier selon l'heure de la journée). L'ajout de ressources augmente le temps de résolution des sous-problèmes.

2.4.3 Traitement des activités fermées

Une activité est dite fermée à une période lorsque la demande de l'activité est nulle. Une liste d'activités peut être associée à chaque activité fermée. En cas de besoin,

un employé affecté à l'une des activités associées peut alors effectuer les tâches de l'activité fermée. Les activités fermées sont traitées de façon très semblable aux autres activités, i.e., à l'aide de contraintes et de variables d'écart Y_{ap} . La particularité de l'activité $a \in A$ fermée durant la période $p \in P$ est que ce sont d'autres activités $a_i \in A'$, $A' \subset A$, $a \notin A'$ qui vont contribuer à sa couverture durant la période p . Ces nouvelles contraintes et ces nouvelles variables étant en tout point semblables à celles utilisées pour pénaliser la sous-couverture d'une activité dans le modèle de base, l'impact de cette extension se limitera à l'ajout d'une contrainte et d'une variable par période $p \in P$ et par activité $a \in A$ tel que a est fermée durant la période p .

2.4.4 Respect de la durée de base

On souhaite, dans la mesure du possible, que les durées d'affectation des employés aux activités soient des multiples entiers de la durée de base, noté Δ ($\Delta = m\delta$, $\delta =$ durée d'une période). Pour ce faire, des arcs de type $aDebA$ et $aAct$ ayant des durées qui sont des multiples de la durée de base sont ajoutés aux réseaux si l'arc n'est pas déjà présent dans le réseau. Les nouveaux arcs $aDeb$ ont une durée supérieure au temps minimum d'affectation. Les arcs $aDebA$ et $aAct$ déjà existants se voient ajouter une pénalité strictement positive γ^{DB} . L'ajout de nouveaux arcs alourdit le réseau ce qui augmente le temps de résolution des sous-problèmes.

2.4.5 Répartition de la sous-couverture entre les périodes

On ajoute, pour chaque période $p \in P$, une variable entière non négative $TOTSCP_p$ qui représente la somme des sous-couvertures à la période p . Une pénalité correspondant à la sous-couverture des demandes durant une période est modélisée à l'aide d'une fonction linéaire par morceaux. Cette structure de coût permet d'augmenter

le coût unitaire par sous-couverture pour une période $p \in P$ lorsque la valeur de $TOTSCP_p$ croît, ce qui favorise la répartition de la sous-couverture entre les périodes. Cette extension ajoute $|P^k|$ contraintes et $|P^k| * nbSeg$ variables au problème maître, où $nbSeg$ est le nombre de segments de la fonction linéaire par morceaux. Ces nouvelles contraintes et variables permettent aussi d'imposer une borne supérieure sur le nombre de sous-couvertures à une période $p \in P$ donnée.

2.4.6 Répartition de la sous-couverture entre les activités

Selon les priorités, il peut être préférable de répartir la sous-couverture entre les activités. On veut donc augmenter le coût unitaire d'une sous-couverture pour une activité $a \in A$ à la période $p \in P$ à mesure que le ratio $\frac{Y_{ap}}{n_{ap}}$ se rapproche de 1. Pour ce faire, on modélise cette pénalité à l'aide d'une fonction linéaire par morceaux. Cette pénalité exige, $\forall a \in A$ et $\forall p \in P$, de remplacer la variable Y_{ap} par les variables Z_{api} , $i = \{1, \dots, nbSeg\}$, où $nbSeg$ est le nombre de segments de la fonction linéaire par morceaux. Cette extension augmente le nombre de variables du problème maître.

2.4.7 Respect des distributions cibles pour l'entretien des compétences

Lors de l'élaboration du quart $k \in K$ d'un employé donné, on cherche à obtenir pour chaque activité $a \in A$, une durée d'affectation ciblée, notée nmc_{ak} et exprimée en minutes, permettant l'entretien des compétences. Pour tout $a \in A$, on définit la valeur nmt_{ak} , le nombre de minutes déjà allouées à l'activité a au début de la tranche de temps. Pour chaque activité $a \in A^k$, on favorise l'obtention de la valeur cible nmc_{ak} en ajoutant, pour chaque période $p \in P^k$, des coûts aux arcs correspondant à l'affectation à l'activité a à la période p qui sont fonctions du ratio $r_{ak}^{EC} = \frac{nmt_{ak}}{nmc_{ak}}$.

Les coûts sont strictement positifs si $r_{ak}^{EC} \geq 1$, et strictement négatifs sinon. Cette extension ne fait que modifier les coûts dans les sous-problèmes et n'a que très peu d'impact sur les temps de résolution.

2.4.8 Traitement de la formation des stagiaires

Le traitement de la formation des stagiaires va de pair avec la disponibilité des instructeurs. On simplifie donc le problème de la façon suivante, un stagiaire associé au quart $k_s \in K$ suit l'activité de formation $a_s \in A^k$ à une période $p \in P$ s'il existe un instructeur affecté au quart $k_i \in K$ travaillant, à la période p , à une activité a_i favorisant la formation associée à l'activité a_s . On incite les instructeurs à participer à des activités de formation à l'aide d'une méthode similaire à celle utilisée pour le respect des distributions cibles pour l'entretien des compétences à l'exception que les ratios utilisés sont ceux associés aux activités de formation des stagiaires. Les quarts des stagiaires sont donc retirés des sous-problèmes, et sont calqués sur les quarts des instructeurs. Une heuristique très rapide intervient pour compléter certaines périodes des quarts attribués aux stagiaires. Cette extension n'affecte pratiquement pas le temps de résolution.

2.4.9 Traitement de la réserve

Dans certains contextes, il est possible de mettre des employés en réserve. Des noeuds et des arcs sont donc ajoutés aux réseaux pour modéliser la possibilité d'affectation à la réserve. Pour chaque quart $k \in K$, une valeur cible est associée à la réserve. Le respect de cette valeur cible est traité de façon semblable à l'entretien des compétences.

2.4.10 Traitement des quarts avec conditions initiales

Un quart avec conditions initiales commence avant le début de l'horizon d'optimisation. Ces quarts de travail sont complètement remplis avant le début de l'horizon par des activités, des pauses ou des périodes de réserve qui doivent rester intactes. Pour inclure un tel quart $k \in K$ dans le modèle, on étend le réseau G^k de façon à y inclure le début du quart. La nouvelle partie du réseau est un chemin isolé qui représente la suite des activités, des pauses ou des périodes de réserve établie pour la période du quart se déroulant avant l'horizon d'optimisation. Ce chemin force le respect des conditions initiales et permet de cumuler l'information associée aux différentes ressources du modèle. L'extension du réseau n'a pas d'impact sur le temps de résolution des sous-problèmes puisqu'il s'agit d'un chemin isolé.

2.4.11 Traitement de la sur-couverture

On pourrait interdire la sur-couverture en forçant l'égalité sur les contraintes (2.2). En pratique, cette méthode augmente considérablement le temps de résolution du problème. C'est pourquoi des variables de surplus sont utilisées pour permettre la sur-couverture. Il est donc possible que la solution générée renferme des sur-couvertures, et ce même si un coût est associé à chacune de ces variables de surplus pour pénaliser ce genre de solution. Pour cette raison, une heuristique est utilisée pour éliminer la sur-couverture des demandes.

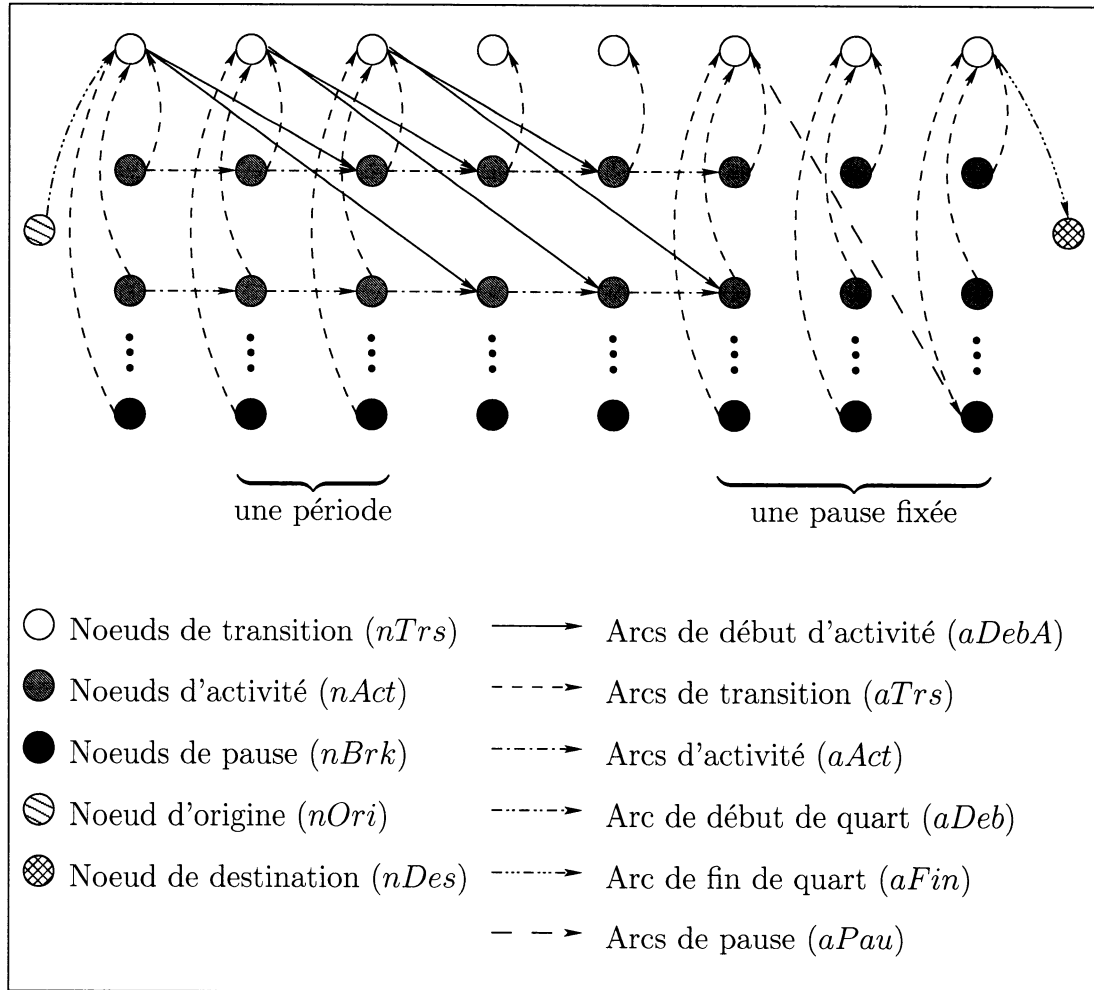


Figure 2.1 – Un exemple de réseau $G^k = (V^k, E^k)$

CHAPITRE 3 : OPTIMISATION DES PAUSES

L'optimisation des pauses consiste à modifier le modèle de base afin qu'il puisse traiter le problème plus général qui consiste à attribuer des activités et des pauses à un quart de travail. L'optimisation des pauses peut survenir dans deux contextes différents. Dans le premier cas, un ensemble de pauses initiales est donné, mais on veut les déplacer dans l'espoir d'obtenir une meilleure solution. Le deuxième scénario veut qu'il n'y ait pas d'ensemble de pauses initiales, la méthode de résolution devra donc placer les pauses dans les quarts au même titre qu'elle place les activités dans les quarts. Dans ce chapitre, une notation spécifique aux notions associées aux pauses est d'abord introduite. Une méthode permettant la réoptimisation des pauses initiales (le premier scénario) est ensuite présentée. Une troisième section présente les méthodes que nous avons développées pour faire le placement des pauses dans les quarts (le deuxième scénario). Les algorithmes que nous avons étudiés sont un amalgame de ces différentes méthodes. La section suivante présente ces différentes compositions de méthodes. Le chapitre se termine sur une présentation des résultats obtenus sur des jeux de données provenant de l'industrie.

3.1 Notation concernant les pauses

L'optimisation des pauses a pour résultat l'obtention, dans la solution du problème, d'un ensemble de pauses B^k pour chaque quart $k \in K$ et, pour chaque pause $b \in B^k$, d'un intervalle $Pause_b^k = [t, t']$ où $t, t' \in T^k$ correspondent respectivement au temps de début et au temps de fin de la période de pause. Les pauses ainsi générées doivent satisfaire aux contraintes suivantes :

- CP1 : respect de la durée minimale et maximale permise pour une pause ;
- CP2 : respect de la durée de pause totale minimale et maximale permise pour un quart ;
- CP3 : respect de la durée minimale et maximale permise entre deux pauses consécutives.

