Titre: Title:	Ré-optimisation de l'horaire de travail d'employés en surtemps
Auteur: Author:	Cherifa Saadi
Date:	2018
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Saadi, C. (2018). Ré-optimisation de l'horaire de travail d'employés en surtemps [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/3044/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/3044/
Directeurs de recherche: Advisors:	Guy Desaulniers
Programme: Program:	Maîtrise recherche en mathématiques appliquées

UNIVERSITÉ DE MONTRÉAL

RÉ-OPTIMISATION DE L'HORAIRE DE TRAVAIL D'EMPLOYÉS EN SURTEMPS

CHERIFA SAADI
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)
AVRIL 2018

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

RÉ-OPTIMISATION DE L'HORAIRE DE TRAVAIL D'EMPLOYÉS EN SURTEMPS

présenté par : SAADI Cherifa
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. SOUMIS François, Ph. D., président
M. DESAULNIERS Guy, Ph. D., membre et directeur de recherche
M. ROUSSEAU Louis-Martin, Ph. D., membre

DÉDICACE

À Dieu le tout miséricordieux.

À celui qui m'a indiqué la bonne voie en me rappelant que la volonté est la clé de la réussite, mon père.

À celle qui a attendu avec patience les fruits de sa bonne éducation, ma mère.

À mes frères et sœurs qui je le sais ma réussite est très importante pour vous.

À mon fiancé, ton amour et ta sollicitude à mon égard me marqueront à jamais.

Que Dieu vous paye pour tous vos bienfaits.

À tous mes amis et tous ceux qui me sont chers...

À tous ceux qui ne cessent de m'encourager.

À tous ceux qui me font confiance.

REMERCIEMENTS

Je réserve ces lignes en signe de gratitude et de reconnaissance à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce travail.

Je tiens à témoigner ma gratitude et mes vifs remerciements à mon encadrant M. Guy Desaulniers, directeur du centre de recherche GERAD et professeur à l'École Polytechnique Montréal, pour avoir accepté de m'accueillir au GERAD sous sa direction, pour m'avoir permis de travailler sur ce projet et pour son soutien régulier. C'est grâce à son aide, sa disponibilité, sa gentillesse et ses conseils précieux que ce soit pour me donner des idées ou pour les relectures que j'ai pu achever ce travail.
Je souhaite remercier François Lessard et Reinhard Bürgy pour leur aide dans la partie programmation tout au long de ce projet, François Soumis et Louis-Martin Rousseau qui ont accepté de faire partie de mon jury.
Je remercie également le Conseil de recherche en sciences naturelles et génie du Canada ainsi que l'entreprise KRONOS qui ont participé à mon financement.

On n'a pas souvent l'occasion de remercier ses parents dans une vie, alors je suis contente que l'occasion m'en soit donnée ici. Merci maman ! Merci papa ! Je vous remercie tout au fond de mon cœur d'avoir cru en moi. Je vous remercie pour votre présence dans ma vie. Vous serez toujours la lueur de bonheur dans ma vie. Merci pour votre confiance, gentillesse et bonté. Je vous aime.

Mes remerciements sont aussi adressés à mon fiancé Anis pour son soutien et ses encouragements.

Finalement, je remercie tous mes amis de Montréal pour tous les moments de complicité partagés.

RÉSUMÉ

L'objectif principal de chaque entreprise est de maximiser ses profits ou d'offrir un service de qualité à moindre coût. L'entreprise cherche alors à générer des horaires de travail de ses employés qui couvrent la demande de sa clientèle tout en respectant les règlementations et les conventions collectives. L'entreprise cherche aussi à générer des horaires de travail qui couvrent la demande de sa clientèle tout en offrant un service de qualité. Ainsi, les problèmes de construction d'horaires de personnel ont été étudiés depuis plusieurs décennies en recherche opérationnelle.

Suite à une perturbation de la demande, il est possible que l'horaire généré une semaine plus tôt avec une certaine prévision de demande ne soit plus valable pour la nouvelle demande observée le jour même. Par exemple, un changement climatique non prévu peut diminuer considérablement la demande. Dans le sens contraire, une demande plus élevée que celle prévue au début de la semaine nécessite plus de quarts de travail que ceux déjà planifiés pour la couvrir. Dans les deux cas, il faut mettre à jour l'horaire pour répondre au mieux à cette demande. Cette mise à jour est souvent effectuée en temps réel et peut faire en sorte que le quart d'un employé est allongé ce qui l'amène en surtemps, augmentant ainsi son salaire horaire pour le temps supplémentaire accumulé tout au long d'une semaine. En fin de journée, lorsque le gérant dispose de plus de temps pour revoir les horaires, une ré-optimisation des horaires peut être faite pour éviter le temps supplémentaire engendré en cours de journée.

Ce projet de maîtrise répond à ce problème de ré-optimisation de l'horaire de personnel qui tombe en surtemps. Nous nous plaçons dans le domaine de vente au détail où l'horaire est non continu. Pour résoudre le problème, nous proposons une méthode exacte basée sur un modèle linéaire en nombres entiers et deux méthodes heuristiques en deux phases qui consistent à résoudre un programme linéaire en nombres entiers de taille réduite en limitant le nombre de quarts de travail que nous générons et ainsi accélérer le temps de résolution. Pour chaque méthode de résolution, nous introduisons deux approches. Chaque approche est basée sur la résolution d'un ou plusieurs programmes linéaires en nombres entiers. La première est une approche séquentielle dans laquelle les employés en surtemps sont traités l'un après l'autre et la deuxième est une approche simultanée où les employés en surtemps sont traités tous à la fois.
Nous présentons ensuite différents types de transformations possibles pour les quarts. Avec ces transformations effectuées sur les quarts d'un horaire initial, nous générons des nouveaux

quarts modifiés qu'on appelle quarts proposés. Enfin, nous définissons une structure de coût adaptée à la ré-optimisation. Cette structure pénalise toute modification de l'horaire.
L'analyse des solutions obtenues après avoir effectué des tests sur des jeux de données réels montrent qu'on est arrivé à enlever le surtemps et améliorer les coûts des solutions initiales. En outre, l'utilisation des deux heuristiques accélère le temps de calcul mais dégrade légèrement la qualité de la solution.

ABSTRACT

Each company is looking to generate work schedules that cover the demand of its customers while offering a service of quality and being sure to enforce the regulations and collective agreements.
However, due to some unpredictable demand variation, it is possible that the schedule planned a week ahead with some demand forecast is no longer valid for the new demand observed the same day. For example, a higher demand than scheduled a week earlier requires more shifts than those already planned. In this case, the schedule needs to be updated to meet this request. This update is often done in real time by extending some employees' shifts which brings them into overtime, thus increasing their salary for additional time accumulated throughout a week. At the end of the day, a schedule re-optimization can be made to avoid the extra time generated during the day.

This master's thesis is responding to this shift rescheduling problem. We deal with a non-continuous retail environment. To solve the problem, we propose an exact method based on an integer programming model and two two-phase heuristic methods that consist in solving a reduced-sized integer linear program by limiting the number of generated shifts and thus reduce the computational time. For each solution method, we introduce two approaches. Each approach is based on solving one or more linear integer programs. The first is a sequential approach in which employees with overtime are treated one after other and the second is a simultaneous approach where these employees are treated all at once.

We then present different types of possible shift transformations to generate new modified ones called proposed shifts. Finally, we define a cost structure adapted to the re-optimization. This structure penalizes every schedule modification.

After performing tests on real datasets, we analyze the computed solutions to show that we have succeeded in removing overtime and improving the initial solutions costs. In addition, we observe that both heuristics speed up the computation time but slightly degrades the solution quality.

TABLE DES MATIÈRES

DÉDICACE iii

REMERCIEMENTS iv

RÉSUMÉ v

ABSTRACT vii

TABLE DES MATIÈRES viii

LISTE DES TABLEAUX x

LISTE DES FIGURES xi

LISTE DES SIGLES ET ABRÉVIATIONS xii

LISTE DES ANNEXES xiii

CHAPITRE 1 INTRODUCTION 1
- 1.1 Contexte de l'étude 1
- 1.2 Motivation 4
- 1.3 Problématique et objectifs 5
- 1.4 Plan de mémoire 6

CHAPITRE 2 REVUE DE LITTÉRATURE 7
- 2.1 Génération d'horaires de personnel 7
- 2.2 Mise à jour des horaires 9

CHAPITRE 3 MODÈLE MATHÉMATIQUE ET PROPOSITION DES QUARTS 11
- 3.1 Modèle mathématique 11
  - 3.1.1 Idée générale du modèle 11
  - 3.1.2 Structure des coûts et gestion des contraintes 12
  - 3.1.3 Modèle mathématique 14
- 3.2 Génération des quarts transformés 17
  - 3.2.1 Types de transformations 17
  - 3.2.2 Propositions des quarts 18

3.2.3 Contraintes et paramètres . . . 22
3.3 Coûts et pénalités . . . 23
3.3.1 Pénalités de modification . . . 24
3.3.2 Tableau récapitulatif . . . 25

CHAPITRE 4 APPROCHES ET MÉTHODES DE RÉSOLUTION . . . 26
4.1 Approches de résolution . . . 26
4.1.1 Approche séquentielle . . . 26
4.1.2 Approche simultanée . . . 26
4.2 Méthodes de résolution . . . 29
4.2.1 Méthode exacte ME . . . 29
4.2.2 Méthodes heuristiques . . . 29
4.3 Gestion du nombre d'employés non en surtemps avec horaire modifié . . . 32

CHAPITRE 5 RÉSULTATS NUMÉRIQUES . . . 33
5.1 Ensemble de scénarios et données . . . 33
5.1.1 Les contextes . . . 33
5.1.2 Les instances . . . 34
5.1.3 Données . . . 34
5.2 Cas extrêmes . . . 36
5.2.1 Ré-optimisation exacte . . . 36
5.2.2 Fixation de l'horaire initial avec le surtemps . . . 36
5.2.3 Impact des pénalités de modification . . . 37
5.3 Résultats . . . 37
5.3.1 Coûts . . . 38
5.3.2 Temps de résolution . . . 42
5.3.3 Transformation des quarts . . . 48
5.4 Conclusion . . . 49

CHAPITRE 6 CONCLUSION . . . 50
6.1 Synthèse des travaux . . . 50
6.2 Limitations de la solution proposée . . . 51
6.3 Améliorations futures . . . 52

RÉFÉRENCES . . . 53

ANNEXES . . . 55

LISTE DES TABLEAUX

Tableau 3.1	Paramètres par défaut des transformations pour un employé $e \in ES$	23
Tableau 3.2	Tableau récapitulatif	25
Tableau 5.1	Informations sur les solutions de base	35
Tableau 5.2	Résultats des méthodes $exact$ et $fixed$ pour les jeux de données	37
Tableau 5.3	Coûts des contextes 1 et 2 (275 employés)	38
Tableau 5.4	Coûts des contextes 3 et 4 (275 employés)	39
Tableau 5.5	Coûts des contextes 1 et 2 (47 employés)	39
Tableau 5.6	Coûts des contextes 3 et 4 (47 employés)	40
Tableau 5.7	Temps de résolution en secondes des contextes 1 et 2 (275 employés)	43
Tableau 5.8	Temps de résolution en secondes des contextes 3 et 4 (275 employés)	43
Tableau 5.9	Temps de résolution en secondes des contextes 1 et 2 (47 employés)	44
Tableau 5.10	Temps de résolution en secondes des contextes 3 et 4 (47 employés)	44
Tableau 5.11	Transformation de quarts (275 employés)	48
Tableau 5.12	Transformation de quarts (47 employés)	48

LISTE DES FIGURES

Figure 3.1 Fonction des coûts linéaire par morceau 13
Figure 3.2 Proposition de quarts 20
Figure 3.3 Algorithme de génération des quarts anonymes 22
Figure 5.1 Comparaison des coûts des méthodes de résolution pour 275 employés 41
Figure 5.2 Comparaison des coûts des méthodes de résolution pour 47 employés 42
Figure 5.3 Comparaison de temps de calcul des méthodes de résolution 275 employés 45
Figure 5.4 Comparaison de temps de calcul des méthodes de résolution 47 employés 45
Figure 5.5 Quarts proposés versus temps de résolution (275 employés) 47
Figure 5.6 Quarts proposés versus temps de résolution (47 employés) 47

LISTE DES SIGLES ET ABRÉVIATIONS

OEP	Overtime Employee Problem
ME	Méhode exacte
MH1	Méthode heuristique 1
MH2	Méthode heuristique 2

LISTE DES ANNEXES

ANNEXE A STRUCTURE DE COÛTS 55

CHAPITRE 1 INTRODUCTION

La confection de bons horaires de travail est un problème qui consiste à produire des horaires répondant aux besoins de l'entreprise tout en tenant compte des contraintes imposées par la législation et les conventions collectives. Ce problème suscite toujours beaucoup d'intérêt car la compétitivité des entreprises repose essentiellement sur leur capacité à bien exploiter les ressources dont elles disposent afin de satisfaire au mieux la demande de leurs clients.
Dans notre cas, nous nous intéressons à la gestion efficace du personnel qui est primordiale pour de nombreuses entreprises à savoir la production d'un bon horaire adapté qui respecte les contraintes et qui répond à la demande.
Dans la littérature en recherche opérationnelle, les problèmes de planification de personnel ont été étudiés depuis plusieurs décennies. En effet, ils apparaissent dans des secteurs variés et touchent un grand nombre d'organisations que ce soit dans le domaine de la vente au détail, du transport (aérien, transport en commun, ...), de la santé (hôpitaux, ambulances, ...).
Dans ce mémoire, nous nous plaçons dans le domaine de la vente au détail qui diffère largement des autres domaines parce qu'il y a beaucoup de flexibilité dans les quarts que peuvent travailler les employés dû à des fluctuations possiblement fréquentes de la demande en personnel.

1.1 Contexte de l'étude

La planification d'horaires de personnel est une tâche compliquée. Cette planification se fait, en général, sur un horizon d'une semaine qui est divisée en périodes de 15 minutes. En outre, il y a différents types de travail appelés des activités à effectuer nécessitant chacun des qualifications spécifiques.
L'entreprise cherche à minimiser le coût de fonctionnement lié à la main d'œuvre tout en prenant en compte l'ensemble des contraintes suivantes :

— **La satisfaction de la demande :**La demande est exprimée en nombre d'employés par activité et pour chaque période de l'horizon. S'il n'y a pas un nombre d'employés suffisant pour répondre à la demande en un temps raisonnable, les clients seront insatisfaits et l'entreprise risque alors de perdre temporairement sa clientèle si ces personnes ne souhaitent pas attendre d'être servies et quittent sans bénéficier du service offert ou encore à long terme s'ils décident de changer de fournisseur. Inversement, s'il y a trop d'employés, l'entreprise aurait pu garder sa qualité de service en mobilisant moins d'employés. L'entreprise est perdante dans les deux cas. La demande pour une période et une activité données est proportionnelle au nombre de clients attendus à cette période et pour cette activité. Il est à noter qu'il est difficile de satisfaire la demande exactement et que celle-ci peut alors être violée moyennant une pénalité.