Cette section introduit la notation permettant de décrire les périodes de pause et se termine en utilisant cette notation pour exprimer formellement les contraintes (CP1)-(CP3).

- $bd_{b,max}^k$, (bd = break duration) la durée maximale de la pause $b \in B^k$ pour le quart $k \in K$.
- $bd_{b,min}^k$, la durée minimale de la pause $b \in B^k$ pour le quart $k \in K$.
- td_{max}^k , (td = total duration) la durée totale maximale des périodes de pause pour le quart $k \in K$.
- td_{min}^k , la durée totale minimale des périodes de pause pour le quart $k \in K$.
- cb_{max}^k , (consecutive breaks) la durée maximale pouvant séparer deux pauses consécutives pour le quart $k \in K$.
- cb_{min}^k , la durée minimale pouvant séparer deux pauses consécutives pour le quart $k \in K$.
- SB_b^k , une variable indiquant le temps de début de la pause b . alors $SB_b^k = t$.
- EB_b^k , une variable indiquant le temps de fin de la pause b . alors $EB_b^k = t'$.
- β_i^k , la pause dont la position dans l'ordre chronologique d'apparition des débuts de pause est l'entier $i \in \{1, \dots, |B^k|\}$. Si on a $i, j \in \{1, \dots, |B^k|\}$ et $i < j$ alors $SB_{\beta_i^k}^k < SB_{\beta_j^k}^k$.
- $num_b^k \in \{1, \dots, |B^k|\}$, la position chronologique de la pause $b \in B^k$, ainsi $\beta_{num_b^k}^k = b$.

Considérant une solution donnant un ensemble de pauses B^k , on peut formuler les contraintes (CP1)-(CP3) de la façon suivante :

$$bd_{b,min}^k \leq EB_b^k - SB_b^k \leq bd_{b,max}^k, \quad \forall b \in B^k \quad (3.1)$$

$$td_{min}^k \leq \sum_{b \in B^k} (EB_b^k - SB_b^k) \leq td_{max}^k \quad (3.2)$$

$$cb_{min}^k \leq EB_{\beta_i^k}^k - SB_{\beta_{i+1}^k}^k \leq cb_{max}^k, \quad \forall i = \{1, \dots, |B^k| - 1\} \quad (3.3)$$

$$SB_b^k, EB_b^k \in T^k, \quad \forall b \in B^k. \quad (3.4)$$

Les équations (3.1)-(3.3) correspondent respectivement aux contraintes (CP1)-(CP3). Les contraintes (3.4) assurent que les pauses commencent et se terminent dans l'horizon d'optimisation H et dans le quart $k \in K$.

3.2 Réoptimisation des pauses initiales

Lors de la présentation du modèle de base, nous avons défini l'ensemble $Pause_b^k$ des intervalles $[t, t']$ où $t, t' \in T^k$ durant lequel la pause $b \in B^k$ est fixée. Quoique ces pauses respectent les contraintes (3.1)-(3.4), il existe potentiellement d'autres ensembles du même type que $Pause_b^k$ respectant ces contraintes pour lesquels la solution au problème d'attribution des activités serait de meilleure qualité. Cette section présente une technique développée par un analyste du GERAD (É.Vatri) et qui consiste à perturber, au cours du processus d'optimisation, l'ensemble initiale $Pause_b^k$ afin d'améliorer la solution du problème. Notons que les conditions d'application de cette méthode (i.e., posséder un ensemble de pauses initiales) ne s'applique pas au problème plus général que nous voulons étudier. Cette méthode est néanmoins importante car elle est utilisée à l'intérieur des méthodes visant à résoudre le problème d'affectation des activités et des pauses dans le cas où nous n'avons pas d'ensemble de pauses initiales.

3.2.1 Méthode de perturbation des pauses initiales

Le principe général de cette technique est de permettre à la technique de résolution de modifier quelque peu le début de chacune des pauses $b \in B^k$. On définit fx_{max} comme étant la flexibilité permise (en nombre de périodes) sur le début d'une pause initiale. Concrètement, on cherche pour chaque pause $b \in B^k$, un entier fx_b , $|fx_b| \leq fx_{max}$, nous permettant de définir $Pause'_b^k = [t + fx_b \cdot \delta, t' + fx_b \cdot \delta]$.

Modification du réseau

Dans le modèle de base, la pause $b \in B^k$, ayant l'intervalle $Pause_b^k = [t, t']$, est fixée en plaçant dans le réseau G^k un arc de type $aPau$ entre le noeud $(nTrs, t)$ et le noeud $(nPau, t)$. On force le passage sur cet arc en enlevant du réseau G^k tous les arcs des types $aDebA$ et $aAct$ commençant ou finissant dans l'intervalle $[t, t']$. À l'exception de ces arcs, le réseau utilisé dans la méthode de perturbation des pauses initiales est très similaire à celui présenté à la sous-section 2.2.2. Voici comment construire ce réseau à partir du réseau G^k du modèle de base.

- Tous les arcs de types $aDebA$ et $aAct$ qui ont été retirés du réseau G^k parce qu'ils commençaient ou finissaient dans l'intervalle occupé par une pause sont remplacés dans le réseau G^k .
- Pour chaque pause $b \in B^k$ et l'intervalle correspondant $Pause'_b^k = [t, t']$ et pour chaque $i \in [-fx_{max}, fx_{max}]$, on ajoute, si $t + i \cdot \delta, t' + i \cdot \delta \in T^k$, un arc de type $aPau$ entre le noeud $(nPau, t + i \cdot \delta)$ et le noeud $(nPau, t' + i \cdot \delta)$.

Ce nouveau réseau (voir Figure 3.1 pour le cas $fx_{max} = 2$) permet de déplacer le début des pauses. On dénote par $FlexB_b^k$ l'ensemble des arcs du réseau associé à la pause $b \in B^k$. Soit $t \in T^k$ le temps de début de la pause initiale $b \in B^k$. Si

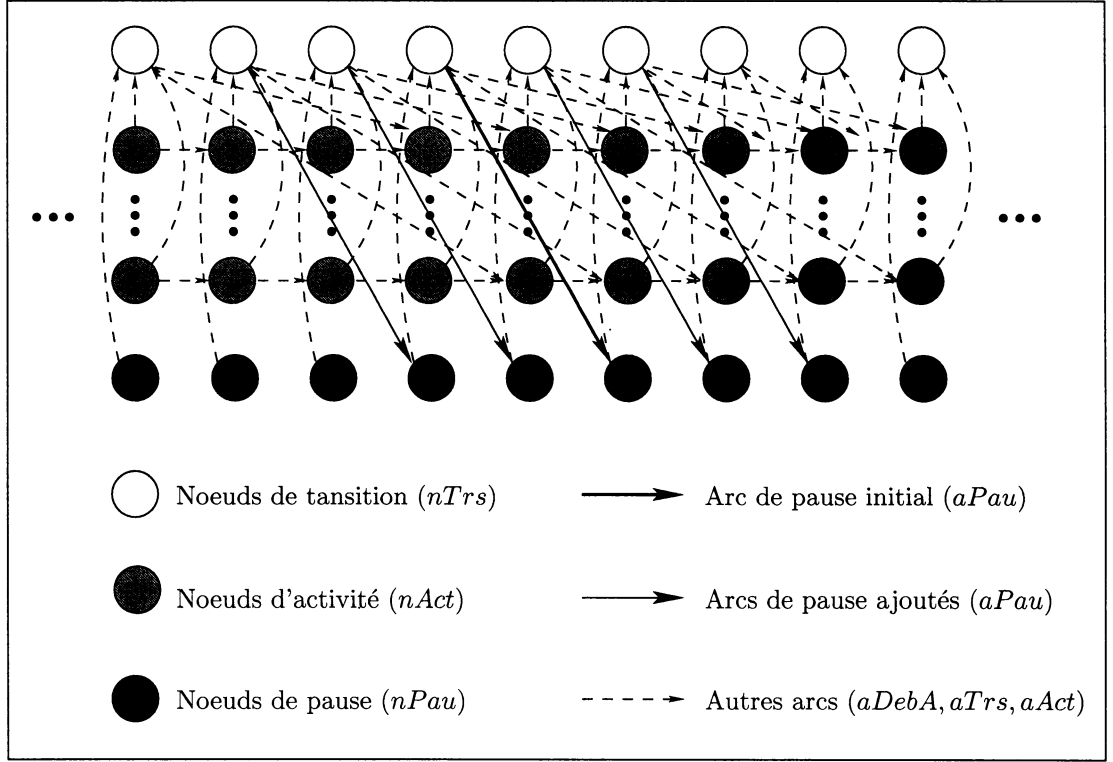


Figure 3.1 – Une partie d'un réseau G^k utilisé par la méthode de perturbation des pauses initiales

le chemin parcourant le réseau passe par l'arc $(i, j) \in FlexB_b^k$ dont l'origine est le noeud $(nTrs, t + l \cdot \delta)$, où l est un entier $\in [-fx_{max}, fx_{max}]$ alors $fx_b = l$. fx_b sera défini de façon univoque pour tout $b \in B^k$ si et seulement si le chemin parcourant le réseau emprunte, pour chaque $b \in B^k$, exactement un seul arc $(i, j) \in FlexB_b^k$. On obtient ce résultat en utilisant la ressource $rPau$ dont le comportement est semblable à la ressource $rNum$ décrite à la section 2.2.2. Les conditions imposées à $rPau$ sont définies comme un droit de passage sur un arc. Plus précisément, on peut emprunter l'arc $(i, j) \in E^k$ si on respecte la condition suivante :

$$(\Phi_i^{k,rPau} = \varphi_{ij}^{k,rPau} - 1) \vee (\varphi_{ij}^{k,rPau} = 0). \quad (3.5)$$

Tableau 3.1 – Fonctions de prolongation de la ressource $rPau$

Type d'arc	$\varphi_{ij}^{k,rPau}$	$f_{ij}^{rPau}(\Phi_i^{k,rPau})$
$aDeb$	0	0
$aFin$	$ B^k + 1$	$\Phi_i^{k,rPau}$
$aDebA$	0	$\Phi_i^{k,rPau}$
$aAct$	0	$\Phi_i^{k,rPau}$
$aPau$	$num_b^k, (i, j) \in FlexB_b^k$	$\Phi_i^{k,rPau} + 1$
$aTrs$	0	$\Phi_i^{k,rPau}$

La valeur $\Phi_i^{k,rPau}$ représente la position chronologique de la dernière pause assignée. Ainsi si le dernier arc de type $aPau$ parcouru appartient à l'ensemble $FlexB_b^k$ alors $\Phi_i^{k,rPau} = num_b^k$. $\Phi_i^{k,rPau}$ prend cependant la valeur 0 si on est au début du quart $k \in K$. Les valeurs de $\varphi_{ij}^{k,rPau}$ et les définitions des fonctions de prolongation $f_{ij}^{k,rPau}$ sont données au tableau 3.1.

On constate que le passage sur un arc appartenant à l'ensemble $FlexB_b^k$, $b \in B^k$ n'est permis que si la valeur de la ressource $rPau = num_b^k - 1$. Puisque la fonction de prolongation incrémente de 1 la valeur de cette ressource lors du passage sur un arc de l'ensemble $FlexB_b^k$ et que cette valeur ne décroît jamais, on peut conclure qu'on ne pourra passer que par un seul arc de cet ensemble. De plus, puisque que la valeur $\varphi_{ij}^{k,rPau}$ de l'arc (i, j) de type $aFin$ est $|B^k| + 1$, cet arc ne pourra être emprunté que si, pour chaque $b \in B^k$, on a emprunté exactement un arc de l'ensemble $FlexB_b^k$. Puisque qu'il est absolument nécessaire de passer par l'arc de type $aFin$, on conclut que la ressource $rPau$ assure que chaque pause $b \in B^k$ est assignée exactement une fois durant le quart $k \in K$. Il est à noter qu'il n'y pas de dominance entre deux étiquettes qui n'ont pas la même valeur de $\Phi_i^{k,rPau}$.

Toutes les pauses ainsi assignées sont dans le réseau G^k et respectent par conséquent les contraintes (3.4). Rien ne garantit cependant que ces pauses respectent les

contraintes (3.1)-(3.3). Les sous-sections suivantes présentent les ajouts qu'on doit faire au modèle pour que les pauses générées respectent toutes les contraintes qui leur sont associées.