— **Les contraintes de travail :** Différentes contraintes liées à l'horaire des employés sont imposées par les lois en vigueur et les conventions collectives signées avec les syndicats. Parmi ces contraintes, nous retrouvons les durées minimale et maximale de quart, le temps de repos minimal entre deux quarts, le temps de travail maximal dans la semaine et le nombre de jours de repos minimal dans la semaine.
— **La qualification des employés :** Chaque employé est qualifié ou non pour chacune des activités. Une activité ne peut pas être assignée à un employé qui n'est pas qualifié pour l'effectuer.
— **Les préférences des employés :** Les entreprises cherchent aussi à satisfaire les préférences de leurs employés (par exemple, les heures de travail désirées). La satisfaction de ces préférences est une sorte de motivation pour les employés et permet de créer une meilleure ambiance dans l'entreprise ainsi qu'une meilleure productivité.

Certes, avec un grand nombre d'employés, il est impossible pour un gestionnaire de générer un horaire optimal manuellement en respectant l'ensemble de ces contraintes. Cela induit le besoin de développer des outils d'aide à la décision permettant aux planificateurs d'horaires de concevoir plus rapidement et plus efficacement les bons horaires.

Nous assistons de plus en plus à l'apparition d'études abordant plusieurs aspects du problème, tels que l'attribution des jours de repos, la construction des horaires de travail et l'affectation des activités de travail aux employés. En raison de leur complexité théorique, ces problèmes requièrent des méthodes d'optimisation dédiées. Dans ce contexte, plusieurs entreprises proposent actuellement des logiciels de gestion d'horaires de personnel travaillant sur des quarts. Ces logiciels font également la mise à jour d'horaires déjà générés lors d'une perturbation. En effet, la demande observée en cours d'opération peut différer de la demande prévue à cause de plusieurs facteurs non maîtrisés. Si la prévision de la demande est mauvaise, la demande réelle risque de ne pas être satisfaite par l'horaire généré et les coûts pour l'entreprise seront plus élevés que ce qu'ils auraient pu être. Prenant l'exemple de la météo pour un magasin de vente au détail, une mauvaise météo non prévue en plein été peut diminuer la demande. De même, si un ou plusieurs employés s'absentent sans préavis, il faut les remplacer ce qui peut être considéré comme une augmentation de la demande à satisfaire. Il faut alors modifier l'horaire prévu à moindre coût tout en respectant l'ensemble des contraintes citées plus haut.

Il est essentiel que nous précisions l'environnement dans lequel ce projet de recherche se place. En effet, il y a de nombreuses variantes du problème d'horaires de personnel.
Nous distinguons deux types d'environnements : l'environnement continu où le travail se fait tout au long de la journée (hôpitaux, usines) et l'environnement non continu où il y a des heures fixes de fermeture (vente au détail, conducteurs de métro, ...). Dans le premier environnement, il y aura nécessairement des quarts de travail chevauchant deux jours successifs et il est difficile de séparer le problème par jour. Dans le deuxième cas, la variation de la demande (une demande positive le jour et presque nulle la nuit) pourrait empêcher ces chevauchements. Par contre, l'impact des temps de repos inter-journée et des heures travaillées chaque jour versus les cibles hebdomadaires demeure important. Dans notre projet, nous nous plaçons dans le cas non continu et la période de mise à jour s'étendra sur une semaine car c'est souvent la période maximale sur laquelle il y des restrictions mais aussi les horaires sont souvent planifiés une semaine à la fois dans la vente au détail.

Dans une entreprise, il y a souvent plusieurs activités à effectuer. Par exemple, dans un magasin, il faut entre autres des caissiers, des commis dans chaque département, des gardiens de sécurité, des proposés à l'entretien ménager, ... Dans un même quart, un employé pourrait donc effectuer plusieurs activités. Dans notre cas, nous considérerons que les employés ne font qu'une activité par quart afin de simplifier le problème et cela correspond aussi à la pratique de nombreuses entreprises. Un employé peut toutefois changer d'activité d'un jour à l'autre.

Dans les horaires de personnel, il y a également les pauses. Généralement, le placement des pauses à l'intérieur d'un quart est flexible mais défini par différentes règles. Ces flexibilités amènent de meilleures solutions mais aussi une plus grande difficulté de gestion. En effet, de nombreux travaux sur les horaires de personnel focalisent sur la manière de placer les pauses lors de la génération de l'horaire. Pour l'instant, nous ne nous préoccuperons pas du placement des pauses dans la ré-optimisation de l'horaire car, dans la vente au détail, ces pauses sont souvent affectées durant les opérations. La demande est alors légèrement augmentée durant les périodes usuelles de pause.

Dans une entreprise, on trouve souvent plusieurs départements où les tâches sont différentes surtout dans le domaine de la vente au détail. Chaque employé appartient alors à un département précis. Par contre, au moment de la génération d'horaires, il se peut que des transferts d'employés entre départements soient permis. Cela permet une plus grande flexibilité et donc de meilleures solutions. En effet, dans un contexte multi-départements, il y a des règles de transfert entre les départements. Dans notre cas de mise à jour de l'horaire, nous n'en prenons pas en compte parce que nous nous occupons d'un unique département à la fois.

Dans un problème d'horaire de personnel, nous parlons aussi de la sur-couverture (resp. sous-couverture) lorsque sur certaines périodes courtes, il y a trop (resp. trop peu) d'employés par rapport à la demande. Dans l'optimisation, nous pouvons soit interdire ce genre de situation soit le pénaliser pour l'éviter le plus possible. Dans notre cas, nous interdisons la sous-couverture et pénalisons la sur-couverture. Dans le premier cas, nous introduisons toutefois des quarts anonymes qui sont des quarts non attribués à aucun employé mais pris en compte comme des quarts travaillés pour répondre à la demande. Le planificateur trouve ensuite qui pourra travailler pendant ces quarts en déplaçant, par exemple, des employés d'un autre département qui a des employés à disposition. L'utilisation de quarts anonymes permet de produire une solution réalisable lorsque la sous-couverture est interdite et que la demande est dfficile à combler.

1.2 Motivation

Dans de nombreuses entreprises (par exemple, dans les magasins à grande surface), les employés sont payés selon un taux horaire qui augmente lorsque l'employé dépasse un certain nombre d'heures travaillées dans la même semaine. En effet, l'horaire planifié d'un employé pour une semaine ne dépasse pas un seuil bien déterminé. Par contre, en cours d'opération, il arrive qu'on prend une décision en temps réel pour modifier l'horaire initial pour différentes raisons ce qui peut entrainer du surtemps. Par exemple, un employé à qui on demande d'allonger son quart pour faire face à une demande plus forte que prévue peut alors tomber en surtemps, augmentant ainsi son salaire horaire pour le temps supplémentaire accumulé tout au long d'une semaine. Le taux horaire que paiera l'employeur pour une heure de travail au-delà du maximum pouvant être travaillé à taux régulier dans une semaine est plus grand que le taux qu'il pourrait payer pour d'autres employés qui ne sont pas en surtemps.
Suite à une ou plusieurs mises à jour des horaires, l'entreprise peut revoir l'horaire planifié pour le restant de l'horizon afin d'éviter le plus possible le surtemps s'accumulant. L'objectif de l'entreprise consiste alors à modifier le moins possible les quarts de travail prévus tout en cherchant à éviter de payer le surtemps (i.e., en minimisant les coûts totaux) et en assurant une couverture adéquate de tous les postes de travail.

Les logiciels de gestion d'horaires de personnel existants traitent le problème de la mise à jour d'horaires suite à une perturbation de la demande. Toutefois, cette mise à jour n'est pas effectuée de manière optimale car elle doit être faite bien souvent en temps réel. Ce projet est donc motivé par la volonté d'améliorer les logiciels existants en proposant des approches d'optimisation pour traiter en fin de journée les employés dont les quarts ont été modifiés au cours de la journée et qui tombent en surtemps. Dans la suite de ce mémoire, OEP indique le problème de ré-optimisation d'horaires de travail des employés en surtemps. Comme base de notre étude, nous utiliserons un logiciel déjà en service qui est commercialisé par notre partenaire Kronos.

1.3 Problématique et objectifs

Pour mieux expliquer la problèmatique de ce projet, nous présontons un exemple des contextes que nous avons étudié afin de tester notre méthode. Plaçons nous un mardi en fin de journée. L'horaire de lundi et mardi a déjà été travaillé. Une augmentation de la demande le mardi a eu lieu pour une certaine raison. À la fin de la journée, le gérant se rend compte que des employés vont travailler plus de 40 heures dans la semaine à cause d'heures supplémentaires qui leur ont été assignées pour couvrir la demande non prévue. On doit alors réoptimiser le reste de la semaine pour éviter de payer le surtemps. Pour la mise à jour, on ne peut modifier que l'ensemble de l'horaire du reste de la semaine (du mercredi au dimanche).

On a alors un horaire initial avec des employés en surtemps suite à des modifications apportées un jour donné. On veut revoir l'horaire à partir du lendemain afin d'éviter le plus possible le surtemps.

Dans cet horaire initial, on trouve les informations suivantes sur le problème :

- L'horizon d'étude du problème d'une durée typique d'une semaine. L'horizon est discrétisé en des sous-périodes de 15 minutes.
- La valeur de la demande pour chacune des activités et chaque période de la semaine étudiée.
- Les informations sur les employés tel que l'horaire planifié pour la semaine étudiée, les disponibilités, les qualifications, la durée minimale et maximale des quarts et la durée de repos minimale entre deux quarts.
- La liste des quarts planifiés pour la semaine et les contraintes de durée minimale et maximale pour les quarts anonymes (banque de quarts anonymes).

L'idée consistera à générer des quarts modifiés réalisables pour les employés en surtemps et pour certains autres et de contrôler le nombre d'employés non en surtemps impliqués dans les changements d'horaires. On définit alors certains types de transformations autorisées appliquées sur les quarts initiaux. On doit ré-optimiser l'horaire du reste de la semaine afin de minimiser les coûts totaux tout en préservant le plus possible l'horaire planifié. Pour ce faire, on introduit des contraintes qui limitent le nombre d'employés non en surtemps avec horaire modifié.

Pour résoudre ce problème, nous définirons un modèle en nombres entiers qui peut être résolu par un logiciel de programmation en nombres entiers commercial. Afin d'obtenir des temps de calcul plus rapides, nous explorerons aussi différentes stratégies heuristiques d'accélération. Nous testerons nos méthodes sur des données réelles fournies par la compagnie Kronos.

1.4 Plan de mémoire

Le présent document s'articule autour de six chapitres. Dans la revue de littérature présentée dans le prochain chapitre, nous répertorions certains travaux de recherche effectués sur les horaires de personnel et de leur mise à jour. Nous présenterons ensuite dans le chapitre 3 le modèle de réoptimisation ainsi que la génération des propositions de quarts modifiés. Le quatrième chapitre décrit deux approches de résolution et différentes stratégies d'accélération. Quant au chapitre 5, il comporte une illustration des tests et des résultats obtenus avant de conclure au chapitre 6.

CHAPITRE 2 REVUE DE LITTÉRATURE

Dans ce chapitre, nous présentons les travaux de recherche déjà effectués sur le problème traité. Nous commençons par discuter des recherches effectuées sur la génération d'horaires de personnel (section 2.1), puis nous présentons les travaux réalisés plus particulièrement sur la mise à jour de ces horaires (section 2.2).

2.1 Génération d'horaires de personnel

C'est depuis les années 50 que les problèmes de fabrication d'horaires sont traités en recherche opérationnelle. En effet, Dantzig a été le premier à les aborder en 1954 (Dantzig (1954)). C'était suite à un article d'Edie (1954) dans lequel Edie avait tenté d'optimiser le fonctionnement des postes de péage de New York et plus précisément le temps d'attente des véhicules qui s'y présentaient. Pour résoudre le problème, Edie proposa une méthode heuristique. Quant à lui, Dantzig a proposé de résoudre un programme linéaire en nombres entiers en se basant sur un problème de recouvrement qui vise à déterminer le nombre d'agents de péage requis pour divers quarts de travail afin de répondre à la demande et ainsi minimiser le temps d'attente des véhicules. Ainsi, le problème devient plus facile à formuler et le nombre des contraintes est beaucoup plus petit puisque celles-ci sont intégrées dans l'énumération des quarts de travail réalisables. Le cas qu'a confronté Dantzig est un cas facile. En effet, il ne considère pas les pauses à l'intérieur des quarts et il n'y a qu'une activité à la fois à effectuer. C'est pour cela que ça fonctionnait très bien dans son cas. Par contre, cette façon de procéder a l'inconvénient de contenir un nombre de variables énorme dès que l'on veut apporter un peu de flexibilité au problème. Par conséquent, malgré sa formulation assez simple, le problème devient difficile à résoudre. Par la suite, dans le but de permettre plus de flexibilité au modèle, de nombreuses recherches ont été menées pour améliorer la proposition de Dantzig (1954).

Les flexibilités qui ont fait l'objet de plusieurs travaux sont particulièrement le problème du placement des pauses et le cas multi-activités. Nous distinguons également quelques articles traitant la continuité du problème (problème continus et non continus). Par exemple, en 1974, Segal (1974) a travaillé dans un contexte non continu mono-activité. Il a proposé une approche en deux phases utilisant des modèles de flot pour résoudre le problème appliqué aux centres d'appels. La première phase consiste à construire un premier horaire en fixant les pauses au milieu du quart. Dans la deuxième phase, Segal autorise le déplacement des pauses

en fixant les horaires de début et de fin du quart. Ceci permettait d'obtenir une solution meilleure que celle de Dantzig mais qui n'est évidemment pas optimale.
Plus tard, Bechtold and Jacobs (1990) ont formulé un modèle implicite permettant de placer uniquement une pause de durée fixée à l'intérieur de chaque quart de travail, en considérant un environnement mono-activité. Par contre, il arrive souvent que nous devons placer plusieurs pauses à l'intérieur d'un même quart. Pour résoudre ce problème, Aykin (1996) a étendu le modèle établi par Dantzig. Il permet des pauses multiples dans des fenêtres de temps et, pour ce faire, il utilise des variables pour chaque fenêtre où une pause peut commencer.
Toujours dans un environnement non continu et mono-activité, Rekik et al. (2010) proposent des extensions des deux modèles de Dantzig (1954) et Aykin (1996) en gardant les avantages de chacun d'eux. Ils permettent de décomposer une pause en deux temps. Par exemple, nous pouvons répartir une pause de 30 minutes en deux pauses de 15 minutes pendant les heures de travail considérées.