Respect des contraintes sur la durée totale et individuelle des pauses

Si EB_b^k et SB_b^k sont les temps de départ et de fin de la pause $b \in B^k$ pour l'intervalle initial $Pause_b^k$ et $EB_b'^k$ et $SB_b'^k$ sont les temps de départ et de fins de la même pause pour l'intervalle $Pause_b'^k$, alors $EB_b'^k = EB_b^k + f_{x_b}$ et $SB_b'^k = SB_b^k + f_{x_b}$. Si les pauses initiales respectaient les contraintes (3.1)-(3.2), les nouvelles pauses les respecteront aussi puisque

$$\begin{aligned} bd_{b,min}^k &\leq EB_b'^k - EB_b'^k = EB_b^k - SB_b^k \leq bd_{b,max}^k, \forall b \in B^k; \\ td_{min}^k &\leq \sum_{b \in B^k} (EB_b'^k - SB_b'^k) = \sum_{b \in B^k} (EB_b^k - SB_b^k) \leq td_{max}^k. \end{aligned}$$

Respect de la contrainte sur la durée séparant deux pauses consécutives

En modifiant le début des pauses, les durées séparant les pauses consécutives peuvent avoir changées et ne plus se trouver dans la fourchette $[cb_{min}^k, cb_{max}^k]$ et ainsi violer les contraintes (3.3). Pour assurer le respect de cette contrainte, nous utiliserons deux ressources $rSansPauMin$ et $rSansPauMax$ dont les conditions sont exprimées comme des fenêtres de ressources sur les noeuds du réseau. Ces deux ressources comptabilisent la durée qui s'est écoulée depuis la fin de la dernière pause. À chaque noeud $i \in E^k$, la fenêtre pour la ressource $rSansPauMin$ est $[0, cb_{min}^k]$ et la fenêtre pour la ressource $rSansPauMax$ est $[0, cb_{max}^k]$. La fonction de prolongation est la même pour les deux ressources $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$. Les valeurs de $\varphi_{ij}^{k,rSansPauMin}$ et de $\varphi_{ij}^{k,rSansPauMax}$ sont données au tableau 3.2.

Tableau 3.2 – Consommation des ressources $rSansPauMin$ et $rSansPauMax$

Type d'arc	$\varphi_{ij}^{k,rSansPauMin}$	$\varphi_{ij}^{k,rSansPauMax}$
$aDeb$	cb_{min}^k	0
$aFin$	cb_{min}^k	0
$aDebA$	$-d_{ij}$	d_{ij}
$aAct$	$-d_{ij}$	d_{ij}
$aPau$	cb_{min}^k	$-cb_{max}^k$
$aTrs$	0	0

Performance de la méthode

La méthode présentée permet effectivement de faire varier le début des pauses initiales tout en respectant les contraintes qui ont trait aux pauses. Notons cependant que cette technique ne couvre pas tous les débuts de pause possibles avec l'ensemble B^k et peut donc donner une solution sous-optimale. En pratique, la méthode fonctionne très bien, permettant d'améliorer sensiblement les horaires fabriquées sans augmenter substantiellement les temps de calcul.

3.3 Placement des pauses dans les quarts

Dans la section précédente, nous avons utilisé un ensemble de pauses initiales dont les temps de début ont légèrement été modifiés dans le but d'améliorer la qualité des solutions obtenues. Cette section attaque le problème plus large où aucun ensemble de pauses initiales n'est disponible. Le concept de ressources étant un puissant outil de modélisation, nous présenterons, en premier lieu, un modèle où les contraintes (3.1)-(3.4) sont toutes imposées à l'aide de ressources. Comme cette modélisation entraîne d'importants problèmes de performance, une analyse de sa rapidité est présentée

dans le but de dégager des pistes pouvant mener à une méthode plus rapide. Deux approches visant à accélérer les temps de résolution sont ensuite présentées suivies d'une analyse de leurs performances.

3.3.1 Le modèle de pauses exact

Le modèle avec ressources est un modèle exact dans le sens où il permet d'obtenir des solutions optimales. Il est obtenu en modifiant légèrement le modèle de base et y ajoutant 5 ressources. La première sous-section présente le nouveau réseau G^k utilisé. Les sous-sections suivantes présentent les ressources utilisées pour imposer les contraintes. Enfin, la dernière partie de cette section fait une analyse des performances de la méthode.

Modification du réseau

L'objectif du réseau utilisé dans le modèle avec ressources est de permettre en tout temps $t \in T^k$ de commencer soit une activité, soit une pause. Le réseau G^k associé à ce modèle utilise un ensemble d'arcs E^k quelque peu différent de celui utilisé dans le modèle de base. Les arcs de type $aPau$ sont remplacés par deux autres types d'arcs, soit des arcs de type $aBegB$, semblables aux arcs $aDebA$, qui relient, pour chaque $t \in T^k$ le noeud $(nTrs, t)$ et le noeud $(nPau, t + bd_{min}^k)$ et des arcs de type $aBrk$, semblables aux arcs de type $aAct$, qui relient, pour chaque $t \in T^k$ le noeud $(nPau, t)$ et le noeud $(nPau, t + \delta)$.

Ce nouveau réseau est beaucoup plus général et permet de générer des quarts qui sont des suites de pauses et de différentes activités. De plus, ce réseau peut être étendu de façon à y inclure différents types de pauses en ajoutant de nouvelles séries

Tableau 3.3 – Consommation de la ressource $rDurPauMax$

Type d'arc	$\varphi_{ij}^{k,rDurPauMax}$
$aDeb$	0
$aFin$	0
$aDebA$	0
$aAct$	0
$aBegB$	d_{ij}
$aBrk$	d_{ij}
$aTrs$	$-bd_{max}^k$

de noeuds semblables aux noeuds de type $nPau$ et en attachant à ces nouvelles séries des arcs semblables aux arcs $aBegB$ et $aBrk$. Cependant, seul les contraintes (3.4) et une partie des contraintes (3.1) sont imposées par la structure du réseau. Les prochaines sous-sections présenteront les différentes méthodes utilisées pour imposer les contraintes (3.1)-(3.3).

Respect de la contrainte sur la durée individuelle des pauses

Le respect de la durée minimale est imposée lors de la construction du réseau en attribuant aux arcs de type $aBegB$ la longueur $bd_{b,min}^k$. En ce qui à trait au respect de la la durée maximale, la ressource $rDurPauMax$ est ajoutée au modèle. Cette ressource comptabilise la durée écoulée depuis le début d'une pause. À chaque noeud $i \in E^k$, la fenêtre pour cette ressource est $[0, bd_{max}^k]$. La fonction de prolongation est $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$. Les valeurs de $\varphi_{ij}^{k,rDurPauMax}$ sont données au tableau 3.3.

Tableau 3.4 – Consommation des ressources $rDurTotPauMin$ et $rDurTotPauMax$

Type d'arc	$\varphi_{ij}^{k,rDurTotPauMin}$	$\varphi_{ij}^{k,rDurTotPauMax}$
$aDeb$	td_{min}^k	0
$aFin$	td_{min}^k	0
$aDebA$	0	0
$aAct$	0	0
$aBegB$	$-d_{ij}$	d_{ij}
$aBrk$	$-d_{ij}$	d_{ij}
$aTrs$	0	0

Respect de la contrainte sur la durée totale des pauses

Pour s'assurer que le temps total alloué à des périodes de pause au cours d'un quart $k \in K$ reste à l'intérieur de l'intervalle $[td_{min}^k, td_{max}^k]$, on procède à l'ajout des ressources $rDurTotPauMin$ et $rDurTotPauMax$ au modèle. Ces ressources comptabilisent la durée cumulée du temps alloué aux pauses depuis le début du quart. À chaque noeud $i \in E^k$, la fenêtre pour la ressource $rDurTotPauMin$ est $[0, td_{min}^k]$ et la fenêtre pour la ressource $rDurTotPauMax$ est $[0, td_{max}^k]$. La fonction de prolongation est la même pour les deux ressources $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$. Les valeurs de $\varphi_{ij}^{k,rDurTotPauMin}$ et $\varphi_{ij}^{k,rDurTotPauMax}$ sont données au tableau 3.4.

Respect de la contrainte sur la durée séparant deux pauses consécutives

De la même façon que dans la méthode de perturbation des pauses, la contrainte sur la durée séparant deux pauses consécutives est prise en charge par les ressources $rSansPauMin$ et $rSansPauMax$.

Performance de la méthode

Selon ce modèle, on peut générer tous les ensembles de pauses rendant l'horaire réalisable. En particulier, ce modèle permet d'obtenir l'ensemble de pauses associé à l'horaire réalisable optimal. Cependant, cette souplesse a un coût considérable sur le temps d'exécution. En fait, sur des instances de taille typique, la méthode est complètement inutilisable car beaucoup trop longue. Deux raisons expliquent cette extrême lenteur, la première concerne le nombre élevé de ressources (la complexité de l'algorithme de programmation dynamique dépend du nombre de ressources), la deuxième est due à la nature contraire des ressources $rDurTotPauMin$, $rDurTotPauMax$ et des ressources $rSansPauMin$, $rSansPauMax$ (elles servent à imposer des bornes supérieures et inférieures strictes) car il est connu que de telles ressources allongent considérablement les temps de calculs. Les sections qui suivent présentent différentes façons de modifier le modèle avec ressources dans le but d'exploiter au maximum sa puissance de modélisation tout en obtenant des temps de résolution raisonnables.

3.3.2 Modélisation des sous-problèmes permettant la génération des pauses par agrégation des activités

Dans une des stratégies de la méthode à deux phases décrite à la prochaine section, on doit générer rapidement un ensemble de pauses valides pour un quart k en utilisant un modèle approximatif dans le sous-problème correspondant au quart k . Cette section a pour but de présenter un tel modèle, celui-ci diffère des modèles vus précédemment au sens qu'il ne représente pas complètement le problème d'assignation des activités. L'idée est de transformer le réseau G^k en un réseau ne comprenant qu'une seule activité, nommé activité générique, et une pause. Le modèle de pauses exact décrit à la sous-section 3.3.1 est utilisé pour résoudre chaque sous-problème associé à un

quart $k \in K$, à l'exception que les contraintes sur la durée maximale et minimale d'affectation à une activité sont relâchées. La construction précise de ce nouveau réseau se fait de la façon suivante (voir figure 3.2) : pour chaque $t \in T^k$, un arc de type $aAct$ relie $(nActGen, t)$ et $(nActGen, t + \delta)$ et un arc de type $aDebA$ relie $(nTrs, t)$ et $(nActGen, t + td_{min}^k)$. La valeur de td_{min}^k est égale à 2 sur la figure 3.2. À chacun de ces arcs est associé un coût égal à $-\min_{a \in A} \sum_{p \in P_{arc}} \alpha_{ap}$ où P_{arc} est l'ensemble des périodes comprises dans l'intervalle couvert par l'arc $[t, t + \delta]$ pour les arcs $aAct$ et $[t, t + d^{k,min}]$ pour les arcs $aDebA$ où $t + d^{k,min}$ est la plus petite durée minimale d'affectation à une activité). En ce qui à trait aux pauses, on construit, pour chaque $t \in T^k$, un arc de type $aPau$ reliant $(nPau, t)$ et $(nPau, t + \delta)$ et un arc de type $aDebP$ reliant $(nTrs, t)$ et $(nPau, t + bd_{min}^k)$. La valeur de bd_{min}^k est égale à 2 sur la figure 3.2.

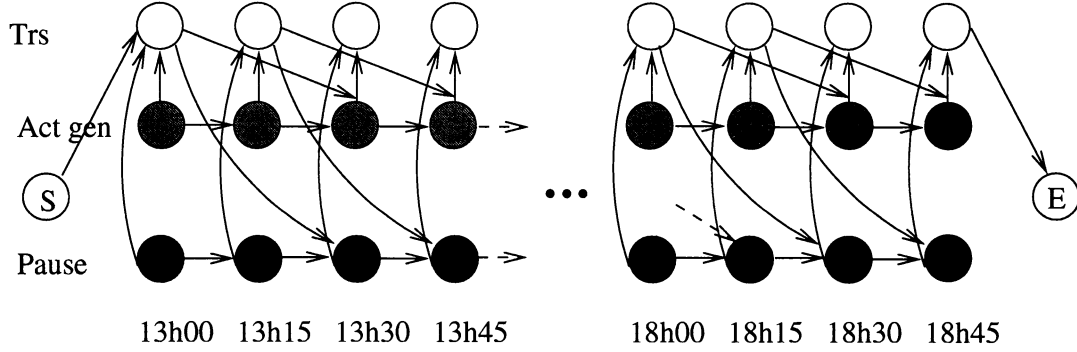


Figure 3.2 – Réseau agrégé utilisé lors de la première phase

Pour que cette méthode ait du succès, il est primordial que l'activité générique représente le mieux possible l'ensemble des activités qu'elle substitue. Pour ce faire, on utilisera, dans l'itération courante, l'information duale de la solution du dernier problème maître résolu pour effectuer l'agrégation des activités. Pour bien comprendre l'utilité des variables duales dans la construction de l'activité générique, considérons l'interprétation suivante de la variable duale α_{ap} associée aux contraintes (2.2) pour

l'activité $a \in A$ à la période $p \in P_a$. La valeur de α_{ap} correspond au gain (ou la perte en cas de sur-couverture) qu'on ferait si la demande pour l'activité $a \in A$ à la période $p \in P_a$ diminuait de 1. On peut interpréter cette information comme le gain (ou la perte) potentiel associé à la couverture de $a \in A$ à la période $p \in P_a$. Ainsi, couvrir l'activité générique à la période $p \in P$ procure le même avantage que couvrir l'activité la plus prometteuse pour cette période du quart de travail. De cette façon, les pauses devraient être attribuées aux tranches de temps où la couverture des activités est la plus facile à réaliser.