Concernant l'environnement multi-activités qui est l'environnement du domaine de la vente en détail, Lequy et al. (2012) distinguent deux catégories pour le travail des employés ; le travail interruptible qu'ils appellent activité et le travail que nous ne pouvons pas interrompre appelé tâche. Travailler en tant qu'un caissier dans un supermarché est ainsi un exemple d'activité alors qu'effectuer un vol en tant que pilote d'une compagnie aérienne est une tâche. Dans la vente en détail, on s'intéressera majoritairement au travail interruptible.
En considérant cette catégorie de travail, les recherches ont commencé avec les travaux de Vatri (2001) et d'Omari (2002). Ils appliquaient l'affectation d'activités au domaine du contrôle aérien. Omari (2002) a proposé une méthode heuristique en se basant sur un réseau multiflots. Dans cette heuristique, il introduit des variables de ressources et des contraintes supplémentaires pour placer les activités à l'intérieur de quarts construits avec des pauses déjà fixées. Il découpe l'horizon de planification en intervalles de temps et résout un problème par intervalle en utilisant la décomposition de Dantzig-Wolfe. Vatri (2001) étend le modèle d'Omari (2002) afin d'effectuer la création des quarts et l'affectation des activités simultanément. Pour résoudre le problème, il a utilisé une heuristique basée sur l'énumération implicite et la génération de colonnes.

Bard and Purnomo (2005) résolvent un problème de construction d'horaire en prenant en compte les préférences des employés en utilisant de la génération de colonnes dans le domaine infirmier. Le programme en nombre entiers est un problème de recouvrement pour lequel les colonnes correspondent à des horaires alternatifs qu'un infirmier pourrait réaliser.

Demassey et al. (2005) ont aussi travaillé sur un contexte multi-activités. Dans leurs travaux, ils utilisent une approche de génération de colonnes pour la construction des horaires réalisables. Ils introduisent une contrainte appelée cost-regular global constraint. Cette contrainte force l'outil d'optimisation utilisé à ne produire que des horaires de coût réduit négatif. En 2007, Côté et al. (2007) intègrent la contrainte cost-regular global constraint dans des programmes mixtes. L'emploi de cette contrainte leur a permis d'obtenir des meilleurs résultats par rapport à d'autres obtenus sans l'utiliser.

Plus tard en 2008, Bouchard (2008) aborde le problème d'Omari (2002) en proposant une méthode exacte de programmation linéaire en nombres entiers. Le modèle qu'il a formulé est basé sur un problème de flot dans un réseau. Il utilise la coloration de graphe pour résoudre ce problème. Bouchard arrivait à résoudre le problème traité par Omari à savoir l'affectation d'activités dans des quarts déjà créés et à obtenir une solution optimale dans un temps de calcul rapide.

De nombreux autres travaux ont porté sur la construction et la sélection des quarts, comme par exemple par Côté et al. (2011a) et Côté et al. (2011b), où les auteurs choisissent d'utiliser une méthode basée sur une grammaire hors contexte.

Enfin, pour une revue de littérature plus détailléê sur la fabrication d'horaires de travail, on peut se référer à Ernst et al. (2004) et Bergh et al. (2013) qui utilisent pour la construction d'horaire une décomposition en six étapes : modéliser la demande pour chaque activité, déterminer les jours de travail et de congé de chaque employé, construire les quarts de travail, construire les lignes de travail, affecter les tâches aux quarts de travail et affecter les employés aux quarts.

2.2 Mise à jour des horaires

Concernant la mise à jour des horaires, plusieurs travaux sont effectués dans le domaine des horaires d'infirmiers. Par exemple, Moz and Pato (2003) traitent ce problème dans le cas d'absence de personnel. Ils proposent un modèle en nombres entiers qui repose sur un problème de flot avec contraintes supplémentaires. Ils ont utilisé une méta-heuristique basée sur les algorithmes génétiques pour effectuer les tests puisque les temps de résolution avec une méthode exacte étaient trop long. L'objectif de la mise à jour était de minimiser le nombre de changements imposés aux infirmières. La mise à jour se fait en jouant sur les affectations des infirmières et en ne touchant pas aux quarts de travail initialement fixés. Cependant, le domaine infirmier diffère du domaine du commerce au détail en plusieurs points. Tout d'abord, dans ce domaine, on travaille dans un environnement continu où le travail doit être effectué 24h/24. De plus, dans un tel domaine, on accorde plus d'importance aux solutions de bonne qualité qu'aux solutions les moins coûteuses. Par exemple, les infirmiers ne peuvent pas tomber en surtemps car le temps supplémentaire est rarement permis. En effet, il faut éviter que les infirmiers soient trop fatigués pour ne pas commettre des erreurs professionnelles. On ne peut donc pas permettre beaucoup de flexibilité au niveau de l"horaire comme dans le domaine de la vente au détail car l'objectif ici est très différent. Dans notre cas, nous orientons notre travail vers la mise à jour des quarts de travail planifiés et ne souhaitons donc pas fixer ces quarts comme le font Moz and Pato (2003).

En 2000, Sakkout and Wallace (2000) ont travaillé sur la mise à jour d'un horaire initial suite à une perturbation de la demande prévue (par exemple une machine qui tombe hors service). Pour résoudre le probème, il ont utilisé la programmation linéaire et la programmation par contrainte. En 2015, dans ce même contexte de mise à jour des quarts déjà planifiés, Froger (2015) propose un modèle de programmation en nombres entiers qui vise à minimiser les coûts de main-d'œuvre pour l'entreprise tout en satisfaisant la demande et les différentes contraintes de travail des employés. Elle traitait le problème de la mise à jour des horaires en travaillant essentiellement sur la modification de la demande. Puis, en 2016, Michon-Lacaze (2015) a travaillé elle aussi sur le domaine de la vente au détail. Son objectif était d'intégrer lors de la planification des horaires certains critères pour réagir face aux perturbations probables. Pour résoudre ce problème, elle a proposé deux programmes linéaires mixtes. Enfin, en 2017, Hassani et al. (2017) ont travaillé sur la mise à jour en temps de l'horaire des employés suite à une perturbation de demande pour différentes raisons.

CHAPITRE 3 MODÈLE MATHÉMATIQUE ET PROPOSITION DES QUARTS

Dans ce chapitre, nous présentons le modèle que nous avons utilisé pour résoudre le problème OEP ainsi que la manière dont nous proposons les quarts. Nous commençons par énoncer le modèle dans la section 3.1 puis nous expliquons en détail la manière dont nous générons les quarts modifiés (section 3.2). Enfin nous présentons l'ensemble des pénalités et des coûts de modification associées aux quarts générés (section 3.3).

3.1 Modèle mathématique

3.1.1 Idée générale du modèle

L'objectif de l'entreprise est de minimiser les coûts liés à la main-d'œuvre tout en satisfaisant la demande et les différentes contraintes de travail des employés. Pour modéliser le problème OEP, nous proposons un programme en nombres entiers. Ce modèle se base sur l'énumération de tous les quarts qui peuvent être utilisés. En effet, à partir d'une planification d'horaire initiale, nous générons un certain nombre de quarts modifiés que nous proposons pour la ré-optimisation. Nous expliquerons dans la section 3.2 la manière dont nous générons ces quarts. Une partie de l'horaire initial peut être fixée au besoin. Ces quarts que nous pouvons fixer sont à prendre en compte pour le calcul du coût, de la couverture de la demande et pour certaines contraintes de travail. Nous distinguons deux raisons pour lesquelles nous fixons des quarts :

- Horaire non modifiable : Il est possible que nous commençons la ré-optimisation au milieu de la semaine alors que le début de la semaine a déjà été travaillé. Nous devons alors fixer l'horaire du début de la semaine car nous ne pouvons plus le modifier.
- Accélération de la résolution : Afin d'accélérer la résolution du problème, nous pouvons choisir de fixer une partie de l'horaire si nous savons que cette partie a peu de chances d'être modifiée dans une solution optimale.

Toutes les durées sont exprimées en nombre de périodes et toutes les demandes sont définies par période. Le temps est discrétisé en périodes d'une durée de 15 minutes. Nous avons donc 96 périodes par jour et pour chaque période de 15 minutes, nous avons une valeur de demande constante pour chaque activité.

3.1.2 Structure des coûts et gestion des contraintes

Dans cette sous-section, nous présentons la structure de coûts utilisée pour le modèle et la manière dont nous gérons les contraintes.

Nous avons deux types de coûts à prendre en considération.

— Les coûts de l'optimisation : Ces coûts comprennent les coûts liés à la main-d'œuvre, les pénalités de sur-couverture et des quarts anonymes. Ces coûts pénalisent les situations que nous aimerions éviter dans l'horaire. Ils ne sont pas forcément les coûts réels payés par l'entreprise et sont déjà pris en compte par le logiciel lors de l'optimisation. Nous présentons cette structure des coûts dans l'annexe A.

— Les coûts de la ré-optimisation : Nous souhaitons modifier l'horaire initial le moins possible. Nous accordons donc à chaque quart modifié un coût qui pénalise la modification que subit l'horaire si le quart en question est choisi lors de la ré-optimisation. Nous détaillons la définition de ces coûts à la section 3.3.

Les employés sont souvent payés selon un taux horaire qui augmente avec le nombre d'heures travaillées. En plus, augmenter les coûts en fonction du temps travaillé dans la semaine permet d'assurer une équité parmi les employés par rapport au nombre d'heures travaillées dans la semaine. En effet, avoir deux employés travaillant chacun 8h coûte moins cher qu'avoir un seul employé travaillant 16h. C'est pour cela que les coûts de main-d'œuvre, des quarts anonymes et de sur-couverture sont modélisés par une fonction linéaire par morceaux. Prenons par exemple le cas des coûts de main-d'œuvre et considérons le salaire d'un seul employé qui travaille N périodes.

Le coût de la main-d'oeuvre par période est une fonction non-décroissante en escalier. Soit l^m le nombre de marches et u_k^m le nombre de périodes sur la marche k, $k = 1, \ldots, l^m$. L'utilisation de l'exposant m dans la notation indique que la fonction coût étudiée est celle de la main-d'œuvre. Nous considérons qu'un employé est en temps supplémentaire s'il travaille des heures sur la marche l^m.

Le coût unitaire de main-d'oeuvre sur la marche k est c_k^m, $k = 1, \ldots, l^m$. Ces coûts sont tels que $c_k^m < c_{k+1}^m$, $k = 1, \ldots, l^m - 1$. Ainsi, l'employé sera payé c_1^m pour les u_1^m premières périodes travaillées, c_2^m pour les u_2^m périodes suivantes et ainsi de suite. Soit T_k, $k = 1, \ldots, l^m$, le nombre de périodes payées sur chaque marche. Pour déterminer les valeurs de T_k, nous identifions d'abord k^* la dernière marche avec des périodes payées : k^* est le plus petit indice k tel que $N \leq \sum_{\ell=1}^{k} u_\ell^m$. Nous obtenons alors $T_k = u_k^m$ pour tout $k = 1, \ldots, k^* - 1$, $T_k = 0$ pour tout $k = k^* + 1, \ldots, l^m$ et $T_{k^*} = \sum_{\ell=1}^{k^*} u_\ell^m - N$. Comme le montre la figure 3.1, le salaire de cet employé qui travaille N périodes sera alors :

$$C(N) = \sum_{k=1}^{k^*} c_k^m T_k.$$

Nous utiliserons une modélisation similaire pour les coûts des quarts anonymes par période (avec la notation l^{qa}, c_k^{qa} et u_k^{qa}) et de sur-couverture par période et activité (avec la notation l^{sc}, c_k^{sc} et u_k^{sc}) pour bien répartir les heures de travail entre les employés et la sur-couverture et la sous-couverture (quarts anonymes) entre les périodes.

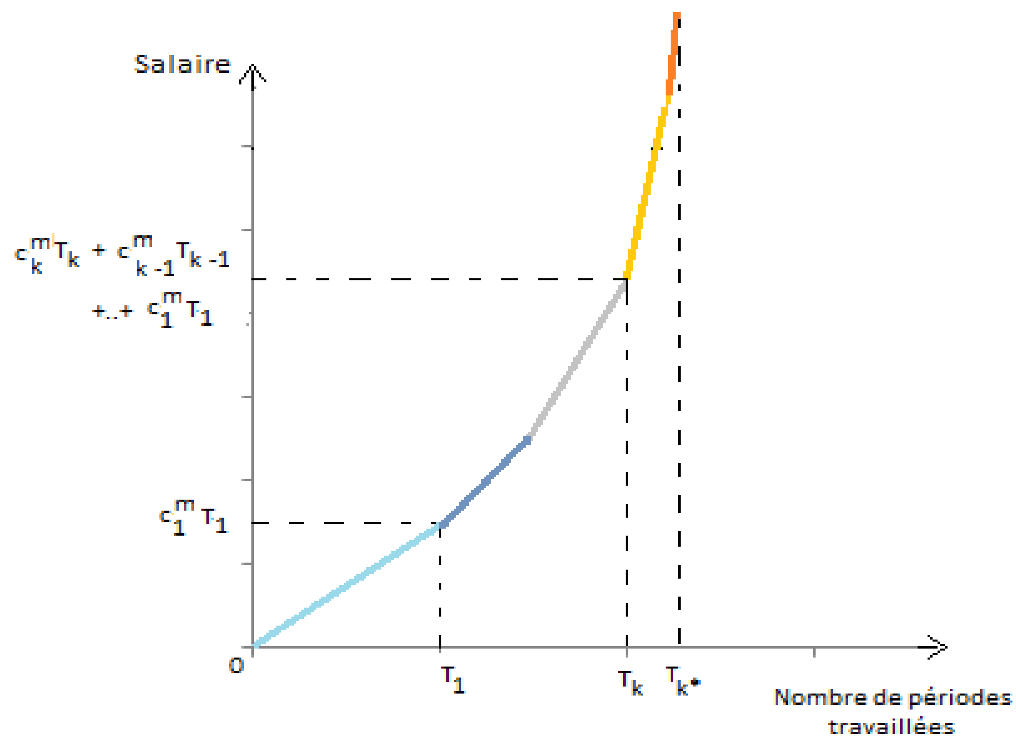

Figure 3.1 Fonction des coûts linéaire par morceau

Les contraintes que nous prenons en compte sont les suivantes :

— Satisfaction de la demande : Pour chaque période et chaque activité, on a une valeur de demande constante (un nombre d'employés) qui doit être couverte.

— Qualifications des employés : Chaque employé est qualifié pour un certain nombre d'activités. Nous pouvons affecter un quart à un employé seulement si l'activité effectuée pendant ce quart appartient à ses qualifications.

- Disponibilité des employés : Nous devons respecter la disponibilité de chaque employé. Les disponibilités sont précisées pour chacune des périodes.
- Durées de quart minimale et maximale : Pour chaque employé et chaque jour, des durées de travail minimale et maximale sont spécifiées.
- Temps de repos minimum entre les quarts : Entre deux quarts, un employé doit se reposer pendant une certaine durée minimale. Cette durée peut être différente pour chaque employé et pour chaque jour.
- Nombre de jours de repos minimal : Chaque employé doit se reposer pendant un certain nombre minimal de jours. Ce nombre varie d'un employé à un autre.
- Nombre de quarts par jour : Nous pouvons affecter au plus un quart par jour à chaque employé.
- Délai minimum d'avertissement : Pour informer un employé que nous avons modifié son horaire, un certain délai est requis. Dans notre cas, ce délai minimum d'avertissement est mis à zéro par défaut car il n'est pas précisé dans les données fournies.

Toutes ces contraintes en dehors des contraintes de la couverture de la demande, de temps de repos minimal et du nombre de quarts par jour sont prises en compte lors de la génération des quarts. Nous ne générons donc pas de quarts qui ne respectent pas l'une de ces contraintes. Le reste des contraintes est présent dans notre formulation du problème en nombres entiers.

3.1.3 Modèle mathématique

Le modèle mathématique proposé pour le problème OEP est donné dans cette section.

Ensembles

- J : L'horizon de planification d'une semaine. $J = \{1, \ldots, 7\}$
- W : Ensemble des activités.
- E : Ensemble des employés.
- ES : Ensemble des employés en surtemps.
- ENS : Ensemble des employés non en surtemps et dont on peut modifier l'horaire.
- E^m : Ensemble des employés dont l'horaire est modifiable. $E^m = (ES \cup ENS)$
- Q_e^f : Ensemble des quarts fixés pour l'employé $e \in E$.
- Q^f : Ensemble des quarts fixés pour tous les employés. $Q^f = \cup_{e \in E} Q_e^f$.
- Q^{fa} : Ensemble des quarts anonymes fixés.
- Q_e^p : Ensemble des quarts proposés pour l'employé e.

- Q^{pe} : Ensemble des quarts proposés pour tous les employés. $Q^{pe} = \cup_{e \in E} Q^p_e$.
- Q^{pa} : Ensemble des quarts anonymes proposés pour la ré-optimisation.
- Q^p : Ensemble des quarts proposés. $Q^p = Q^{pe} \cup Q^{pa}$.
- Q : Ensemble de tous les quarts du problème. $Q = Q^p \cup Q^f$
- P : Ensemble des périodes.

Constantes

- d_{wp} : Demande pour l'activité w et la période p.
- a_{qwp} : Vaut 1 si le quart q couvre l'activité w pendant la période p, 0 sinon.
- D_q : Période de début d'un quart q.
- n^R : Nombre de jours de repos minimal.
- r^j_e : Repos minimal en nombre de périodes pour l'employé e à partir du jour j.
- n_q : Durée en périodes du quart q.
- c_q : Coût de la modification du quart proposé $q \in Q^p$.
- c^m : Tableau des coûts de main-d'oeuvre.
- c^{sc} : Tableau des coûts de la sur-couverture.
- c^{qa} : Tableau des coûts des quarts anonymes.
- l^m : Taille du tableau c^m.
- l^{sc} : Taille du tableau c^{sc}.
- l^{qa} : Taille du tableau c^{qa}.
- u^m_k : Nombre maximal de périodes sur la marche k pour tout $k \in \{1, \ldots, l^m\}$.
- u^{qa}_k : Nombre maximal de quarts anonymes sur la marche k pour tout $k \in \{1, \ldots, l^{qa}\}$.
- u^{sc}_k : Nombre maximal de périodes en sur-couverture sur la marche k pour tout $k \in \{1, \ldots, l^{sc}\}$.
- M : Une constante suffisamment grande.

Variables

- x^j_{qe} : Vaut 1 si on associe le quart $q \in Q$ à l'employé $e \in E$ le jour $j \in J$, 0 sinon.
- y^j_q : Vaut 1 si on choisit le quart anonyme $q \in Q^{pa}$ le jour $j \in J$, 0 sinon.
- R^j_e : Vaut 1 si un repos est assigné à l'employé $e \in E$ le jour $j \in J$, 0 sinon.
- T_{ek} : Vaut le nombre de périodes travaillées par l'employé $e \in E$ pour tout $k \in \{1, \ldots, l^m\}$.
- QA_{pk} : Vaut le nombre de quarts anonymes couvrant la période $p \in P$ pour tout $k \in \{1, \ldots, l^{qa}\}$.
- SC_{wpk} : Vaut le nombre de quarts en sur-couverture pour l'activité $w \in W$ à la période

$p \in P$ pour tout $k \in \{1, \ldots, l^{sc}\}$.

Avec cette notation, nous proposons de formuler le problème OEP sous forme du programme linéaire en nombres entiers suivant :

$$\min \sum_{e\in E}\sum_{k=1}^{l^m} c_k^m T_{ek} + \sum_{p\in P}\sum_{w\in W}\sum_{k=1}^{l^{sc}} c_k^{sc} SC_{wpk} + \sum_{p\in P}\sum_{k=1}^{l^{qa}} c_k^{qa} QA_{pk} + \sum_{j\in J}\sum_{e\in E_m}\sum_{q\in Q^{pe}} c_q x_{qe}^j \tag{3.1}$$

sujet à :

$$\sum_{e\in E}\Big(\sum_{q\in Q^{pe}} a_{qwp} x_{eq} + \sum_{q\in Q^{fe}} a_{qwp}\Big) + \sum_{q\in Q^{pa}} a_{qwp} y_q + \sum_{q\in Q^{fa}} a_{qwp} - \sum_{k=1}^{l^{sc}} SC_{wpk} = d_{wp} \quad \forall w \in W \quad \forall p \in P \tag{3.2}$$

$$\sum_{q\in Q} x_{qe}^j + R_e^j = 1 \quad \forall e \in E \quad \forall j \in J, \tag{3.3}$$

$$\sum_{j\in J} R_e^j \geq n^R \quad \forall e \in E \tag{3.4}$$

$$MR_e^{j+1} + \sum_{q\in Q} D_q x_{qe}^{j+1} - \sum_{q\in Q} (D_q + n_q + r_e^j) x_{qe}^j \geq 0 \quad \forall e \in E \quad \forall j \in J \setminus \{7\}, \tag{3.5}$$

$$\sum_{j\in J}\sum_{q\in Q_e^p} n_q x_{qe}^j + \sum_{q\in Q^{fe}} n_q = \sum_{k=1}^{l^m} T_{ek} \quad \forall e \in E \tag{3.6}$$

$$\sum_{w\in W}\Big(\sum_{j\in J}\sum_{q\in Q^{pa}} a_{qwp} y_q^j + \sum_{q\in Q^{fa}} a_{qwp}\Big) = \sum_{k=1}^{l^{qa}} QA_{pk} \quad \forall p \in P \tag{3.7}$$

$$x_{qe}^j \in \{0,1\}, \quad \forall e \in E \quad \forall q \in Q \tag{3.8}$$

$$y_q^j \in \{0,1\}, \quad \forall q \in Q \quad \forall j \in J \tag{3.9}$$

$$SC_{wpk} \in \{0,1,\ldots,u_k^{sc}\}, \quad \forall w \in W \quad \forall p \in P \quad \forall k \in \{1,\ldots,l^{sc}\} \tag{3.10}$$

$$QA_{pk} \in \{0,1,\ldots,u_k^{qa}\}, \quad \forall p \in P \quad \forall k \in \{1,\ldots,l^{qa}\} \tag{3.11}$$

$$T_{ek} \in \{0,1,\ldots,u_k^m\}, \quad \forall e \in E \quad \forall k \in \{1,\ldots,l^m\} \tag{3.12}$$

Dans ce modèle, la fonction objectif vise à minimiser les coûts totaux tels que décrits dans la section 3.1.2. Le dernier terme représente les pénalités associées à la modification de l'horaire. Ensuite, les contraintes (3.2) sont les contraintes de couverture de la demande. La sous-couverture est comblée par les quarts anonymes et la sur-couverture est autorisée moyennant des pénalités. Les contraintes (3.3) assurent que pour chaque employé, nous n'assignons pas plus d'un quart par jour. Les contraintes (3.4) représentent le nombre de jours de repos minimal par semaine pour chaque employé. Les inégalités (3.5) représentent les contraintes de temps de repos minimal qui doivent être respectées entre les quarts travaillés. Les contraintes (3.6) et (3.7) permettent de modéliser les fonctions de coûts. Enfin, les contraintes (3.8), (3.9), (3.10), (3.11) et (3.12) sont les contraintes d'intégrité des variables.

3.2 Génération des quarts transformés

Dans cette section, nous explicitons la manière de générer les quarts que nous proposons. Nous présentons la liste de transformations possibles pour les quarts de chaque employé et l'ensemble des paramètres et des contraintes prises en compte.

3.2.1 Types de transformations

Afin de pouvoir tester notre méthode, il est nécessaire que le choix des transformations soit le plus grand et le plus flexible possible. Ces transformations doivent respecter les règles et les conventions collectives de l'entreprise. Nous définissons trois types de transformations simples :

1. Raccourcir un quart d'un côté (red).
2. Allonger un quart d'un côté (ext).
3. Modifier l'activité d'un quart (JMod).

Les transformations peuvent être aussi combinées ensemble :

— Combiner 1. avec 2. Nous obtenons alors un quart raccourci d'un côté et allongé de l'autre côté. Il est important que ces deux types de transformations interviennent de deux côtés différents du quart. La longueur d'allongement d'un côté n'est pas forcément égale à la longueur de raccourcissement de l'autre côté.

— Combiner 3. avec 1. ou 2. Nous obtenons alors un quart raccourci ou allongé d'un côté et dont l'activité est modifiée.

— Combiner 1., 2. et 3. Nous obtenons un quart décalé et dont l'activité est modifiée. Nous n'utiliserons cependant jamais cette possibilité dans nos propositions.

Nous pouvons autoriser plus ou moins de transformations. Cela nous permet d'avoir un nombre de quarts proposés différents et donc des solutions de qualités différentes et des temps de résolution différents.

Dans le problème OEP, nous n'utilisons que deux types de transformations qui sont la réduction (red) pour les employés en surtemps et l'extension (ext) pour les employés non en surtemps. Nous pourrons combiner ces deux types de transformations pour les employés non

en surtemps mais nous avons choisi de ne pas le faire car ça va inefficacement augmenter le nombre des quarts proposés. Par exemple, si nous réduisons le quart d'un employé $e \in ES$, nous allongeons ainsi le quart d'un autre employé $e_1 \in ENS$ qui peut combler le temps réduit. Si nous choisissons de réduire le quart de e_1 de l'autre côté afin de minimiser ses heures travaillées, nous allons chercher à allonger le quart d'un autre employé $e_2 \in ENS$ pour combler l'horaire réduit de e_1, etc.
Ainsi, nous interdisons la combinaison de ces deux transformations et nous n'utilisons jamais la transformation (JMod). En effet, pour les employés en surtemps, nous réduisons seulement leur quarts initiaux. nous n'avons donc pas besoin de changer leurs activités. Ensuite, puisque nous somme dans un contexte mono-activité, pour combler un horaire réduit d'un employé en surtemps, l'employé non en surtemps devra avoir la même activité que ce dernier.

Par défaut, nous générons des quarts transformés sur l'ensemble des jours où la modification de l'horaire est possible. Les quarts travaillés en dehors de ces journées sont alors fixés. Par exemple, si on est jeudi et on veut ré-optimiser le reste de la semaine pour éviter de payer le surtemps, on ne peut proposer des quarts qu'à partir de jeudi. Le début de semaine est déjà travaillé et on ne peut plus le modifier.

3.2.2 Propositions des quarts

Nous souhaitons produire des instances avec des quarts proposés et fixés de différentes natures afin de voir quelle possibilité donne de meilleurs résultats. Nous énumérons ici les différentes propositions que nous utilisons dans la génération des quarts.

Utilisation du générateur de quarts du logiciel
Nous pouvons décider de ne pas générer les quarts nous-mêmes mais de laisser le générateur de quarts du logiciel le faire. Nous pouvons décider d'utiliser cette option pour les quarts anonymes Q^{pa}, pour les quarts des employés Q^{pe} ou pour les deux Q^{p}.

Quarts fixés Q^f
Nous pouvons aussi décider de fixer ou non un certain nombre de quarts de l'horaire initial. Par exemple, nous devons fixer les quarts qui sont déjà travaillés.

Quarts initiaux
Nous devons inclure les quarts initiaux sujets à être modifiés dans l'ensemble des quarts proposés. En effet, le logiciel peut choisir de garder les quarts initiaux dans la réoptimisation.

Quarts modifiés

Soit s_e le surtemps en minutes (un multiple de 15) d'un employé $e \in ES$ en surtemps. Nous proposons les deux types de quarts modifiés suivants.

— Quarts réduits : Ces quarts sont obtenus à partir des quarts initiaux des employés en surtemps. Nous raccourcissons le quart d'un employé $e \in ES$ d'au plus s_e minutes au début ou à la fin. Par exemple si l'écart entre deux temps de coupe différents est égale à une période, nous proposons des quarts réduits de 15 min, 30 min,.. , s_e minutes.

— Quarts allongés : Par rapport à un quart réduit d'un employé $e \in ES$, nous proposons des quarts allongés pour les employés qui ne sont pas en surtemps afin de combler ce temps réduit d'au plus s_e minutes. Ces quarts allongés sont obtenus à partir des quarts initiaux des employés $e' \in ENS$. Nous allongeons alors le quart d'un employé $e' \in ENS$ d'au plus s_e minutes au début ou à la fin mais nous interdisons l'allongement de deux cotés.

Dans la figure 3.2, on présente un exemple de proposition de quarts. On réduit un quart q_e de l'employé $e \in ES$ de s_e minutes (un multiple de 15) à la fin. Soit f_{q_e} la date de fin du quart q_e et $f^r_{q_e}$ sa date de fin après raccourcissement. Pour qu'un employé non en surtemps puisse combler au plus les s_e minutes coupées, il doit posséder un quart avec la même activité et le même jour que celui du quart q_e. Ce quart doit en plus avoir une des spécifications suivantes :

— Sa date de début est entre $f^r_{q_e}$ et f_{q_e}. On allonge alors ce quart au début de 15 min, 30 min, .., s_e minutes (voir possibilité 1 sur la figure 3.2).

— Sa date de fin est entre $f^r_{q_e}$ et f_{q_e}. On allonge alors ce quart à la fin de 15 min, 30 min, .., s_e minutes (voir possibilités 2 et 3 sur la figure 3.2).