3.3.3 La méthode à deux phases

La méthode à deux phases est un regroupement de différentes stratégies qui reposent toutes sur un principe commun, soit de séparer, à chaque itération de génération de colonnes, la résolution de chaque sous-problème en deux phases. Lors de la résolution des sous-problèmes, la première phase génère un ensemble de pauses respectant les contraintes (CP1)-(CP3). On peut obtenir cet ensemble soit en résolvant le problème de plus court chemin associé au modèle de génération des pauses par agrégation des activités, soit en résolvant celui associé au modèle de pauses exact. La deuxième phase génère un quart rempli d'activités, en utilisant la partie du modèle de base ayant trait aux sous-problèmes (avec ou sans la méthode de perturbation des pauses). Les pauses fixes utilisées lors de cette deuxième phase sont celles obtenues à la première phase. Les stratégies diffèrent selon la façon retenue pour construire les ensembles de pauses. De plus, certaines stratégies se distinguent en utilisant l'ensemble de pauses générées à la première phase pour plus d'une itération (i.e. certaines itérations n'exécuteront que la deuxième phase). On doit alors déterminer des critères nous indiquant quand de nouvelles pauses doivent être générées, c'est-à-dire, déterminer à chaque itération si l'on doit ou non exécuter la première phase. Les algorithmes 3.1, 3.2 et 3.3 expliquent le fonctionnement des différentes stratégies utilisées.

Algorithme 3.1 Algorithme pour la génération continue des pauses par agrégation des activités

```

for all itération de génération de colonnes do
  Résoudre le problème maître
  for all Quart à optimiser do
    % Sous-problèmes %
    % Phase 1 %
    - Générer le sous-problème correspondant selon
      le modèle de génération des pauses
      par agrégation des activités
    - Résoudre et garder les pauses obtenues en mémoire
    % Phase 2 %
    if méthode de perturbation des pauses then
      - résoudre le modèle avec pauses fixées avec la
        méthode de perturbation des pauses et l'ensemble
        de pauses en mémoire
    else
      résoudre le modèle avec pauses fixées et l'ensemble
      de pauses en mémoire

```

Il n'y a pas stratégie de génération continue des pauses avec boucle utilisant le modèle exact, car une telle stratégie serait équivalente à utiliser une méthode utilisant le modèle de pauses exact sur un problème ayant un ensemble initial de colonnes de bonne qualité (malgré cet ensemble initial de colonnes, les temps d'exécution restent considérables).

3.3.4 Critères déterminant le potentiel d'un ensemble de pauses

Considérant un ensemble de pauses donné, on décrètera que cet ensemble n'a plus de potentiel si un des scénarios suivants se produit :

- Les sous-problèmes ont généré quelques colonnes, mais les variables duales du problème maître n'ont pas changées.

Algorithme 3.2 Algorithme pour la génération intermittente des pauses par agrégation des activités

```

for all itération de génération de colonnes do
  Résoudre le problème maître
  for all Quart à optimiser do
    % Sous-problèmes %
    if Pauses sans potentiel ou première itération then
      % Phase 1 %
      - Générer le sous-problème correspondant selon le
        modèle de génération des pauses
        par agrégation des activités
      - Résoudre et garder les pauses obtenues en mémoire pour
        les prochaines itérations
    % Phase 2 %
    if méthode de perturbation des pauses then
      - résoudre le modèle avec pauses fixées avec la
        méthode de perturbation des pauses et l'ensemble
        de pauses en mémoire
    else
      - résoudre le modèle avec pauses fixées et l'ensemble
        de pauses en mémoire

```

- Les sous-problèmes n'ont généré aucune nouvelle colonne.
- Le gain obtenu, depuis les n (typiquement 20) dernières itérations, sur la fonction objectif depuis qu'on utilise cet ensemble de pauses est inférieur à une certaine fraction (typiquement 10%) du gain obtenu avec l'ensemble de pauses précédent. Supposons qu'en utilisant l'ensemble de pauses précédent la valeur de l'objectif soit passée de 15970 à 15620 pour une réduction de 350. L'ensemble de pauses actuel sera jugé sans potentiel si il n'a pas permis une réduction d'au moins 35 depuis les 20 dernières itérations.

Ce dernier critère permet de juger l'efficacité d'un ensemble de pauses relativement aux ensembles précédents. En général, la fonction objectif a tendance à décroître rapidement durant les premières itérations, et lentement lorsqu'on s'approche de la valeur optimale. C'est pourquoi il est utile de disposer d'un critère permettant de

Algorithme 3.3 Algorithme pour la génération intermittente des pauses avec boucle utilisant le modèle exact

```

Utiliser modèle exact = FAUX
repeat
  % itération de génération de colonnes %
  if critère d'arrêt atteint then
    Utiliser modèle exact = VRAI
  Résoudre le problème maître
  for all Quart à optimiser do
    % Sous-problèmes %
    if Pauses sans potentiel ou première itération then
      % Phase 1 %
      if Utiliser modèle exact then
        - Générer le sous-problème correspondant selon le
          le modèle de pauses exact
      else
        - Générer le sous-problème correspondant selon le
          modèle de génération des pauses
          par agrégation des activités
        - Résoudre et garder les pauses obtenues en mémoire pour
          les prochaines itérations
      % Phase 2 %
      if méthode de perturbation des pauses then
        - résoudre le modèle avec pauses fixées avec la
          méthode de perturbation des pauses et l'ensemble
          de pauses en mémoire
      else
        - résoudre le modèle avec pauses fixées et l'ensemble
          de pauses en mémoire
  until !critère d'arrêt atteint OU !Utiliser modèle exact

```

déterminer dynamiquement le potentiel de l'ensemble actuel de pauses.

3.4 Résultats numériques

Dans le but d'évaluer les performances du modèle à deux phases, nous l'avons exécuté sur un ensemble de problèmes issues d'instances réelles de problèmes de fabrication d'horaires pour des contrôleurs aériens. Ces problèmes se répartissent en deux ensembles distincts :

- Des problèmes opérationnels où un horaire est en majeure partie rempli et doit être complété. Les problèmes de ce type partagent tous un horizon d'une journée ou moins.
- Des problèmes complets où un horaire doit être construit à partir de zéro. Deux problèmes de ce type sont définis pour un horizon d'un mois, alors qu'un autre s'étend sur 13 jours.

Les caractéristiques de ces problèmes sont données au tableau 3.5. À l'intérieur de ce tableau, on retrouve les informations suivantes : la longueur de l'horizon d'optimisation en jours (Durée), le nombre d'activités (Nb. Act), le nombre de quarts (Nb. Quarts), le nombre d'employés (Nb. Emp.), le nombre de demandes (Nb. Dem.) pour des activités de couverture sur l'horizon d'optimisation (une demande pour une activité = une période) et finalement le nombre de tranches de temps (Nb. Tranches de temps) générées par la stratégie de décomposition temporelle.

L'analyse des performances s'est faite sur deux paramètres : le nombre de sous-couvertures et le temps de calcul (en minutes). Rappelons qu'une sous-couverture correspond à un manque d'effectif pour une activité pour un intervalle de temps de longueur δ . Ce paramètre est primordial puisque l'objectif principal des problèmes qu'on cherche à résoudre est de minimiser le nombre de sous-couvertures. Les résultats ont été comparés à ceux obtenus avec la méthode de perturbation des pauses initiales, pour laquelle l'ensemble de pauses initiales avait été généré par un module

Tableau 3.5 – Résultats pour la méthode de génération continue des pauses par agrégation des activités

Scénario	Durée (jour)	Nb. Act.	Nb. Quarts	Nb. Emp.	Nb. Dem.	Nb. Tranches de temps
aa12-11-330-1350/	.7	12	51	51	568	8
aa23-11/	1	12	53	51	604	11
aa4au16/	13	9	563	51	5880	137
aa7-11/	1	12	58	51	622	11
aa25-11/	1	11	55	51	604	11
aa-EBG1-oct/	32	9	1336	51	14894	341
aa-EBG2-oct/	31	13	1698	51	22220	331
aaEBG3-01-11-01/	1	12	58	51	568	11

spécifique. Il est à noter que l'utilisation de la méthode de perturbation des pauses initiales ralentit passablement l'algorithme existant auquel on se compare. Un tableau de résultats est donné pour chacune des trois stratégies présentées précédemment, de plus, pour les deux dernières stratégies, un tableau de résultats est donné pour illustrer leur performance lorsqu'utilisées en conjonction avec la méthode de perturbation des pauses (flexibilité). Ces tableaux contiennent l'information suivante : le nombre de sous-couverture (SC), le temps de résolution en minutes (Temps), la différence de sous-couverture en pourcentage de la quantité totale de sous-couvertures obtenues avec l'algorithme auquel on se compare (Diff SC) et le rapport du temps de calcul de la stratégie étudiée sur celui de l'algorithme auquel on se compare (Ratio temps). Notons finalement que les tests ont été effectués sur une station de travail Sun Sun-Fire avec un processeur à 880 MHz.

3.4.1 Génération continue des pauses par agrégation des activités

Les résultats pour la stratégie de génération continue des pauses par agrégation des activités (voir tableau 3.6) sont très encourageants car ils montrent que l'on peut effectuer relativement rapidement l'optimisation simultanée des activités et des pauses. Rappelons que le premier modèle proposé, le modèle de pauses exact, présentait des temps de résolution excessif. Cette stratégie augmente les temps de calcul que de 73% ce qui est un réel progrès. Notons cependant que l'augmentation du nombre de sous-couvertures de 5% est difficile à expliquer. Nous avons pensé que la qualité des solutions obtenues serait au moins aussi bonne qu'avec le modèle avec pauses fixées. Deux causes peuvent expliquer cet écart. La première est que l'activité générique utilisée ne représente pas assez bien les différentes activités qu'elle remplace (d'autres modèles pourraient être utilisés). La deuxième hypothèse est que les critères d'arrêt de la boucle d'optimisation sont trop sensibles, ce qui a pour effet d'arrêter l'optimisation avant que les ensembles de pauses les plus prometteurs soient générés.

3.4.2 Génération intermittente des pauses par agrégation des activités

Les résultats pour la stratégie de génération intermittente des pauses par agrégation des activités sont présentés au tableau 3.7. Cette stratégie a été développée dans le but d'accélérer la stratégie précédente. Pour ce faire, la phase de génération de pauses, qui représente environ 80% du temps de calcul des sous-problèmes d'une itération de génération de colonnes (voir tableau 3.11), est exécutée de façon intermittente. On tire ainsi le maximum de chaque ensemble de pauses généré. Cette approche est justifiée par le fait que dans la stratégie précédente, la phase de génération des ensembles de pauses peut générer plusieurs fois le même ensemble de pauses. Comme prévu,

Tableau 3.6 – Résultats pour la méthode de génération continue des pauses par agrégation des activités

Scénario	Pauses Fixées		Pauses Opt.		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	4	12	-56	4.00
aa23-11/	11	1	12	5	9	5.00
aa4au16/	19	14	39	40	105	2.86
aa7-11/	9	4	11	7	22	1.75
aa25-11/	14	3	5	12	-64	4.00
aa-EBG1-oct/	139	60	165	81	19	1.35
aa-EBG2-oct/	2145	142	2236	244	4	1.72
aaEBG3-01-11-01/	15	10	12	9	-20	0.90
Total	2361	237	2484	422	5	1.73

le temps d'exécution a significativement baissé et est maintenant un peu inférieur (-11%) à celui du modèle avec pauses fixées. La qualité des solutions n'est pas significativement différente de celle obtenue avec la stratégie précédente.

3.4.3 Génération intermittente des pauses par agrégation des activités avec flexibilité

Ici on a ajouté la méthode de perturbation des pauses à la stratégie précédente dans le but d'améliorer la qualité des solutions. On aperçoit (voir tableau 3.8) que le nombre de sous-couvertures a augmenté de 2% en comparaison au nombre de sous-couvertures obtenu avec le modèle avec pauses fixées, ce qui est mieux que le 5% d'augmentation obtenue sans flexibilité. La flexibilité permet donc d'améliorer la qualité des solutions obtenues par la stratégie de génération intermittente des pauses par agrégation des activités. Cependant ce gain est obtenu au coût d'un temps de résolution presque

Tableau 3.7 – Résultats pour la méthode de génération intermittente des pauses par agrégation des activités

Scénario	Pauses Fixées		Pauses Opt.		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	4	5	-56	1.67
aa23-11/	11	1	11	5	0	5.00
aa4au16/	19	14	38	18	100	1.29
aa7-11/	9	4	11	3	22	0.75
aa25-11/	14	3	5	5	-64	1.67
aa-EBG1-oct/	139	60	146	39	5	0.65
aa-EBG2-oct/	2145	142	2250	127	5	0.89
aaEBG3-01-11-01/	15	10	13	9	-13	0.90
Total	2361	237	2478	211	5	0.89

deux fois plus élevé, ce qui est en accord avec le fait que la méthode de perturbation des pauses accroît le temps de calcul passé à la résolution des sous-problèmes. Notons toutefois que l'algorithme utilisé pour l'application de la méthode de perturbation des pauses est très sommaire et pourrait être amélioré.

3.4.4 Génération intermittente des pauses avec boucle utilisant le modèle exact

Cette stratégie prend en compte le fait que la méthode de génération des pauses par agrégation des activités ne générera pas nécessairement les ensembles de pauses optimaux. Elle consiste à ajouter à la stratégie de génération intermittente des pauses par agrégation des activités une deuxième boucle d'itérations de génération de colonnes utilisant, de façon intermittente toujours, le modèle de pauses exact pour générer les ensembles de pauses. Cette stratégie mise sur l'hypothèse qu'exécuter de façon intermittente sur une solution de départ de qualité (celle obtenue par la boucle utilisant la

Tableau 3.8 – Résultats pour la méthode de génération intermittente des pauses par agrégation des activités avec flexibilité

Scénario	Pauses fixées		Pauses Opt.		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	5	9	-44	3.00
aa23-11/	11	1	13	5	18	5.00
aa4au16/	19	14	24	38	26	2.71
aa7-11/	9	4	10	6	11	1.50
aa25-11/	14	3	5	63	-64	21.00
aa-EBG1-oct/	139	60	158	76	14	1.27
aa-EBG2-oct/	2145	142	2182	233	2	1.64
aaEBG3-01-11-01/	15	10	12	11	-20	1.10
Total	2361	237	2409	441	2	1.86

méthode de génération des pauses par agrégation des activités) le modèle de pauses exact gardera sa puissance de modélisation tout en diminuant considérablement son temps de résolution. Les résultats obtenues (voir tableau 3.9) sont intéressants. Ils présentent un gain énorme sur la qualité des solutions, soit une diminution de 14% du nombre de sous-couvertures en comparaison au nombre de sous-couvertures obtenu avec le modèle avec pauses fixées. Cependant le temps de résolution, quoique maintenant raisonnable, reste très élevé (environ 8 fois le temps pris par le modèle avec pauses fixées).

3.4.5 Génération intermittente des pauses avec boucle utilisant le modèle exact avec flexibilité

On a testé la stratégie précédente en y ajoutant la méthode de perturbation des pauses. Les résultats obtenus sont surprenants. Premièrement, considérant que le

Tableau 3.9 – Résultats pour la méthode de générations intermittente des pauses avec boucle utilisant le modèle exact

Scénario	Pauses fixées		Pauses Opt.		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	0	21	-100	7.00
aa23-11/	11	1	9	17	-18	17.00
aa4au16/	19	14	31	126	63	9.00
aa7-11/	9	4	9	15	0	3.75
aa25-11/	14	3	6	40	-57	13.33
aa-EBG1-oct/	139	60	138	294	-1	4.90
aa-EBG2-oct/	2145	142	1826	1462	-15	10.30
aaEBG3-01-11-01/	15	10	14	16	-7	1.60
Total	2361	237	2033	1991	-14	8.40

modèle de pauses exact était très puissant, nous avons pensé que la méthode de perturbation des pauses n'aurait pas d'impact sur la qualité des solutions obtenues. Cette méthode aurait cependant pu diminuer le temps de calcul en diminuant le nombre d'ensembles de pauses à générer. Les résultats montrent (voir tableau 3.10) exactement le contraire, soit une amélioration substantielle de la qualité des solutions (diminution de 19% du nombre de sous-couvertures) et une augmentation considérable du temps de résolution (environ 24 fois le temps pris par le modèle avec pauses fixées). Les différentes investigations que nous avons menées ne nous ont pas permis de trouver une explication valable pour ces deux phénomènes. Seule une constatation a pu être dégagée : l'utilisation de la méthode de perturbation des pauses augmente énormément le nombre d'itérations de génération de colonnes requis pour résoudre le problème.

Tableau 3.10 – Résultats pour la méthode de génération intermittente des pauses avec boucle utilisant le modèle exact avec flexibilité

Scénario	Pauses fixées		Pauses Opt.		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	0	43	-100	14.33
aa23-11/	11	1	9	33	-18	33.00
aa4au16/	19	14	30	266	58	19.00
aa7-11/	9	4	10	38	11	9.5
aa25-11/	14	3	7	41	-50	13.67
aa-EBG1-oct/	139	60	131	545	-6	9.08
aa-EBG2-oct/	2145	142	1720	4729	-20	33.30
aaEBG3-01-11-01/	15	10	12	49	-20	4.90
Total	2361	237	1919	5744	-19	24.24

3.5 Conclusion

Le but du travail présenté dans ce chapitre visait principalement à développer une approche permettant de résoudre le problème d'assignation des activités dans le cas où l'on ne possède pas de pause initiale. Les méthodes présentées sont satisfaisantes. Elles permettent d'obtenir des nombres de sous-couvertures similaires à ceux obtenus par l'approche existante utilisant la méthode de perturbation des pauses sur un ensemble de pauses initiales et ce pour des temps de résolution semblables. On a aussi réussi à obtenir moins de sous-couvertures mais cette fois en utilisant plus de temps de calcul.

Les informations présentées au tableau 3.11 ont été obtenues en résolvant les scénarios par la stratégie de génération intermittente des pauses par agrégation des activités qui sera la seule stratégie retenue dans la méthode développée au chapitre 4. Ces informations nous permettent d'apporter quelques remarques. Le tableau 3.11 pré-

sente, pour chaque scénario, le temps passé à résoudre les sous-problèmes (SP Total) en secondes et le temps passé à résoudre le problème maître (PM Total) en secondes. De plus, pour chacune des phases, le tableau présente le nombre d'itérations (Nb. It.) utilisant cette phase et le temps de résolution de la partie des sous-problèmes résolvant cette phase (SP). Une première constatation concerne la différence de temps de calcul entre la phase 1 et la phase 2 : le temps moyen par itération pour résoudre les sous-problèmes à la phase 1 ($683/12074 = 0.057$) est quatre fois plus élevé que celui pris par la phase 2 ($893/63654 = 0.014$). On remarque aussi que le temps total dans la phase 1 correspond à 43% du temps de résolution des sous-problèmes même si la phase 1 n'est utilisée que dans 19% des itérations. La phase 1 est donc très lente par rapport à la phase 2 ce qui justifie l'utilisation d'une stratégie intermittente.

Notons aussi que des temps de résolution de 1 à 5 minutes pour les problèmes d'une durée d'un jour et de 20 minutes à une heure pour les problèmes d'une durée d'un mois sont jugés acceptables. Ainsi, les stratégies présentées, à l'exception des stratégies utilisant le modèle exact, ont des temps de résolution relativement acceptables considérant que l'on pourrait diviser par 2 où 3 les temps de calcul en utilisant une station de travail plus récente.

Tableau 3.11 – Répartition des temps de résolution

Scénario	Phase 1		Phase 2				
	Nb. It.	SP (s)	Nb. It.	SP (s)	SP Total (s)	PM Total (s)	Temps Total (s)
aa12-11-330-1350/	125	10	969	24	37	72	124
aa23-11/	151	7	846	11	27	28	67
aa4au16/	1825	80	8896	71	208	140	475
aa7-11/	114	5	653	6	17	19	44
aa25-11/	166	12	1267	28	46	103	166
aa-EBG1-oct/	4131	165	20746	144	462	261	1127
aa-EBG2-oct/	5212	380	27802	561	1040	2252	4030
aaEBG3-01-11-01/	350	24	2475	48	109	192	331
Total	12074	683	63654	893	1946	3067	6364

CHAPITRE 4 : INTÉGRATION À L'ISGAA

Ce chapitre explique la stratégie choisie pour inclure l'optimisation des pauses dans le module plus global qu'est le générateur de quarts/attributeur d'activités intégré (Integrated Shift Generator/Activity Assigner ou ISGAA), développé par Vatri (2001). Dans un premier temps, un bref rappel du modèle sous-tendant l'ISGAA sera présenté. Puis nous introduirons les différentes modifications nécessaires à la prise en charge des pauses lors de l'optimisation. Une généralisation de la méthode à deux phases, permettant la résolution de l'ISGAA avec pauses, sera alors présenté. Nous concluons ce chapitre par une présentation des résultats obtenues sur les scénarios présentés au chapitre précédent.

4.1 Définitions et notation concernant les quarts

Cette section introduit la notation permettant de décrire les différents types de quart auquel un employé peut travailler.

- Q , l'ensemble des types de quart.
- dq_{min}^q , la durée minimale du quart de type $q \in Q$.
- dq_{max}^q , la durée maximale du quart de type $q \in Q$.
- td_{min}^q , la durée totale minimale des pauses du quart de type $q \in Q$.
- td_{max}^q , la durée totale maximale des pauses du quart de type $q \in Q$.
- cb_{min}^q , la durée minimum de temps sans pause permis pour le quart de type $q \in Q$.
- cb_{max}^q , la durée maximum de temps sans pause permis pour le quart de type $q \in Q$.
- dq_{sup} , une borne supérieure sur la durée d'un quart.
- td_{sup} , une borne supérieure sur la durée totale des pauses dans un quart.

- cb_{sup} , une borne supérieure sur la durée maximum sans pause.

Un type de quart est caractérisé par un intervalle de durée et une fenêtre de temps à l'intérieur de laquelle le quart doit débiter. Cette fenêtre et cet intervalle permettent de classer les quarts en quarts de jour, de soir et de nuit. On définit un horaire comme étant l'ensemble des informations définissant le travail à effectuer par chaque employé durant l'horizon de planification. Ces informations spécifient pour chaque employé et chaque jour où il travaille, l'heure de début de son quart de travail, la durée de ce quart, les activités et les pauses remplissant ce quart ainsi que leur heure de début et leur durée. Dans ce chapitre, nous appellerons l'algorithme permettant de résoudre le problème d'assignation des activités à l'intérieur des quarts : l'attributeur d'activités (AA).

4.2 Présentation du problème et de l'ISGAA

Le problème que l'ISGAA résout est le suivant : pour un horizon d'optimisation donné, déterminer pour chaque employé et pour chaque journée qu'il doit travailler, le quart de travail qu'il doit accomplir et remplir ce quart d'activités et de pauses. Les horaires ainsi réalisés doivent principalement minimiser le nombre de sous-couvertures sur l'horizon d'optimisation. Comme dans le problème d'attribution des activités, l'horizon de planification est discrétisé en périodes de même durée. On connaît les jours de travail et de repos des employés de même que les patrons de quarts associés à chaque jour de travail. Un patron de quart limite l'ensemble des types de quart pouvant être affectés à un employé. Ces patrons sont associés à une partie spécifique de la journée (jour, soir ou nuit). Par exemple, un employé associé à un patron de quart de nuit pour une journée de travail ne pourra travailler que sur des quarts se déroulant la nuit pour cette journée. En plus de respecter les contraintes du problème

d'assignation des activités (voir chapitre 2 et 3), deux contraintes supplémentaires sont nécessaires pour la création des quarts. La première assure un repos minimal entre deux quarts consécutifs d'un même employé, la deuxième impose que la durée d'un quart soit comprise entre une valeur minimale et une valeur maximale (la flexibilité sur la durée dépend du type de quart).

La méthode générale de résolution est la même que pour l'AA, c'est-à-dire une méthode de génération de colonnes dont les sous-problèmes sont des problèmes de plus court chemin avec ressources. Pour diminuer la taille du problème, on utilise une stratégie de décomposition temporelle. Dans sa version la plus simple (pas d'objectifs secondaires), le problème maître est semblable à celui présenté au chapitre 2, à la différence qu'on y ajoute des contraintes visant à obtenir un nombre minimal de quarts pour chacun des types. En ce qui à trait aux sous-problèmes, ces derniers ont pour but de choisir, pour chaque employé, l'horaire de travail réalisable ayant le plus petit coût en regard aux valeurs courantes des variables duales du problème maître.