— Sa date de fin est avant $f^r_{q_e}$. On allonge alors ce quart à la fin de 15 min, 30 min, .., s_e minutes. Si dans la solution ce quart est allongé par exemple de s_e minutes, il y aura alors une sur-couverture entre la date de fin du quart avant allongement et $f^r_{q_e}$ puisque dans cette période la demande est nulle.

— Sa date de début est après f_{q_e}. On allonge alors ce quart au début de 15 min, 30 min, .., s_e minutes. Cette proposition peut également inclure une sur-couverture.

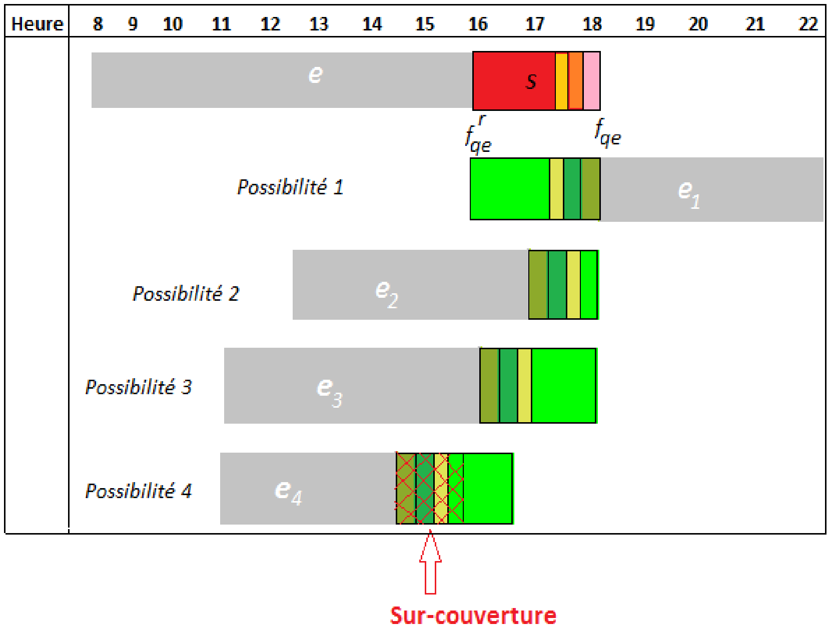

Figure 3.2 Proposition de quarts

Quarts anonymes

S'il existe des périodes pour des activités non couvertes et avec une demande positive, nous générons des quarts anonymes supplémentaires couvrant les périodes et les activités problématiques.

Nous nous inspirons du travail de Froger (2015) pour la génération des quarts anonymes. Pour générer ces quarts, Froger (2015) propose les étapes suivantes pour chaque activité :

1. Calculer pour chaque période le nombre de quarts nécessaires pour couvrir la période considérée. On obtient un tableau dont les indices correspondent aux périodes.
2. Parcourir le tableau. Si on trouve une période p_1 où la valeur est strictement positive, faire :
   — Chercher le nombre de périodes consécutives pour lesquelles cette valeur est strictement positive. On trouve p_2 la dernière période considérée.
   — Construire un ou plusieurs quarts couvrant les périodes $p \in [p1, p2]$. La durée d'un quart anonyme est comprise entre minOSLength et maxOSLength.

La construction des quarts anonymes est faite de la manière suivante :

— Si $p_2 - p_1 < minOSLength$, on construit un quart anonyme commençant à p_1 et de durée minOSLength.

— Si $minOSLength \leq p2 - p1 \leq maxOSLength$, on construit un quart anonyme commençant à p_1 et de durée $p_2 - p_1$.

— Sinon si $p_2 - p_1 > maxOSLength$, on cherche à couvrir l'intervalle $[p_1,\, p_2]$ en couvrant le moins possible de périodes en dehors de cet intervalle pour éviter d'augmenter le coût des quarts anonymes dans la solution de manière inutile. Nous utilisons pour cela les divisions euclidiennes :

— $p_2 - p_1 = A * minOSLength + r_1 \quad avec \quad 0 \leq r_1 < minOSLength.$

— $p_2 - p_1 = B * maxOSLength + r_2 \quad avec \quad 0 \leq r_2 < maxOSLength.$

On a donc $A \geq B$ et il faut $A + 1$ quarts de longueur minOSLength ou $B + 1$ quarts de longueur maxOSLength pour couvrir tout l'intervalle $[p_1,\, p_2]$.

— Si $A > B$, on a :

$$
\begin{aligned}
r_1 &= B * maxOSLength + r_2 - A * minOSLength \\
&< (B + 1) * maxOSLength - A * minOSLength \\
&= (B + 1) * (maxOSLength - minOSLength) - (A - B - 1) * minOSLength \\
&\leq (B + 1) * (maxOSLength - minOSLength) \\
&\leq A * (maxOSLength - minOSLength)
\end{aligned}
$$

r_1 peut donc être divisée en A parties de longueur l_k, $k \in \{1, \ldots, A\}$ inférieure à $(maxOSLength - minOSLength)$. Ainsi,

$$
\begin{aligned}
p_2 - p_1 &= A * minOSLength + r_1 \\
&= A * minOSLength + Al_k \\
&= A * (minOSLength + l_k)
\end{aligned}
$$

On crée donc A quarts consécutifs de longueur $minOSLength{+}l_k$. La somme des longueurs de ces quarts vaut bien $p_2 - p_1$ et les quarts couvrent donc exactement l'intervalle $[p_1,\, p_2]$.

— Si $A = B$, il en faudra au moins $A + 1 = B + 1$ quarts pour couvrir la totalité de l'intervalle quelque soit la longueur des quarts. La longueur cumulée minimum sera atteinte pour $A + 1$ quarts de longueur minOSLength et ces quarts couvrent l'intervalle.

La figure 3.3 résume la manière dont nous générons les différents types de quarts anonymes pour couvrir la demande sur un intervalle $[p_1,\, p_2]$ de longueur d. Les triangles en gris présentent les quarts anonymes utilisés dans chaque cas.

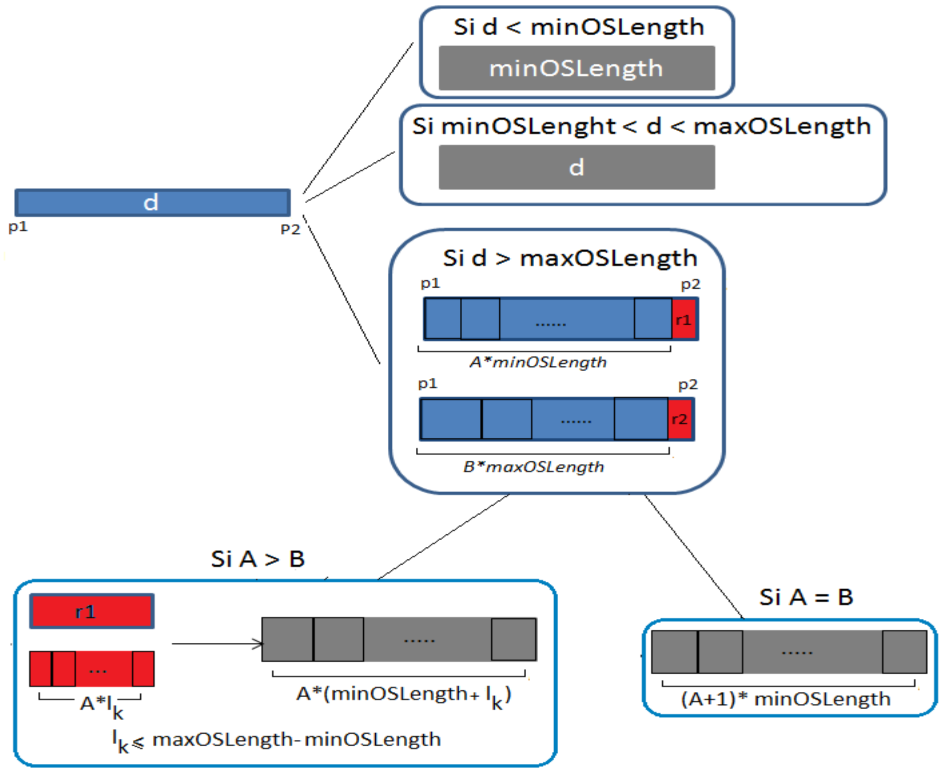

Figure 3.3 Algorithme de génération des quarts anonymes

3.2.3 Contraintes et paramètres

Il faut respecter les contraintes présentées à la section 3.1.2. Ainsi, pour qu'un quart généré fasse partie de la liste des quarts proposés pour l'employé, il faut qu'il remplisse toutes ces contraintes.
Pour pouvoir bien définir les transformations, nous présentons un ensemble de paramètres. Pour l'extension (resp. la réduction) d'un quart, nous devons définir :

— Les côtés qui peuvent être allongés (resp. réduits) : Nous pouvons permettre l'allongement (resp. la réduction) du quart seulement d'un côté (au début ou à la fin).
— Le temps minimum et maximum ajouté (resp. retiré) au quart : Nous devons définir un intervalle représentant les limites de durée que nous pouvons ajouter (resp. retirer) aux quarts.
— La discrétisation : La discrétisation est l'écart entre les différents temps de coupe (de l'extension). En effet, nous augmentons (resp. réduisons) les quarts d'une durée d. Ainsi, la différence entre deux durées de transformations proposées est toujours un multiple de la discrétisation. Par exemple, si nous choisissons une résolution égale à une période (15

minutes), nous allons toujours ajouter (retirer) un multiple d'une période aux quarts (0 min, 15 min, 30 min,..,120 min,..).

Les valeurs des paramètres par défaut sont données dans le tableau 5.1.

— s_e est la durée en périodes de surtemps d'un employé $e \in ES$.
— L'attribut **côté** est associé au côté de l'extension (resp. la réduction) et peut être égal à deux, début ou fin.
— $[minExt, maxExt]$ (resp. [minRed, maxRed]) est la fenêtre des limites de durée que nous pouvons ajouter (resp. retirer) aux quarts.
— **dis** est la discrétisation et $d = minExt + K * dis, K \in N$ (resp. $d = minRed + K * dis, K \in N$).

Tableau 3.1 Paramètres par défaut des transformations pour un employé $e \in ES$

Type	Paramètre	Valeur
ext	côté	deux
	minExt(périodes)	0
	maxExt(périodes)	s_e
	dis(périodes)	1
red	côté	deux
	minRed(périodes)	0
	maxRed(périodes)	s_e
	dis(périodes)	1

3.3 Coûts et pénalités

Nous souhaitons modifier l'horaire initial le moins possible. Nous introduisons alors des coûts supplémentaires qui pénalisent la modification des horaires des employés. Ces coûts doivent être cohérents avec la structure de coûts d'optimisation présentée dans l'Annexe A. Les pénalités seront plus ou moins élevées en fonction des priorités de l'entreprise. L'entreprise peut fixer un nombre de périodes de sur-couverture à laisser plutôt que de modifier l'horaire. Il faut aussi savoir si nous préférons garder l'équité de temps de travail entre les employés et modifier l'horaire de deux d'entre eux ou modifier l'horaire d'un seul mais perdre cette équité.

3.3.1 Pénalités de modification

Le fait de transformer le quart est un inconvénient en soi pour l'employé. Cependant, la longueur de cette transformation a aussi une influence sur lui. En effet, pour l'employé, rester une heure de plus n'a pas le même inconvénient que devoir rester un quart d'heure de plus.

Lors de la ré-optimisation, le logiciel peut ne pas choisir un quart parmi l'ensemble des quarts proposés Q_p pour un employé sur une journée.

Nous définissons alors les pénalités suivantes :

— Une pénalité fixe $\boldsymbol{fix}$ appliquée si le quart est transformé.

— Une pénalité variable $\boldsymbol{var}$ par période appliquée à chaque période modifiée. Ainsi, pour un quart réduit de deux périodes, nous avons une pénalité variable de $2var$ car il y a 2 périodes affectées par la modification au début ou à la fin du quart.

Nous devons choisir des valeurs par défaut pour nos propres tests de telle façon que les modifications que nous faisons soient préférées à la sur-couverture et aux quarts anonymes. Nous devons donc fixer des pénalités plus basses que les coûts de sur-couverture et de quarts anonymes. Pour les employés en surtemps, la seule modification autorisée est la réduction d'un quart. Pour l'employé, le fait de retrancher des périodes d'un quart qu'il devait normalement travailler ne lui cause pas d'inconvénients. Nous choisissons alors de ne pas appliquer une pénalité variable pour ce type de transformation et de se contenter de la pénalité fixe lorsque nous décidons de réduire un quart initial pour un employé en surtemps. Pour les employés non en sutemps, nous allons appliquer les deux pénalités fixe et variable pour chaque quart allongé.

Par défaut, nous choisissons :

— Pour les employés en surtemps :
  $fix = 4$
  $var = 0$/période

— Pour les employés non en surtemps :
  $fix = 4$
  $var = 7.5$/période

Nous pourrons aussi penser à assurer une équité entre les employés en ne modifiant pas l'horaire de l'un plusieurs fois alors qu'un autre ne subirait aucune modification. Avec la structure des coûts que nous proposons, nous ne pouvons assurer ce genre d'équité que d'une semaine à l'autre en associant des pénalités plus fortes aux quarts d'un employé qui a subi des modifications à son horaire la semaine précédente.

3.3.2 Tableau récapitulatif

L'ensemble des configurations possibles pour les tests et leurs caractéristiques sont résumées dans le tableau 5.2.

Nous précisons dans le tableau :

— ***exact*** si nous choisissons de ne pas générer les quarts nous-mêmes mais de laisser le générateur de quarts du logiciel le faire.

— ***fixed*** si nous choisissons de fixer l'horaire initial sans proposer aucune tranformation des quarts.

— ***trans*** est la transformation autorisée. Ça peut être une réduction *red* ou une extension *ext*.

— ***morePen*** indique la configuration de choisir des pénalités plus élevées.

— ***zeroPen*** indique la configuration de choisir des pénalités nulles.

— « Toutes » (resp. « Aucune ») indique que toute (resp. aucune) transformation est autorisée.

— La valeur des pénalités : par défaut (-), nulles ou multipliées par rapport aux valeurs par défaut.

— Si nous générons des quarts avec le générateur de quarts du logiciel pour les employés ou pour la banque de quarts anonymes.

Tableau 3.2 Tableau récapitulatif

configuration		Transformation autorisée					Pénalité			Gén.quarts	
		red	ext	JMod	Toutes	Aucune	Nulles	Multipliées	-	Emp	QA
exact					x		x			x	x
fixed						x					x
Trans	genOS	x	x						x		x
	zeroPen	x	x				x				
	morePen	x	x					x			

CHAPITRE 4 APPROCHES ET MÉTHODES DE RÉSOLUTION

Pour résoudre le problème OEP, nous avons développé différents algorithmes. Cette variété nous permettra de comparer plusieurs méthodes et approches et les résultats obtenus pour chacune d'elles. Nous avons développé une méthode exacte et deux méthodes heuristiques MH1 et MH2. Pour chacune des méthodes, on procède de deux manières : une approche séquentielle ASq et une approche simultanée ASm. Dans cette section, nous expliquerons en détails ces approches de résolution ainsi que les algorithmes développés.