La figure 4.1 représente le réseau utilisé pour modéliser la construction d'un horaire pour un employé, dans les sous-problèmes. Il y a un sous-problème, et par conséquent un réseau, pour chaque bloc de jours consécutifs de travail de l'employé. Dans cette figure, la boîte appelée réseau de l'AA indique que les réseaux des sous-problèmes utilisés dans l'AA sont repris intégralement dans les réseaux de l'ISGAA. Les lignes constituée de traits représente un chemin dans le réseau de l'AA. Le réseau de l'ISGAA présente certaines différences avec celui de l'AA. La première est l'ajout de noeuds d'attribution de quart de travail (n_{Quart}) qui sont créés pour chaque période de la séquence de travail. La deuxième est la taille du réseau, alors que l'AA ne contenait que les noeuds associés aux temps correspondant au quart de travail de l'employé, le réseau de l'ISGAA contient un noeud de chaque type, pour chaque temps t de la séquence de travail. Par exemple, si le réseau représente un bloc de 3 jours consécutifs, il y a aura $3 * 1440/\delta$ noeuds de chaque type. Les arcs de type

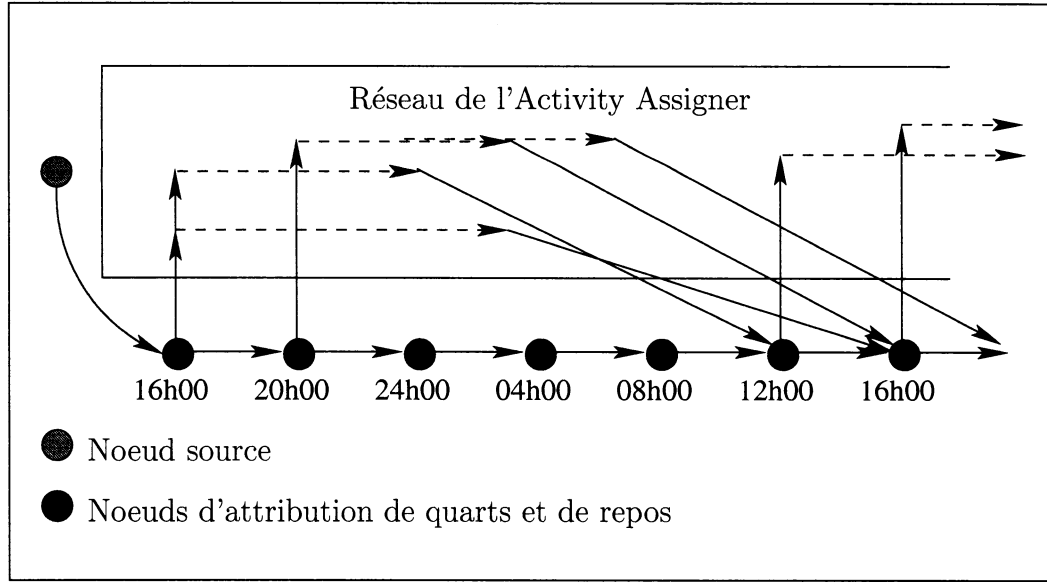


Figure 4.1 – Réseau utilisé par l'ISGAA

$aDebA$, $aAct$ et $aTrs$ sont créés tels que présentés à la section 3.3.1. Un arc de type $aDeb$ va du noeud source au premier noeud $nQuart$ et un arc de type $aFin$ part du dernier $nQuart$ pour aller au noeud puits. Le modèle réseau de l'ISGAA introduit trois nouveaux types d'arc : les arcs de début de quart ($aDebQ$) (chacun de ces arcs est associé à un quart $q \in Q$), les arcs de fin de quarts ($aFinQ$) et des arcs de repos. Pour chaque temps $t \in T$ pour lequel un employé peut commencer un quart de travail, un arc de type $aDebQ$ est créé allant du noeud d'attribution de quart de travail ($nQuart, t$) au noeud de transition ($nTrs, t$). Pour chaque temps $t \in T$ où un quart peut se terminer, on ajoute un arc de type $aFinQ$ allant du noeud ($nTrs, t$) au noeud ($nQuart, t + tr$), la longueur tr de l'arc correspondant au temps minimal de repos (si le noeud ($nQuart, t + tr$) n'appartient pas au réseau alors l'arc aboutit au dernier noeud de type $nQuart$).

Pour modéliser la durée minimale et maximale d'un quart, deux ressources $rDurQ_{Min}$ et $rDurQ_{Max}$ sont utilisées, toutes deux comptabilisent le temps passé à effec-

Tableau 4.1 – Consommation des ressources $rMinPauTotDur$ et $rMaxPauTotDur$

Type d'arc	$\varphi_{ij}^{k,rDurQMin}$	$\varphi_{ij}^{k,rDurQMax}$
$aDeb$	dq_{sup}	0
$aFin$	0	0
$aDebQ$	$-(dq_{sup} - dq_{min}^q)$	$dq_{sup} - dq_{max}^q$
$aFinQ$	dq_{sup}	$-dq_{sup}$
$aDebA$	$-d_{ij}$	d_{ij}
$aAct$	$-d_{ij}$	d_{ij}
$aBegB$	$-d_{ij}$	d_{ij}
$aBrk$	$-d_{ij}$	d_{ij}
$aTrs$	0	0

tuer le quart. Cependant, l'une permet d'assurer que la durée minimale est respectée, alors que l'autre veille au respect de la durée maximum. À chaque noeud du réseau, la fenêtre pour la ressource $rDurQMin$ est $[0, dq_{sup}]$ et celle pour $rDurQMax$ est $[0, dq_{sup}]$. La fonction de prolongation pour ces ressources est $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$. Les valeurs de $\varphi_{ij}^{k,rDurQMin}$ et $\varphi_{ij}^{k,rDurQMax}$ sont données au tableau 4.1.

4.3 Intégration des pauses dans le modèle de l'IS-GAA

Cette section a pour but de générer pour l'ISGAA un modèle permettant d'introduire l'assignation des pauses à l'ISGAA. Considérant que le modèle présenté a été conçu en modifiant le modèle de pauses exact pour l'utiliser dans l'ISGAA, nous l'appellerons le modèle de l'ISGAA exact. Pour inclure les pauses dans cette nouvelle modélisation, trois points sont à couvrir.

- Ajouter les arcs de type $aBegB$ et $aBrk$ tel que vu à la section 3.3.1.

- Ajouter les ressources $rDurPauMax$ tel que vu à la section 3.3.1.
- Modifier les ressources $rDurTotPauMin$, $rDurTotPauMax$, $rSansPauMin$ et $rSansPauMax$ de sorte qu'elles s'incluent au modèle.
- Ajouter une paire de ressources permettant d'inclure dans le modèle le fait que des types de quart différents possèdent des durées minimales/maximales sans pause différentes.

4.3.1 Ajout des ressources de pause existantes

Respect de la contrainte sur la durée individuelle des pauses

La contrainte sur la durée individuelle des pauses ne varie pas selon le type de quart effectué. Par conséquent, l'attribution des bornes sur les noeuds du réseau et la consommation des ressources sur les arcs sont identiques à ce qui a été présenté à la sous-section 3.3.1.

Respect de la contrainte sur la durée totale des pauses

La durée totale des pauses varie selon le type de quart à effectuer. Il est donc nécessaire de modifier quelque peu le modèle présenté à la section 3.3.1. À chaque noeud i du réseau, la fenêtre pour les ressources $rDurTotPauMin$ et $rDurTotPauMax$ est $[0, td_{sup}]$. On définit la durée totale des pauses en fonction du type de quart à effectuer en plaçant une consommation de ressources spécifique sur chaque arc de début de quart. La fonction de prolongation est la même pour les deux ressources $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$. Les valeurs de $\varphi_{ij}^{k,rDurTotPauMin}$ et $\varphi_{ij}^{k,rDurTotPauMax}$ sont données au tableau 4.2.

Tableau 4.2 – Consommation des ressources $rMinPauTotDur$ et $rMaxPauTotDur$

Type d'arc	$\varphi_{ij}^{k,rMinPauTotDur}$	$\varphi_{ij}^{k,rMaxPauTotDur}$
$aDeb$	td_{sup}	0
$aFin$	0	0
$aDebQ$	$-(td_{sup} - td_{min}^q)$	$td_{sup} - td_{max}^q$
$aFinQ$	td_{sup}	0
$aDebA$	0	0
$aAct$	0	0
$aBegB$	$-d_{ij}$	d_{ij}
$aBrk$	$-d_{ij}$	d_{ij}
$aTrs$	0	0

4.3.2 Respect de la contrainte sur la durée séparant deux pauses consécutives

Pour s'assurer que les durées séparant les pauses consécutives se trouvent dans la fourchette $[cb_{min}^k, cb_{max}^k]$, on utilise les ressources $rSansPauMin$ et $rSansPauMax$ présentées à la sous-section 3.2.1. Cependant les valeurs de cb_{min}^q et de cb_{max}^q varient selon le type de quarts que l'on effectue, il est donc nécessaire de trouver une méthode permettant d'évaluer ces valeurs selon l'arc de début de quart emprunter. Pour ce faire, on introduit deux nouvelles ressources : $rCbQmin$ et $rCbQmax$ qui vont garder respectivement les valeurs cb_{min}^q et cb_{max}^q selon le quart de type $q \in Q$ associé à l'arc de début de quart. À chaque noeud i de type $nOri$, $nDes$, $nTrs$ et $nAct$ du réseau, la fenêtre pour les ressources $rDurTotPauMin$ et $rDurTotPauMax$ est $[0, cb_{sup}]$. Pour les noeuds de type $nPau$ et $nQuart$, cette fenêtre est $[cb_{sup}, cb_{sup}]$ pour la ressource $rDurTotPauMin$ et $[0, cb_{sup}]$ pour la ressource $rDurTotPauMax$. Les fonctions de prolongation $f_{ij}^r(\Phi_i^{kr})$ pour les ressources $rSansPauMin$ et $rSansPauMax$ utiliseront les valeurs des ressources $rCbKmin$ et $rCbKmax$ pour réinitialiser leurs valeurs suite à un passage sur un arc de type $aBegB$. Pour ce faire, on doit changer la fonction

Tableau 4.3 – Intrants des fonctions de prolongation des ressources $rSansPauMin$, $rSansPauMax$, $rCbQmin$ et $rCbQmax$

Type d'arc	$\varphi_{ij}^{k,rSansPauMin}$	$\varphi_{ij}^{k,rSansPauMax}$	$\varphi_{ij}^{k,rCbQmin}$	$\varphi_{ij}^{k,rCbQmax}$
$aDeb$	cb_{sup}	0	0	0
$aFin$	0	0	0	0
$aDebQ$	0	$cb_{sup} - cb_{max}^q$	cb_{min}^q	cb_{max}^q
$aFinQ$	cb_{min}^q	0	0	0
$aDebA$	$-d_{ij}$	d_{ij}	0	0
$aAct$	$-d_{ij}$	d_{ij}	0	0
$aBegB$	$\Phi_i^{q,rCbQmin}$	$cb_{sup} - \Phi_i^{q,rCbQmax}$	0	0
$aBrk$	0	0	0	0
$aTrs$	0	0	0	0

de prolongation pour la ressource $rSansPauMax$, celle-ci est $f_{ij}^r(\Phi_i^{kr}) = \Phi_i^{kr} + \varphi_{ij}^{kr}$ sur tous les arcs à l'exception des arcs de type $aBegB$ où elle est $f_{ij}^r(\Phi_i^{kr}) = \varphi_{ij}^{kr}$. L'interaction entre ces deux paires de ressources est résumée au tableau 4.3. Ce mécanisme est nécessaire, car si l'on sait quel type de quart on effectue lorsqu'on passe sur arc de type $aDebQ$, cette information n'est plus disponible lorsqu'on traverse un arc de début de pause. Cette information est cruciale pour remettre le compteur à zéro lors du début d'une pause.

4.3.3 Performance de la méthode de résolution de l'ISGAA

La méthode de résolution de l'ISGAA développée par Vatri (2001) nous permet de résoudre le modèle que nous avons développé pour inclure les pauses. En effet, les modifications apportées aux sous-problèmes n'ont pas modifié leur structure et ces derniers peuvent être résolus par l'algorithme de résolution du problème de plus court chemin avec ressources utilisé par l'ISGAA sans pause. Cependant, le problème résultant est beaucoup trop lourd pour être résolu ainsi. Comme dans le cas du modèle de

pauses exact, l'introduction d'un nombre élevé de ressources et la nature contraire des ressources $rDurTotPauMin$, $rDurTotPauMax$ et des ressources $rSansPauMin$, $rSansPauMax$ font exploser les temps de résolution. Dans la section suivante, nous contournerons ce problème en généralisant la méthode à deux phases présentée au chapitre précédent.

4.4 Généralisation de la méthode à deux phases pour l'ISGAA

Le chapitre précédent a présenté une méthode permettant de résoudre les sous-problèmes en deux phases, la première établissant les ensembles de pauses des quarts et la deuxième venant y placer les différentes activités à effectuer, et ainsi construire un horaire réalisable. On se rappellera que selon la stratégie de résolution employée, les deux phases n'étaient pas nécessairement exécutées à chaque itération de génération de colonnes. Ainsi, dans la stratégie dite de génération intermittente des pauses par agrégation des activités, on n'exécutait la première phase que lorsqu'il était jugé que l'ensemble courant de pauses n'avait plus de potentiel. Cette section a pour but d'étendre cette stratégie à l'ISGAA. Dans ce cas, la première phase consistera à établir, pour le sous-problème associé à un employé, quel type de quart ce dernier doit effectuer, quand ce quart commence et finit, et enfin à déterminer les pauses qui le constitueront. La deuxième phase sera en tout point semblable à celle vu au chapitre 3 : ayant un quart déterminé avec pauses fixées il suffira d'y appliquer la partie du modèle de base visant les sous-problèmes.

4.4.1 Modélisation des sous-problèmes permettant la génération de quarts et de pauses par agrégation des activités

Le but de cette sous-section est de décrire le modèle utilisé pour la résolution de la première phase des sous-problèmes. Rappelons d'abord que dans le modèle de l'ISGAA avec pauses, il y a un sous-problème et, par conséquent, un réseau pour chaque séquence de travail d'un employé. Le but de ce réseau est de modéliser en un problème de plus court chemin avec ressources, l'ensemble des quarts, des activités et des pauses pour la séquence de travail de l'employé. Ceci permet de déterminer l'horaire de travail de plus petit coût pour cet employé, en vertu des valeurs courantes des variables duales du problème maître. Cependant, lors de la première phase, la nature exacte des activités à accomplir n'est pas importante puisqu'on cherche seulement à déterminer la nature du quart de travail que l'employé devra accomplir et la disposition des pauses à l'intérieur de ce quart. Nous modifions donc le réseau des sous-problèmes de manière à ce que ce dernier ne contienne qu'une seule activité. Pour obtenir le réseau d'un tel modèle, il suffit de remplacer la partie du réseau correspondant à l'AA par le réseau présenté à la section 3.3.2. Dans ce modèle, toutes les ressources concernant les pauses et les quarts de travail sont utilisées.

4.4.2 Modélisation des sous-problèmes permettant de remplir d'activités les quarts de travail avec pauses fixées

Lors de l'exécution de la deuxième phase, on connaît pour chaque employé les quarts qu'il devra effectuer et les pauses qui les composent. Il suffit alors d'assigner des activités à ces quarts de travail afin de générer des colonnes de coût réduit négatifs. Le réseau utilisé pour modéliser ce problème est celui de l'ISGAA avec pauses auxquels nous avons enlevé certains arcs. Les arcs qui sont retirés sont les suivants :

- Tous les arcs de début ($aDebQ$) et de fin de quart ($aFinQ$) qui ne correspondent pas aux quarts générés lors de la première phase.
- Tous les arcs de pauses ($aBrk$ et $aBegB$) qui ne correspondent pas aux pauses générés lors de la première phase.
- Tous les arcs de type $aAct$, $aDebA$, $aBrk$, $aBegB$, $aTrs$ dont les temps de début et de fin ne sont pas compris dans les intervalles de temps sur lesquels s’étirent les quarts générés lors de la première phase.

De plus, on retire du modèle de l’ISGAA avec pauses toutes les ressources concernant les pauses et les quarts. Les solutions du problème de plus court chemin avec ressources résultant donnent alors des horaires qui permettront d’enrichir le problème maître.

4.4.3 Stratégie utilisée avec la méthode à deux phases de l’ISGAA

Toutes les stratégies concernant la méthode à deux phases présentées à la sous-section 3.3.3 peuvent être utilisées avec l’ISGAA. Cependant, nous utiliserons seulement la génération intermittente des pauses et des quarts par agrégation des activités pour les raisons suivantes :

- La résolution de la phase 1 prend environ 8 fois plus de temps que la résolution de la phase 2 (voir tableau 4.6), ce qui rend la génération continue des pauses et des quarts très coûteuse.
- Le modèle de l’ISGAA exact est si lourd qu’il n’est pas possible de l’inclure dans une boucle de la méthode à deux phase, comme nous avons fait avec le modèle de pauses exact.
- Comme nous le verrons, l’inconvénient majeur de l’ISGAA avec pauses est la lenteur d’exécution. Comme l’utilisation de la méthode de perturbation des pauses ajoutent beaucoup au temps de calcul, nous ne l’avons pas intégré à l’ISGAA.

Les critères utilisés pour juger de la qualité des quarts et des pauses générés lors de la première phase sont exactement les mêmes que ceux présentés à la sous-section 3.3.4.

Voici donc l'algorithme de génération intermittente des pauses et des quarts par agrégation des activités pour l'ISGAA.

Algorithme 4.1 Algorithme pour la génération intermittente des pauses et des quarts par agrégation des activités

```

for all itération de génération de colonnes do
  Résoudre le problème maître
  for all Séquence de travail d'un employé do
    % Sous-problèmes %
    if Pauses et quarts sans potentiel ou première itération then
      % Phase 1 %
      - Générer le sous-problème correspondant selon le
        modèle de génération des pauses et des quarts
        par agrégation des activités
      - Résoudre et garder les pauses et les quarts obtenus
        en mémoire pour les prochaines itérations
    % Phase 2 %
    - Générer le sous-problème selon le modèle de l'ISGAA exact
    - En utilisant les pauses et les quarts obtenus lors de la
      phase 1, enlever tous les arcs non nécessaires
    - Enlever les ressources concernant les quarts et les pauses
    - Résoudre le problème de plus court chemin avec ressources
      correspondant

```

4.4.4 Vision à moyen terme

La planification des quarts de travail fonctionne d'autant mieux que la fenêtre d'optimisation est grande. Cependant, pour obtenir des temps de calcul raisonnables, les fenêtres d'optimisation doivent être relativement courtes (entre 3 et 5 heures). Cet état de fait mène en pratique à de sérieux problèmes. Par exemple, dans de

nombreux problèmes, il existe un certain nombre de quarts de nuit. Dans ce cas, il n'est pas rare que le processus de résolution, lorsqu'il traite la fenêtre correspondant aux quarts de nuit, ait déjà assigné des quarts à tous les employés pour les heures précédentes, laissant ainsi des sous-couvertures à toutes les activités de nuit. Pour pallier à ce problème, nous avons développé une stratégie de vision à moyen terme. L'idée est de diviser la fenêtre d'optimisation en deux morceaux : le premier morceau $FOI \subseteq T$, que nous nommerons fenêtre d'optimisation à demande individuelle, établit la demande pour chaque activité individuellement, tandis que le deuxième morceau $FOA \subseteq T$, la fenêtre d'optimisation à demande agrégée, détermine la demande comme s'il s'agissait d'un problème mono-activité. Ainsi la demande au temps $t \in FTA$ sera égal à $\sum_{a \in A} Dem_{a,t}$ et chaque activité de couverture aura une contribution d'une unité à cette demande. Pour que cette vision à moyen terme fonctionne de façon appropriée, les fenêtres d'optimisation sont construites de sorte que la somme des durées des fenêtres FOI et FOA soit 12 heures. Puisque la partie mono-activité est beaucoup plus simple à résoudre que la partie multi-activités, cette stratégie nous permet d'assigner efficacement les quarts sans que le temps de calcul en souffre trop. Il faut cependant noter que, si la partie mono-activité allège les temps de calcul dans la deuxième phase des sous-problèmes, il n'en est rien pour la première phase qui ne contient elle qu'une seule activité agrégée. Quoique la résolution de la première phase soit plus longue que la deuxième, elle est exécutée moins souvent en vertu de la stratégie de génération intermittante des quarts et des pauses. Ces deux phénomènes contraires permettent garder des temps de résolution acceptables. La progression des fenêtres d'optimisation à l'intérieur de l'horizon d'optimisation est illustrée à la figure 4.2, où la durée de la fenêtre d'optimisation à demande individuelle est de 3 heures, celle du chevauchement 1 heure et la durée de la fenêtre d'optimisation à demande agrégée est de 9 heures.

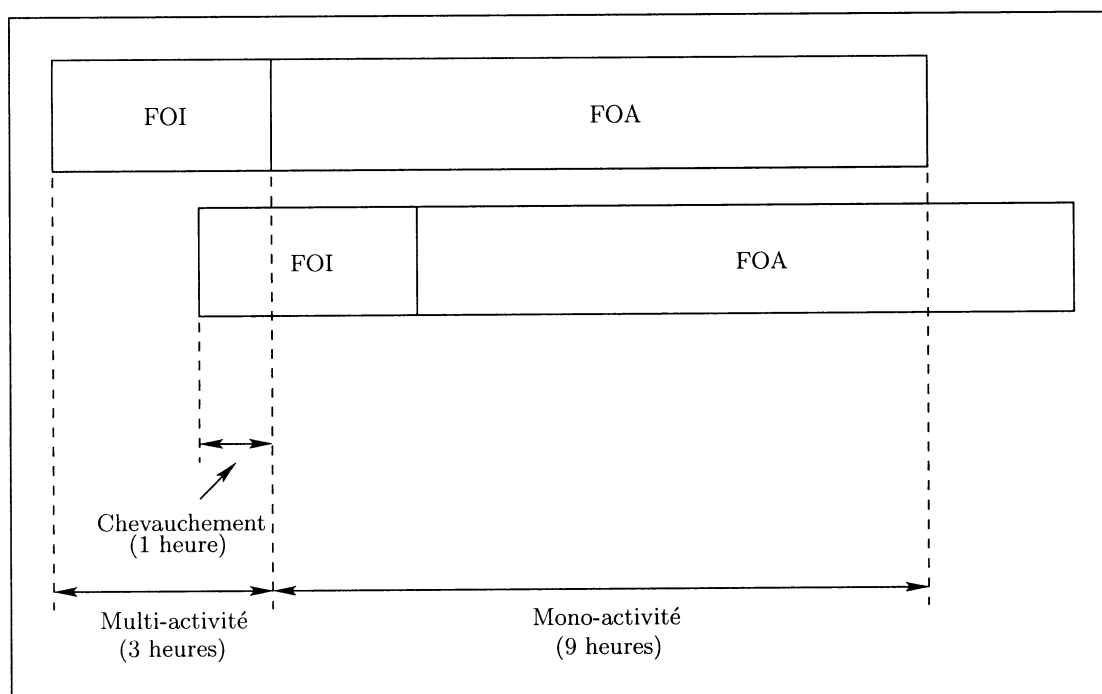


Figure 4.2 – Fenêtres d’optimisation dans la vision à moyen terme

4.5 Résultats numériques

L’efficacité de la méthode développée est évaluée en résolvant les mêmes problèmes qu’au chapitre 3. L’analyse des performances considère toujours les deux mêmes paramètres : le nombre de sous-couvertures et le temps de calcul. Les résultats ont été comparés à ceux obtenus avec la méthode de perturbation des pauses initiales, pour laquelle l’ensemble de pauses initiales avait été généré par un module spécifique. Deux tableaux sont présentés. Dans le premier, les deux méthodes sont comparées en utilisant des fenêtres de temps comparables, soit une fenêtre de temps de 180 minutes avec un chevauchement de 45 minutes. Le deuxième tableau fait une comparaison pour des temps de calcul à peu près similaires pour un même problème. Pour ce faire, les fenêtres de temps pour l’ISGAA avec optimisation des pauses sont les mêmes

Tableau 4.4 – Résultats de l'ISGAA pour les mêmes fenêtres de temps

Scénario	AA		ISGAA		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	3	0	25	-100	8.33
aa23-11/	11	1	1	17	-91	17.00
aa4au16/	19	14	24	65	26	4.64
aa7-11/	9	1	1	25	-88	25.00
aa25-11/	1	2	1	13	0	6.5
aa-EBG1-oct/	161	31	127	346	-21	10.66
aaEBG3-01-11-01/	7	3	2	32	-71	10.67
Total	217	55	156	523	-28	9.51

qu'au tableau précédent tandis que celles pour la méthode de perturbation des pauses initiales sont de 600 minutes pour un chevauchement de 300 minutes.