4.1 Approches de résolution

4.1.1 Approche séquentielle

Dans l'approche ASq, les employés en surtemps sont traités l'un après l'autre comme c'est présentement le cas en pratique. Pour chaque employé en surtemps, on génère un scénario. Dans ce scénario, on propose des quarts réduits proposés seulement pour l'employé en surtemps en question et des quarts allongés pour les employés non en surtemps qui peuvent le remplacer et on fixe l'horaire du reste des employés. Par exemple, soit sc1 le scénario qui traite le premier employé en surtemps. Dans sc1, l'horaire de tous les autres employés en surtemps est fixé. On résout sc1 et on obtient une solution S1. Dans le scénario sc2 qui traite le deuxième employé en surtemps, on fixe l'horaire du premier employé en surtemps obtenu dans la solution S1 et on procède de la même manière pour la proposition des quarts. Un résumé de l'approche séquentielle est donné dans l'algorithme 1.

4.1.2 Approche simultanée

Dans l'approche ASm, les employés en surtemps sont traités tous à la fois. Pour chaque ensemble des employés en surtemps à traiter, on génère un scénario. Dans ce scénario, on propose des quarts réduits proposés seulement pour l'ensemble des employés en surtemps en question et des quarts allongés pour les employés non en surtemps qui peuvent les remplacer et on fixe l'horaire du reste des employés. Par exemple, soit sc2 le scénario qui traite les deux premiers employés en surtemps. Dans sc2, l'horaire de tous les autres employés en surtemps est fixé. La proposition des quarts réduits se fait seulement pour les deux premiers employés en surtemps. Un résumé de l'approche simultanée est donné dans l'algorithme 2.

Algorithme 1 Approche ASq

Soient: P_i le problème où on traite le ième employé e en surtemps, s le nombre de minutes en surtemps (un multiple de 15), ES l'ensemble des employés en surtemps dans l'horaire initial, ENS l'ensemble des employés non en surtemps dans l'horaire initial, $Begin$ la date à partir de laquelle on peut modifier l'horaire, nb la limite sur le nombre d'employés non en surtemps dont l'horaire peut être modifié, sc_i le scénario généré pour le ième employé en surtemps.

Pour tout $e \in ES$ **Faire**

Pour l'employé e, proposer des quarts réduits de 15, 30, ..., s minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Identifier dans l'ensemble ENS les employés qui peuvent remplacer l'employé e. Soit ENS_e l'ensemble de ces employés.

Si Le nombre d'employés non en surtemps modifiés par rapport à la planification initiale est inférieur strictement à nb **Alors**

Pour tout $e' \in ENS_e$ **Faire**

Pour e', proposer des quarts allongés de 15, 30, ..., s minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Finpour

Finsi

Fixer l'horaire du reste des employés.

Fixer l'horaire des jours de la semaine déjà travaillés.

Générer des quarts anonymes qui couvrent les périodes pour les activités avec une demande strictement positive.

Générer un scénario sc_i avec les quarts proposés et fixés.

Résoudre sc_i.

Finpour

Algorithme 2 ASm

Soient: N le nombre total d'employés en surtemps dans l'horaire initial, P le problème où on traite les N employés en surtemps, s_e le nombre de minutes en surtemps (un multiple de 15) de l'employé en surtemps e, ES l'ensemble des N employés en surtemps à traiter dans P, ENS l'ensemble des employés non en surtemps dans l'horaire initial, $Begin$ la date à partir de laquelle on peut modifier l'horaire, sc le scénario généré pour N employés en surtemps.

Pour tout $e \in ES$ **Faire**

Pour l'employé e, proposer des quarts réduits de 15, 30, ..., s_e minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Identifier dans l'ensemble ENS les employés qui peuvent remplacer l'employé e. Soit ENS_e l'ensemble de ces employés.

Pour tout $e' \in ENS_e$ **Faire**

Pour e', proposer des quarts allongés de 15, 30, ..., s_e minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Finpour

Finpour

Fixer l'horaire du reste des employés.

Fixer l'horaire des jours de la semaine déjà travaillés.

Générer des quarts anonymes qui couvrent les périodes pour les activités avec une demande strictement positive.

Générer un scénario sc avec les quarts proposés et fixés pour N employés en surtemps.

Résoudre sc.

4.2 Méthodes de résolution

Pour résoudre les scénarios sc_i et sc des algorithmes 1 et 2, nous avons utilisé une méthode exacte et deux méthodes heuristiques.

4.2.1 Méthode exacte ME

Le logiciel se base sur une méthode de résolution exacte, soit la méthode branch-and-cut.
Nous résolvons le problème à l'aide du solveur commercial XPRESS-MP qui implémente une méthode de type branch-and-cut. L'arbre de branchement et les coupes sont alors gérés par XPRESS-MP.
Nous utilisons les paramètres par défaut à l'exception de la tolérance sur la différence relative entre la borne inférieure et la valeur de la meilleure solution trouvée qui est fixée à zéro. L'arbre de branchement est alors parcouru en entier, assurant ainsi l'exactitude du processus de résolution.

4.2.2 Méthodes heuristiques

Le but de l'utilisation d'une heuristique est d'accélérer les temps de résolution. Dans ce cas, il n'est pas certain que la solution optimale sera obtenue. Dans le problème OEP, le nombre de quarts proposés est la variable qui affecte le plus le temps de résolution. En diminuant ainsi le nombre de quarts proposés, nous espérons diminuer le temps de résolution du problème tout en gardant la même qualité de solution. Pour ce faire, nous proposons deux méthodes heuristiques en deux phases.

MH1

La méthode heuristique MH1 comporte deux phases. Dans la première phase, on essaye de diminuer le nombre des quarts proposés pour les employés non en surtemps. Pour cela, on ne propose pas tous les quarts allongés possibles pour ces employés et on se contente de proposer des quarts qui sont allongés seulement en fin de quart de 15min, 30min, ..., s_e minutes. Par contre, on propose tous les quarts réduits possibles pour les employés en surtemps et on résout. Dans la phase 2, on analyse d'abord la solution obtenue de la phase 1. On identifie les employés non en surtemps avec horaire modifié. On propose tous les quarts allongés possibles seulement pour ces employés non en surtemps et on résout de nouveau. Les deux algorithmes 3 et 4 résument les deux phases de la méthode MH1.

MH2

La méthode heuristique MH2 comporte aussi deux phases. La première phase est la même pour MH1 et MH2. La différence entre ces deux méthodes heuristiques se présente dans la deuxième phase. Dans la phase 2 de MH2, on analyse d'abord la solution obtenue de la phase 1. On identifie les jours avec horaire modifié. On propose tous les quarts allongés possibles pour les employés non en surtemps seulement pour ces jours et on résout de nouveau. Les deux algorithmes 3 et 5 résument les deux phases de la méthode MH2.

Algorithme 3 MH1, MH2 Phase 1

Soient: P le problème traité, s_e le nombre de minutes en surtemps (un multiple de 15) de l'employé en surtemps e, ES l'ensemble des employés en surtemps dans le problème P, ENS l'ensemble des employés non en surtemps dans P, $Begin$ la date à partir de laquelle on peut modifier l'horaire.

Pour tout $e \in ES$ **Faire**

Pour e, proposer tous les quarts réduits possible à partir de $Begin$ jusqu'à la fin de la semaine.

Identifier dans l'ensemble ENS les employés qui peuvent remplacer l'employé e. Soit ENS_e l'ensemble de ces employés.

Pour tout $e' \in ENS_e$ **Faire**

Pour e', proposer des quarts allongés seulement à la fin de 15 , 30, ..., s_e minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Finpour

Finpour

Fixer l'horaire du reste des employés.

Fixer l'horaire des jours de la semaine déjà travaillés.

Générer des quarts anonymes qui couvrent les périodes pour les activités avec une demande strictement positive.

Générer un scénario sc_{It1} avec les quarts proposés et fixés.

Résoudre sc_{It1}

Algorithme 4 MH1 Phase 2

Soient: S la solution obtenue de la résolution du scénario sc_{It1}, s_e le nombre de minutes en surtemps (un multiple de 15) de l'employé en surtemps e, ES l'ensemble de employés en surtemps à traiter, ENS l'ensemble des employés non en surtemps, $Begin$ la date à partir de laquelle on peut modifier l'horaire.

Dans S, identifier les employés non en surtemps modifiés. Soit ENS_m l'ensemble des ces employés.

Pour tout $e \in ES$ **Faire**

Pour e, proposer tous les quarts réduits possible à partir de $Begin$ jusqu'à la fin de la semaine.

Pour tout $e' \in ENS_m$ **Faire**

Pour e', proposer tous les quarts allongés possibles de 15, 30, ..., s_e minutes à partir de $Begin$ jusqu'à la fin de la semaine.

Finpour

Finpour

Fixer l'horaire du reste des employés.

Fixer l'horaire des jours de la semaine déjà travaillés.

Générer des quarts anonymes qui couvrent les périodes pour les activités avec une demande strictement positive.

Générer un scénario sc_{It2} avec les quarts proposés et fixés.

Résoudre sc_{It2}

Algorithme 5 MH2 Phase 2

Soient: S la solution obtenue de la résolution du scénario sc_{It1}, s_e le nombre de minutes en surtemps de l'employé en surtemps e, ES l'ensemble des employés en surtemps à traiter, ENS l'ensemble des employés non en surtemps, $Begin$ la date à partir de laquelle on peut modifier l'horaire.

Dans S, identifier les jours avec horaire modifié. Soit J l'ensemble de ces jours.

Pour tout $e \in ES$ **Faire**

Pour e, proposer tous les quarts réduits possibles à partir de $Begin$ jusqu'à la fin de la semaine.

Pour tout $e' \in Ens$ **Faire**

Pour e', proposer tous les quarts allongés possibles de 15, 30, ..., s_e minutes pour tout jour $j \in J$.

Finpour

Finpour

Fixer l'horaire du reste des employés.

Fixer l'horaire des jours de la semaine déjà travaillés.

Générer des quarts anonymes qui couvrent les périodes pour les activités avec une demande strictement positive.

Générer un scénario sc_{It2} avec les quarts proposés et fixés.

Résoudre sc_{It2}

4.3 Gestion du nombre d'employés non en surtemps avec horaire modifié

Afin de gérer le nombre d'employés non en surtemps avec horaires modifiés, nous avons ajouté les contraintes 4.1 et 4.2 au modèle présenté dans la section 3.1.

Soit m_e la variable qui vaut 1 si l'horaire de l'employé e est modifié, 0 sinon. Les contraintes 4.1 montre qu'on peut modifier au maximum 5 quarts de l'horaire d'un employé non en surtemps soit 5 jours sur un horizon d'une semaine. Les contraintes 4.2 limitent le nombre d'employés non en surtemps dont on peut modifier l'horaire à nb employés.

$$\sum_{q \in Q^p} x_{qe} \leq 5m_e \quad \forall e \in ENS \tag{4.1}$$

$$\sum_{e \in ENS} m_e \leq nb \tag{4.2}$$

$$m_e \in \{0,1\} \tag{4.3}$$

CHAPITRE 5 RÉSULTATS NUMÉRIQUES

Dans ce chapitre, nous présentons et analysons les résultats de nos tests. Les indicateurs que nous observons sont ; le coût de la solution avec et sans prise en compte des pénalités (section 5.3.1), le temps de résolution du problème par l'optimiseur (section 5.3.2) et la transformation des quarts tels que le type et le nombre de transformations subies (section 5.3.3). Après l'observation de ces indicateurs, nous clôturons par une conclusion dans la section 5.4.

5.1 Ensemble de scénarios et données

Nous testons nos méthodes sur trois jeux de données réels fournis par notre partenaire industriel Kronos. Le premier comprend 275 employés, le deuxième 47 et le troisième 17 employés. Pour chacun de ces jeux, on a la valeur de la demande initialement prévue sur une durée d'une semaine pour chaque activité.
Afin de valider notre modèle, nous proposons de générer différentes instances de tests. Nous présentons par la suite les différents contextes ainsi que des informations sur les solutions initiales.

5.1.1 Les contextes

Les différents contextes que nous proposons d'étudier sont les suivants :

— **Contexte 1** On est mardi en fin de journée. L'horaire de lundi et mardi a déjà été travaillé. Une augmentation de la demande le mardi a eu lieu pour une certaine raison. À la fin de la journée, le gérant se rend compte que des employés vont travailler plus de 40 heures dans la semaine à cause d'heures supplémentaires qui leur ont été assignées pour couvrir la demande non prévue. On doit alors réoptimiser le reste de la semaine pour éviter de payer le surtemps. Pour la mise à jour, on peut modifier l'ensemble de l'horaire du reste de la semaine (du mercredi au dimanche) et on traite les employés en surtemps séquentiellement.

— **Contexte 2** C'est comme le contexte 1 à l'exception que pour la mise à jour, on traite les employés en surtemps simultanément.

— **Contexte 3** C'est comme le contexte 1 à l'exception que le temps supplémentaire a été ajouté jeudi (fin de semaine) et que la ré-optimisation porte du vendredi au dimanche.

Nous avons donc moins de latitude pour la mise à jour.

— **Contexte 4** C'est comme le contexte 3 sauf que pour la mise à jour, on traite les employés en surtemps simultanément.

5.1.2 Les instances

Pour résoudre le problème OEP, on a développé 3 méthodes. Une méthode exacte ME et deux méthodes heuristiques $MH1$ et $MH2$. Pour chaque méthode et chaque jeu de données, on teste les 4 contextes décrits ci-dessus.
Soit n le nombre d'employés en surtemps dans un jeu de donnés. Pour chaque parton, on génère n instances de test (ou scénarios) ; soient sc_1, sc_2, ..., sc_n. Par exemple pour le contexte 1, sc_2 est le scénario généré où on traite 2 employés séquentiellement. Par contre pour le contexte 2, sc_2 est le scénario généré où on traite 2 employés simultanément.
Avec ces différents instances, nous représentons un grand panel de situations possibles pour une mise à jour plus ou moins grande de l'horaire.

5.1.3 Données

Solutions initiales

Dans les solutions de base qu'on prend en entrée pour générer nos scénarios, il existe déjà des employés en surtemps. Nous avons ajusté les courbes de demande de telle façon que toutes ces solutions initiales ne comprennent pas de quarts anonymes et ne présentent pas de sous-couverture ni de sur-couverture dans les jours avec horaire modifiable. En effet, si la solution contient des quarts anonymes, il se peut qu'en proposant des quarts allongés, des quarts anonymes vont être éliminés et la solution change de coût sans toucher aux employés en surtemps qui est l'objet principal de la ré-optimisation.
D'un autre coté, s'il y a de la sur-couverture dans le reste de l'horaire modifiable, l'horaire peut être modifié en réduisant seulement les quarts des employés en surtemps car ces derniers peuvent travailler en sur-couverture.