Le tableau 4.4 montre clairement l'efficacité de la méthode : nous avons obtenu une diminution de 38% des sous-couvertures par rapport à la méthode de perturbation des pauses initiales. Cependant, il est important de noter qu'il n'est pas tout à fait juste de comparer ces résultats. En effet, les problèmes résolus par les deux méthodes ne sont pas tout à fait identiques. L'AA prend en entrée des quarts déjà déterminés qui contiennent certaines caractéristiques qu'il est impossible de prendre en compte avec l'ISGAA pour le moment. Par exemple, dans certains quarts, des employés doivent s'absenter pour une partie de celui-ci, pouvant ainsi laisser quelques sous-couvertures lors de leur départ. Ces absences ne peuvent être prises en compte par l'ISGAA car il n'a pas cette information et, de plus, il y a de forte chance que l'ISGAA assigne l'employé à un quart n'ayant aucune intersection avec la période d'absence. Ainsi, une partie difficile à évaluer de cette amélioration peut être due à ces différences. Cependant nous croyons qu'une partie non négligeable de cette amélioration est due à une meilleure planification des pauses et des quarts. Cette opinion vient du fait que,

Tableau 4.5 – Résultats de l’ISGAA pour des temps de résolution de même ordre de grandeur

Scénario	AA		ISGAA		Comparaison	
	SC	Temps (min)	SC	Temps (min)	Diff SC (%)	Ratio temps
aa12-11-330-1350/	9	32	0	25	-100	0,78
aa23-11/	5	42	1	17	-80	0,40
aa4au16/	25	157	24	65	-4	0,41
aa7-11/	5	33	1	25	-80	0,75
aa25-11/	2	30	1	13	-50	0,43
aa-EBG1-oct/	154	558	127	346	-18	0,62
aaEBG3-01-11-01/	6	41	2	32	-67	0,78
Total	206	893	156	523	-24	0,59

sur de petits problèmes où il est facile de vérifier si ces différences interviennent ou pas, nous avons remarqué que l’ISGAA améliorerait la qualité des solutions même en l’absence de ces différences. Toutefois, cette amélioration a été gagnée à un fort prix : un temps de calcul presque 10 fois plus important.

Le tableau 4.5 a pour but de vérifier si, pour des temps de calcul de même ordre de grandeur, l’ISGAA donne toujours de bons résultats si on le compare aux solutions obtenues par l’AA. Pour ce faire, nous avons agrandi les fenêtres d’optimisation de l’AA. Il nous a semblé que c’était la meilleure façon de faire profiter l’AA de beaucoup plus de temps de calcul. Encore là, les résultats sont éloquents : une amélioration de 24% des sous-couvertures pour 60% du temps de calcul de l’AA. Quoique ces résultats ne soient pas décisifs, ils permettent quand même de conclure que l’ISGAA est un outil prometteur pour obtenir de bonnes solutions dans les cas où l’on dispose du temps de calcul nécessaire.

4.6 Conclusion

Ce chapitre visait à intégrer le placement des pauses à l'approche de résolution du PFHQ présentée par Vatri (2001). La méthode présentée a permis d'atteindre cet objectif. Lorsqu'on compare les résultats obtenus avec l'approche existante qui décompose le problème en quatre étapes (construction de cycles, construction de quarts, assignation des quarts et assignation des activités) et utilise la méthode de perturbation des pauses lors de l'assignation des activités, aux résultats fournis par notre approche qui intègre les trois dernières étapes, on constate une diminution significative du nombre de sous-couvertures. Comme on pouvait s'y attendre, la résolution simultanée des trois dernières étapes du problème augmente considérablement les temps de résolution, ces temps restant cependant acceptables.

Les statistiques présentées au tableau 4.6, qui est similaire à celui présenté à la section 3.5, ont été obtenues en résolvant les scénarii avec la méthode présentée dans ce chapitre. On remarque que temps moyen par itération pour résoudre les sous-problèmes à la phase 1 ($14054/17891 = 0.79$) est huit fois plus élevé que celui pris par la phase 2 ($11427/122553 = 0.09$). Le temps total passé dans la phase 1 est encore plus important qu'avec la méthode présentée au chapitre 3, soit 55% du temps de résolution des sous-problèmes et ce même si la phase 1 n'est invoquée, dans le cas présent, que dans 15% des itérations. Finalement, on peut noter que la majeure partie du temps total de résolution est consacrée à la résolution des sous-problèmes.

Tableau 4.6 – Répartition des temps de résolution pour l'ISGAA

Scénario	Phase 1		Phase 2				
	Nb. It.	SP (s)	Nb. It.	SP (s)	SP Total (s)	PM Total (s)	Temps Total (s)
aa12-11-330-1350/	420	425	5468	740	1141	142	1395
aa23-11/	387	334	4006	487	799	100	993
aa4au16/	3251	1557	21396	1529	2939	425	3940
aa7-11/	438	391	6446	869	1230	167	1525
aa25-11/	335	299	4083	463	741	98	924
aa-EBG1-oct/	12538	10660	73297	6123	15281	1686	20734
aaEBG3-01-11-01/	522	388	7857	1216	1567	183	1906
Total	17891	14054	122553	11427	23698	2801	31417

CONCLUSION

Ce mémoire a porté sur le problème de fabrication des horaires de travail des contrôleurs aériens. Ces problèmes sont difficiles à résoudre car ils sont soumis à de nombreuses contraintes et sont définis sur une longue période (horizon continu d'un mois). Partant d'une approche de résolution négligeant le placement des pauses, l'objectif était de modifier cette approche pour y inclure cet aspect du problème.

Nous avons d'abord présenté un modèle et des techniques permettant de résoudre une partie du problème, soit le problème d'attribution des activités aux employés lorsque que leurs quarts de travail et les pauses les constituant sont déterminés. Le modèle présenté permettait de résoudre le problème à l'aide d'une méthode heuristique de séparation et évaluation progressive intégrant une approche de génération de colonnes combinée à une stratégie de décomposition temporelle. Des modifications à ce modèle ont alors été apportées dans le but d'y inclure le placement des pauses. Une méthode de résolution décomposant en deux phases la résolution des sous-problèmes de l'approche de génération de colonnes a ainsi pu être développée pour effectuer le placement des activités et des pauses dans des quarts de travail prédéterminés. Cette méthode a été testée par des expériences numériques sur divers scénarii de planification d'horaires provenant de problèmes pratiques fournis par la compagnie Technologies Ad Opt inc. Pour chaque scénario, différentes stratégies de résolution basées sur la méthode en deux phases ont été testées afin de mettre en lumière leurs forces et faiblesses. Ces expériences ont permis de montrer la capacité de cette méthode à résoudre efficacement, en comparaison au logiciel existant, les problèmes posés et d'exprimer clairement le compromis possible entre qualité des solutions et temps de résolution.

Nous avons intégré cette méthode de résolution à un module plus général permettant

la résolution du PFHQ lorsque les jours de travail des employés sont connus. Des tests ont été effectués en utilisant les mêmes scénarii de planification d'horaires pour comparer l'efficacité de l'approche intégrée de résolution à une approche existante séparant le problème en quatre étapes (construction de cycles, construction de quarts, assignation des quarts et assignation des activités) utilisant la méthode de perturbation des pauses lors de l'assignation des activités. Si on compare les résultats obtenus par l'approche existante en quatre étapes aux résultats obtenus avec notre approche, on remarque deux choses : 1. notre approche permet de trouver de meilleures solutions, 2. les temps de résolution de notre approche sont plus élevés, étant donné qu'on intègre les trois dernières étapes, mais demeurent néanmoins raisonnables.

Notons qu'il reste à inclure une fonctionnalité à notre approche, soit la prise en considération des stagiaires et des instructeurs. Cet aspect du problème permet de placer dans l'horaire de certains employés (les stagiaires) des périodes d'entraînement leur permettant d'obtenir les compétences nécessaires à l'exécution de certaines tâches. Ces périodes d'entraînement doivent correspondre à celle d'un instructeur habilité à former des employés pour la tâche en question.

L'inconvénient majeur de l'approche de résolution présentée étant sa lenteur de calcul, des travaux ultérieurs pourraient être effectués afin d'accélérer sa résolution. Ces investigations pourraient être de deux sortes : la première viserait à garder intacte la modélisation en s'attaquant aux techniques de résolution, alors que la deuxième étudierait une approche différente pour l'obtention des quarts et des pauses lors de la première phase. Des travaux en cours au GERAD, un doctorat (M. Rekik) et une maîtrise (A. Baba-Hadji) permettraient de résoudre simultanément tous les sous-problèmes de la première phase par un PLNE. D'autres travaux seraient aussi possibles afin de tester l'approche dans le contexte multi-activité où chaque quart est complètement mono-activité ou mono-activité entre deux pauses.

BIBLIOGRAPHIE

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.

BABA-HADJI, A. (2004). *Extension des modèles de construction de quarts de travail au cas de fenêtres de pause multiples*. Mémoire de maîtrise, École Polytechnique, Montréal, Canada.

BAILEY, J. et FIELDS, J. (1985). Personnel scheduling with flexshift models. *Journal of Operations Management* **5**, 327-338.

BARTHOLDI, J.J. (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research* **29**, 501-510.

BECHTOLD, S.E. et JACOBS, L.W. (1990). Implicit modeling of flexible break assignments in optimal shift scheduling. *Management Science* **44**, 534-547.

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.

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. (1998). Personnel tour scheduling when starting-time restrictions are present. *Management Science* **44**, 534-547.

BURNS, R.N. et CARTER, M.W. (1985). Work force size and single shift schedules with variable demands. *Management Science* **31**, 599-607.

BURNS, R.N. et KOOP, G.J. (1987). A modular approach to optimal multiple-shift manpower scheduling. *Operations Research* **35**, 100-110.

DANTZIG, G.B. (1954). A comment on Edie's traffic delays at toll booths. *Operations Research* **2**, 339-341.

DESAULNIERS, G., DESROSIERS, J., IOAHIM I., SOLOMON, M.M., SOUMIS, F. et VILLENEUVE, D. (1992). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. *Fleet Management and Logistics*, T.G. Crainic et G. Laporte (eds), Kluwer, Norwell, MA, 57-94.

DESROCHERS, M. et SOUMIS, F. (1988). A generalised permanent labeling algorithm for the shortest path problem with time windows. *INFOR* **26**, 191-212.

- DESROCHERS, M., GILBERT, J., SAUVÉ M. et SOUMIS, F. (1992). CREW-OPT : Subproblem modeling in a column generation approach to urban crew scheduling. *Computer-Aided Transit Scheduling, Lectures Notes in Economics and Mathematical Systems 386*, M. Desrochers et J.-M. Rousseau (eds.), Springer Verlag, Berlin, 395-406.
- DESROSIERS, J., DUMAS, Y., SOLOMON, M.M. et SOUMIS, F. (1995). Time constrained routing and scheduling, Network Routing, M.O. Ball *et al.* (eds), *Handbooks in Operations Research and Management Science* 8, 35-139.
- EASTON, F.F. et ROSSIN, D.F. (1991). Sufficient working subsets for the tour scheduling problem. *Management Science* **37**, 1441-1451.
- EMMONS, H. et BURNS, R.N. (1991). Off-day scheduling with hierarchical worker categories. *Operations Research* **39**, 484-496.
- GAMACHE, M., SOUMIS, F., MARQUIS, G. et DESROSIERS J. (1999). A column generation approach for large-scale aircrew rostering problems. *Operations Research* **47**, 247-263.
- HAASE, K. (1999). Retail business staff scheduling under complex labor regulations. Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel Nr. 511, Kiel, Allemagne.
- HENDERSON, W.B. et BERRY, W.B. (1979). Heuristic methods for telephone operator shift scheduling : an experimental analysis. *Management Science* **22**, 1372-1380.
- JARRAH, A.I.Z., BARD, J.F. et DESILVA, A.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.A. et WATTS, C.A. (1982). A simulation analysis of tour-shift scheduling construction procedures. *Management Science* **28**, 520-532.

MOONDRA S.L. (1976). An L.P. model for work force scheduling in banks. *Journal of Bank Research* **6**, 299-301.

OMARI, Z. (2002). *Attribution des activités aux employés travaillant sur des quarts*. Mémoire de maîtrise, École Polytechnique, Montréal, Canada.

REKIK, M., CORDEAU, J.F. et SOUMIS, F. (2002). Using benders decomposition to implicitly model tour scheduling. Rapport technique, Les Cahiers du Gerad G-2002-32, HEC Montréal, Canada.

REKIK, M., CORDEAU, J.F. et SOUMIS, F. (2003). Implicit shift scheduling with multiple breaks and work stretch duration constraints. Document de travail, École Polytechnique, Montréal, Canada.

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.

VATRI, E. (2001). *Intégration de la génération de quarts de travail et de l'attribution d'activités*. Mémoire de maîtrise, École Polytechnique, Montréal, Canada.