Pour éviter ces deux cas et pour pouvoir jouer avec les employés en surtemps et non en surtemps, on génère une solution initiale avec surtemps sans quarts anonymes ni sur-couverture dans les jours modifiables. Comme le but est d'améliorer ces solutions de base en éliminant le surtemps, les solutions qu'on aura en testant nos instances ne présenteront pas de quarts anonymes, ni de sous-couverture. Seulement les quarts des employés vont être modifiés. Dans le tableau 5.1, nous présentons les informations générales sur les solutions de base pour chaque jeu de données.

Tableau 5.1 Informations sur les solutions de base

		275 employés	47 employés	17 employés
Nombre d'activités		5	2	2
Nombre des quarts	Employés	1088	210	77
	Quarts anonymes	0	0	0

Choix de pénalités

Dans l'annexe A, on présente la structure de coûts utilisée par le logiciel d'optimisation. On a choisi les valeurs des pénalités fix et var décrites dans la section 3.2 par rapport à cette structure. Regardons les coûts pour la main d'œuvre dans l'annexe A. Les employés qui travaillent moins de 40 heures sont payés au plus 130,65/période pour chaque période au-delà de 32 heures de travail. Ceux qui travaillent plus de 40 heures sont payés 150/période pour chaque période au-delà de 40 heures. Bien évidemment, ces coûts ne reflètent pas les coûts réels.

Dans notre cas, un employé est considéré en surtemps s'il dépasse 40 heures de travail par semaine. On considère qu'un employé est non en surtemps et qu'on peut modifier son horaire s'il travaille moins de 38 heures par semaine. Supposant maintenant qu'on a un employé e en surtemps qui dépasse 40 heures par semaine et un autre employé non en surtemps qui travaille 38 heures par semaine et qui peut remplacer e. Soit fix et var les pénalités fixe et variable définies dans la section 3.3. Pour pouvoir éliminer 15 minutes de surtemps, soit une période, tout en réduisant le coût total de la solution, il faut qu'on aie ;

$$
\begin{aligned}
fix + fix + 15var + 130.65 &< 150 \\
2fix &< 19.35 - 15var
\end{aligned}
$$

Si on met var à 7.5/période, on aura :

$$fix < 5.925$$

D'où on choisit comme pénalités :

$$\begin{aligned} fix &= 4 \\ var &= 0.5 \end{aligned} \tag{5.1}$$

Environnement logiciel

Pour la génération de nos scénarios, on a utilisé la langage de programmation Java et l'environnement de développement Eclipse. Pour l'optimisation, on a utilisé le logiciel de Kronos Workforce Central Scheduler (WFC) qui utilise Xpress-MP pour la résolution.
Les tests ont été exécutés sur une machine intel(R) Core(TM) i7 et un système d'exploitation Windows10 64 bits.

5.2 Cas extrêmes

5.2.1 Ré-optimisation exacte

La ré-optimisation exacte (notée *exact*) consiste à utiliser le générateur de quarts du logiciel pour générer des quarts pour les employés et des quarts anonymes pour avoir un horaire adapté à la demande.
La solution renvoyée par le générateur est la solution optimale et donc la solution de coût le plus faible possible. L'horaire initial n'est pas pris en compte lors de la ré-optimisation, il y aura donc beaucoup de modifications pour les quarts. D'autre part, le temps de résolution sera trop long par rapport à une perspective de mise à jour car il est de l'ordre de celui de l'optimisation initiale vu que nous utilisons le logiciel pour l'ensemble de la ré-optimisation. Pour cette proposition, toutes les transformations sont permises et les pénalités de modification sont nulles.

5.2.2 Fixation de l'horaire initial avec le surtemps

On appelle $fixed$ la proposition de fixer l'horaire initial. Ainsi, nous gardons la solution initiale avec surtemps et l'horaire initial n'est pas modifié. En fixant l'ensemble des quarts des employés, nous pourrons atteindre la solution de coût le plus élevé.

5.2.3 Impact des pénalités de modification

Comme expliqué ci-dessus, c'est à l'utilisateur de définir la valeur des pénalités de modification. Nous avons défini des pénalités par défaut fix et var. Afin de pouvoir analyser l'impact de la variation de ces pénalités sur la solution, il faut varier leurs valeurs et étudier la solution optimale à chaque variation.

— Pénalités nulles : Choisir des pénalités nulles *zeropen* revient à ne pas pénaliser la modification de l'horaire tout en gardant la transformation autorisée.

— Pénalités plus élevées : Cette proposition consiste à choisir des pénalités plus élevées. Cette proposition nous permettra d'avoir une indication sur l'impact des pénalités par rapport aux temps de calcul en particulier.

Les coûts des solutions et les temps de calcul obtenus pour les tests des méthodes *exact* et $fixed$ sur nos jeux de données sont présentés dans le tableau 5.2. $fixedD$ est l'horaire fixé avec surtemps en début de semaine (contexte 1 et 2). Par contre, $fixedF$ est la solution fixée avec surtemps en fin de semaine (contexte 3 et 4).

Tableau 5.2 Résultats des méthodes *exact* et $fixed$ pour les jeux de données

	275 employés		47 employés		17 employés	
Méthode	coût	temps (s)	coût	temps (s)	coût	temps (s)
exact	6808	8655	203040	3600	16095	480
$fixedD$	13143	2.95	204990	0.43	98287	0.6
$fixedF$	11610	2.9	205950	0.39	117225	0.67

5.3 Résultats

Nous présentons dans cette section les résultats obtenus pour les jeux de données avec 275 employés et 47 employés. Nous avons aussi réalisé des tests sur le jeu de données avec 17 employés. Le fait d'avoir trop peu d'employés a empêché de trouver des modifications qui améliorent la solution avec surtemps. En effet, pour remplacer un employé en surtemps suite à une réduction d'un de ses quarts, l'optimiseur cherche parmi les quarts proposés un quart d'un employé non en surtemps qui doit travailler la même activité dans la même période pour pouvoir couvrir la demande. Avec ce jeu de donnés de seulement 17 employés, ce n'est pas possible et donc le surtemps est conservé.

Pour le jeux de données avec 275 employés, on a deux solutions initiales. La première correspond à une planification avec surtemps et on est en début de semaine (contexte 1 et 2) et comprend 5 employés en surtemps. La deuxième est la solution initiale avec surtemps et on est à la fin de semaine (contexte 3 et 4) et comprend aussi 5 autres employés en surtemps.

On teste alors 5 scénarios ; sc_1, sc_2, ..., sc_5.
Pour le jeux de données avec 47 employés, on a aussi deux solutions initiales. Chacune comporte 3 employés en surtemps. On teste alors 3 scénario ; sc_1, sc_2 et sc_3.

5.3.1 Coûts

Dans les résultats obtenus, on regarde le coût avec les pénalités de modification, le saut d'intégrité (*gap*) et le coût sans les pénalités de modification (seulement le coût de la main-d'œuvre, des quarts anonymes et de la sur-couverture).
Le *gap* en pourcentage est calculé selon cette formule : $gap = (z_{IP} - z_{LP})/z_{LP}$ où z_{IP} est la valeur optimale en nombres entiers et z_{LP} est la valeur optimale de la relaxation linéaire.
Les coûts des solutions pour les différentes instances sont présentés dans les tableaux 5.3, 5.4, 5.5 et 5.6 pour chaque jeu des données. *withpen* est le coût avec pénalité. *zeropen* est le coût sans pénalité. *var* est la variation entre les coûts des approches séquentielle et simultanée pour chaque contexte.

Tableau 5.3 Coûts des contextes 1 et 2 (275 employés)

		Contexte 1			Contexte 2			*var* (%)	
Méthode	Instance	*withpen*	*gap* (%)	*zeropen*	*withpen*	*gap*(%)	*zeropen*	*withpen*	*zeropen*
	init	13143	0	13143	13143	0	13143	-	-
ME	sc_1	12941	0	12903	12941	0	12903	0	0
	sc_2	12409	0	12303	12409	0	12303	0	0
	sc_3	12207	0	12063	12207	0	12063	0	0
	sc_4	11675	0.012	11463	11675	0	11463	0	0
	sc_5	11578	0	11343	11578	0	11343	0	0
MH1	sc_1	12941	0	12903	12941	0	12903	0	0
	sc_2	12409	0	12303	12409	0	12303	0	0
	sc_3	12207	0	12063	12207	0	12063	0	0
	sc_4	11675	0	11463	11675	0	11463	0	0
	sc_5	11578	0	11343	11578	0	11343	0	0
MH2	sc_1	12941	0	12903	12941	0	12903	0	0
	sc_2	12409	0	12303	12409	0	12303	0	0
	sc_3	12207	0	12063	12207	0	12063	0	0
	sc_4	11675	0	11463	11675	0	11463	0	0
	sc_5	11578	0	11343	11578	0	11343	0	0

Tableau 5.4 Coûts des contextes 3 et 4 (275 employés)

		Contexte 3			Contexte 4			*var* (%)	
Méthode	Instance	*withpen*	*gap* (%)	*zeropen*	*withpen*	*gap* (%)	*zeropen*	*withpen*	*zeropen*
	init	11610	0	11610	11610	0	11610	-	-
ME	sc_1	11513	0	11490	11513	0	11490	0	0
	sc_2	11416	0	11347	11416	0	11347	0	0
	sc_3	11416	0	11347	11416	0	11347	0	0
	sc_4	11319	0	11250	11315	0	11204	-0.035	-0.4
	sc_5	11222	0	11130	11218	0	11130	-0.036	-0.78
MH1	sc_1	11513	0	11490	11513	0	11490	0	0
	sc_2	11513	0	11490	11513	0	11490	0	0
	sc_3	11513	0	11490	11513	0	11490	0	0
	sc_4	11416	0	11370	11416	0	11370	0	0
	sc_5	11416	0	11370	11416	0	11370	0	0
MH2	sc_1	11513	0	11490	11513	0	11490	0	0
	sc_2	11513	0	11490	11416	0	11370	-0.84	-1.04
	sc_3	11513	0	11490	11416	0	11370	-0.84	-1.04
	sc_4	11416	0	11370	11319	0	11250	-0.84	-1.05
	sc_5	11416	0	11370	11319	0	11250	-0.84	-1.05

Tableau 5.5 Coûts des contextes 1 et 2 (47 employés)

		Contexte 1			Contexte 2			*var* (%)	
Méthode	Instance	*withpen*	*gap* (%)	*zeropen*	*withpen*	*gap* (%)	*zeropen*	*withpen*	*zeropen*
	init	204990	0	204990	204990	0	204990	-	-
ME	sc_1	204990	0	204990	204990	0	204990	0	0
	sc_2	204908	0	204870	204908	0	204870	0	0
	sc_3	204782	0	204630	204782	0	204630	0	0
MH1	sc_1	204990	0	204990	204990	0	204990	0	0
	sc_2	204908	0	204870	204908	0	204870	0	0
	sc_3	204782	0	204630	204782	0	204630	0	0
MH2	sc_1	204990	0	204990	204990	0	204990	0	0
	sc_2	204908	0	204870	204980	0	204870	0	0
	sc_3	204782	0	204630	204782	0	204630	0	0

Tableau 5.6 Coûts des contextes 3 et 4 (47 employés)

		Contexte 3			Contexte 4			*var* (%)	
Méthode	Instance	*withpen*	*gap* (%)	*zeropen*	*withpen*	*gap* (%)	*zeropen*	*withpen*	*zeropen*
	init	205950	0	205950	205950	0	205950	-	-
ME	sc_1	205950	0	205950	205950	0	205950	0	0
	sc_2	205913	0	205890	205913	0	205890	0	0
	sc_3	205719	0	205590	205719	0	205590	0	0
MH1	sc_1	205950	0	205950	205950	0	205950	0	0
	sc_2	205950	0	205950	205950	0	205950	0	0
	sc_3	205756	0	205650	205756	0	205650	0	0
MH2	sc_1	205950	0	205950	205950	0	205950	0	0
	sc_2	205950	0	205950	205950	0	205950	0	0
	sc_3	205756	0	205650	205719	0	205590	-0.08	-0.029

On observe dans les tableaux que pour chacune des méthodes ME, MH1 et MH2, le fait de ré-optimiser plutôt que de garder le surtemps (init) permet d'améliorer le coût de la solution. En effet, pour les contextes 1 et 2, on gagne 1565 unités pour l'instance de 275 employés et 208 unités pour l'instance de 47 employés. Pour les contextes 3 et 4, on gagne 388 unités et 392 unités successivement pour 275 employés et 231 unités pour 47 employés.

Nous constatons aussi que l'approche simultanée pour le contexte de fin de semaine (contexte 4) donne des coûts plus faibles que l'approche séquentielle (contexte 3) ce qui explique une variation négative. Cela est dû au fait qu'en ré-optimisant l'horaire des employés en surtemps tous à la fois, on a plus de flexibilité dans les transformations effectuées sur les quarts. L'ensemble des quarts proposés est alors plus grand et la possibilité de trouver des quarts allongés qui remplace les quarts réduits est plus grande que dans l'approche séquentielle. Par contre, on ne voit pas cette différence de coûts entre les deux approches séquentielle et simultanée dans les contextes 1 et 2.

Pour les deux méthodes heuristiques MH1 et MH2, on voit bien que la qualité des solutions s'est dégradée par rapport à celles obtenues avec ME. Par exemple, pour le jeu de données avec 275 employés, dans le parton 3, on perd 194 unités en utilisant MH1 et MH2. Dans le contexte 4, MH2 donne une meilleure solution que MH1 mais elle reste une solution de coût plus élevée que celle obtenue en utilisant ME. On perd 194 unités en utilisant MH1 mais seulement 101 unités avec MH2.

Dans les contextes 1 et 2, MH1 et MH2 donnent les mêmes solutions que la méthode exacte. Cela s'explique par le fait que pour cette instance, les quarts allongés optimaux sont seulement des quarts allongés à la fin. Ainsi, les quarts allongés au début ne font qu'augmenter le nombre des quarts proposés inutilement. C'est pour cela que la proposition des quarts allongés seulement à la fin ne dégrade pas la qualité de la solution. En contrepartie, on proposant

des quarts allongé que d'un seul coté dans la phase 1 de chacune des méthodes MH1 et MH2 peut éliminer des quarts allongés de l'autre côté et qui sont optimaux et donc détruire la qualité de la solution.
Le fait de ne mettre aucune pénalité ($zeropen$) diminue le coût de la solution. Par exemple pour les contextes 1 et 2, au lieu de gagner 1565 unités en prenant en compte les pénalités, on gagne 1800 unités.
Enfin, nous remarquons que presque pour toutes les méthodes et tous les contextes le *gap* est nul. Ce qui montre que la solution optimale en nombres entiers est égale à la solution optimale de la relaxation linéaire dans la plupart des instances.

Nous souhaitons pouvoir comparer dans les figures 5.1 et 5.2 les différentes méthodes en terme de qualité de la solution calculée.

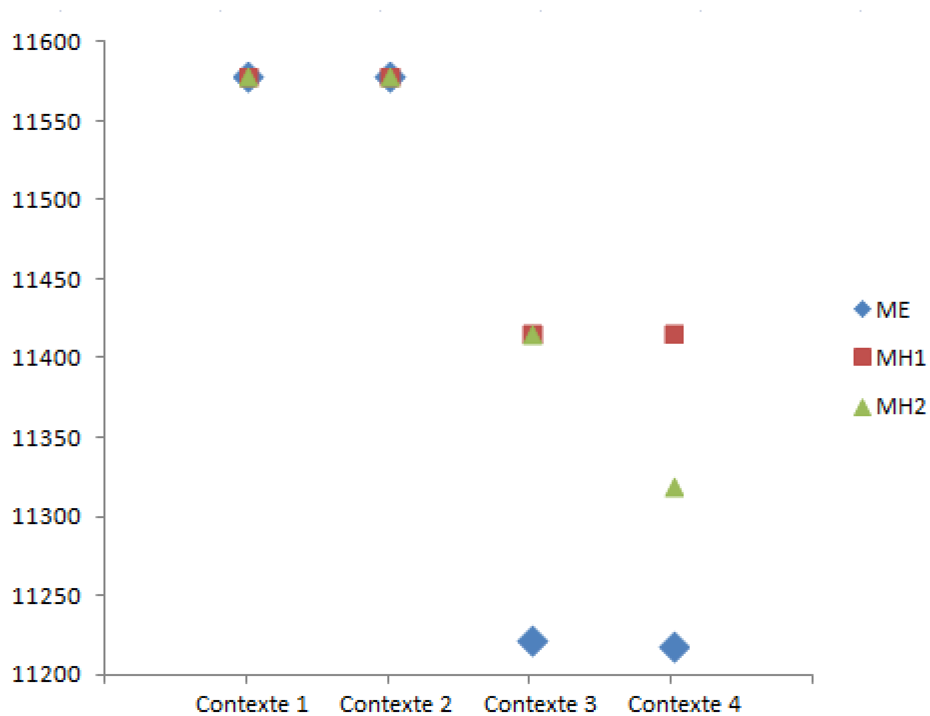

Figure 5.1 Comparaison des coûts des méthodes de résolution pour 275 employés

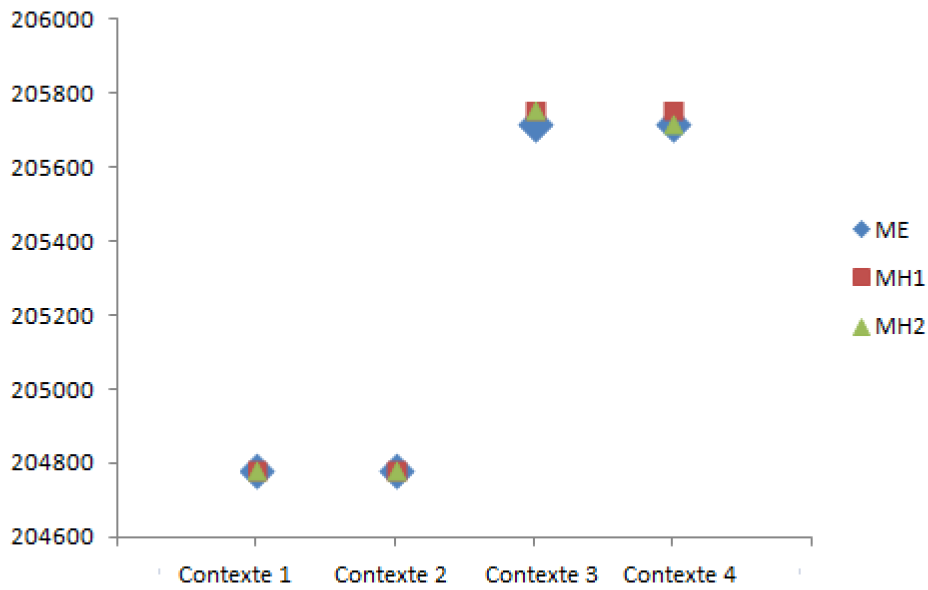

Figure 5.2 Comparaison des coûts des méthodes de résolution pour 47 employés

Il est clair que les coûts des solutions obtenues par ME sont toujours inférieurs par rapport à MH1 et MH2 et que MH2 donne des meilleurs coûts que MH1 pour le contexte 4.

5.3.2 Temps de résolution

Nous observons ensuite les temps de résolution de chaque méthode pour chaque instance. Nous rapportons aussi le nombre de quarts proposés QP dans chaque cas car il influence le temps de calcul. Dans les tableaux 5.7, 5.8, 5.9 et 5.10, le temps de calcul affiché est le temps cumulatif pour résoudre un scénario sc_i. Par exemple, le temps de résolution du scénario sc_2 dans l'approche séquentielle (contextes 1 et 3) inclut le temps de résolution de sc_1. De même pour MH1 et MH2, les temps de résolution affichés sont les sommes des temps de résolution des deux phases 1 et 2 pour chaque scénario. Par contre, le nombre de quarts proposés n'est pas cumulatif.

Tableau 5.7 Temps de résolution en secondes des contextes 1 et 2 (275 employés)

		Contexte 1		Contexte 2		*var* (%)
Méthode	Instance	temps	QP	temps	QP	temps
ME	sc_1	17.9	2583	17.73	2538	0
	sc_2	37.93	2935	61.14	4473	61.2
	sc_3	73.3	3486	84.27	5537	14.99
	sc_4	136.9	4438	112.27	5568	-17.97
	sc_5	149.3	1411	112.13	5583	-24.9
MH1	sc_1	9.58	917	9.454	917	0
	sc_2	28.2	2408	23.97	2504	-15
	sc_3	42.8	1340	28.54	2562	-33.3
	sc_4	61.5	2415	32.19	2666	-47.65
	sc_5	72	869	43.97	2962	-38.9
MH2	sc_1	12.32	1721	12.45	1721	0
	sc_2	39.8	3720	46.5	5149	16.8
	sc_3	58.98	2373	46.7	5185	-20.8
	sc_4	87.41	3710	57.85	5247	-33.8
	sc_5	99.58	2962	64.3	5533	-35.4

Tableau 5.8 Temps de résolution en secondes des contextes 3 et 4 (275 employés)

		Contexte 3		Contexte 4		*var* (%)
Méthode	Instance	temps	QP	temps	QP	temps
ME	sc_1	3.93	469	3.891	469	0
	sc_2	7.85	469	9.157	932	16.64
	sc_3	9.81	0	9.173	932	-6.47
	sc_4	14.18	469	10.57	937	-25.46
	sc_5	23	1050	30.24	2131	31.47
MH1	sc_1	3.17	297	3.59	297	0
	sc_2	6.22	292	5.81	307	-6.6
	sc_3	8.15	0	5.95	307	-27
	sc_4	10.83	313	6.17	322	-43
	sc_5	18.33	571	12.58	917	-31.36
MH2	sc_1	3.1	756	3.4	756	0
	sc_2	6.94	292	7.56	766	8.9
	sc_3	8.9	0	7.67	766	-13.8
	sc_4	12.5	756	8.1	776	-35.2
	sc_5	20.01	571	12.9	1347	-35.55

Tableau 5.9 Temps de résolution en secondes des contextes 1 et 2 (47 employés)

		Contexte 1		Contexte 2		*var* (%)
Méthode	Instance	temps	QP	temps	QP	temps
ME	sc_1	0.19	0	0.26	0	36.8
	sc_2	0.46	162	0.32	192	-30.4
	sc_3	4.39	479	1.78	488	-43.88
MH1	sc_1	0.35	0	0.4	0	14.28
	sc_2	0.83	112	0.54	112	-34.9
	sc_3	2.16	378	1.3	414	-39.8
MH2	sc_1	0.5	0	0.40	0	-18.4
	sc_2	0.98	256	0.67	256	-31.6
	sc_3	3.173	758	1.74	776	-45.16

Tableau 5.10 Temps de résolution en secondes des contextes 3 et 4 (47 employés)

		Contexte 3		Contexte 4		*var* (%)
Méthode	Instance	temps	QP	temps	QP	temps
ME	sc_1	0.15	0	0.338	0	125
	sc_2	0.44	90	0.585	178	33.6
	sc_3	1.94	407	0.95	464	-20.4
MH1	sc_1	0.321	0	0.397	0	23.68
	sc_2	0.428	65	0.664	65	55.14
	sc_3	0.993	289	1.1	297	10.77
MH2	sc_1	0.34	0	0.396	0	16.47
	sc_2	0.58	61	0.514	61	-11.37
	sc_3	1.48	646	1.439	656	-8.54

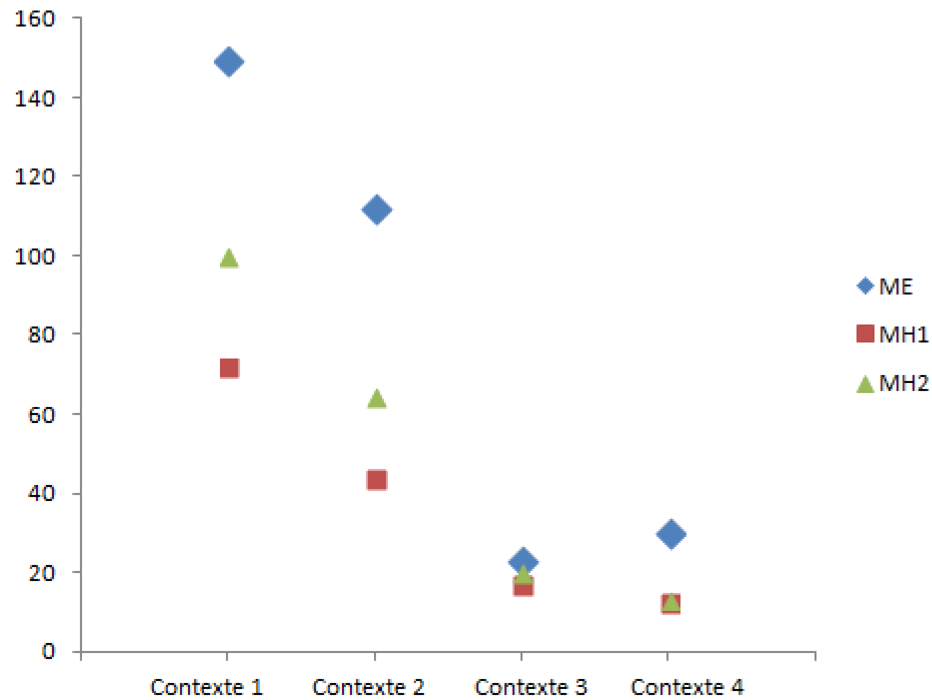

Figure 5.3 Comparaison de temps de calcul des méthodes de résolution 275 employés

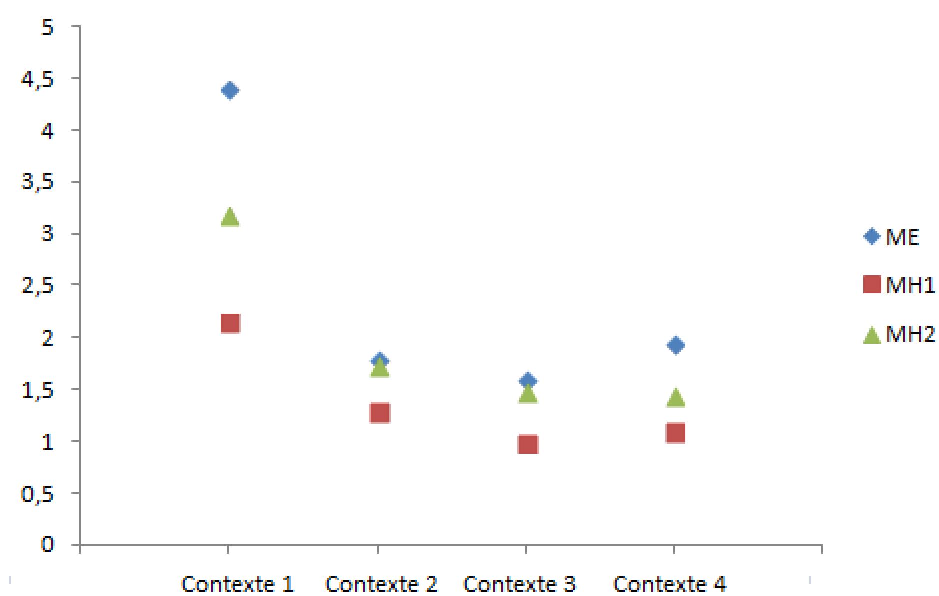

Figure 5.4 Comparaison de temps de calcul des méthodes de résolution 47 employés

De manière attendue, MH1 et MH2 permettent de réduire les temps de calcul par rapport à ME comme le présente la figure 5.3. Par exemple, dans le scénario sc_5 du contexte 2 du jeu de données avec 275 employés, on a passé de 112 secondes avec ME à 43 secondes avec MH1 (-61% de variation). Cela est dû au nombre de quarts proposés qui a diminué en éliminant certaines propositions de quarts.
Nous pouvons constater en observant les temps de résolution pour les deux méthodes MH1 et MH2 que MH1 est plus rapide, mais nous avons observé ci-haut que cette méthode produit des solutions de la plus mauvaise qualité.
Cependant, on ne peut pas conclure sur la variation du temps de résolution entre les approches séquentielle et simultanée. En effet, on observe qu'elle varie d'une instance à une autre mais dans la plupart des instances elle est négative. Ce qui fait que l'approche simultanée est plus rapide dans la majorité de temps. Par exemple, pour le jeu données avec 275 employés, le scénario sc_4 du contexte 2 et la méthode MH1, on a passé de 61 à 32 secondes (-47 % de variation).

On compare aussi le nombre de quarts proposés qui est proportionnel au nombre de variables dans le modèle avec le temps de résolution du problème dans les figures 5.5 et 5.6. Dans les tableaux, on trouve que pour quelques scénarios le nombre des quarts proposés est nul. C'est parce que pour l'employé en surtemps traité en ce scénario, il n'y a aucune possibilité de modification de son horaire pour certaines raisons. par exemple, si tous ses quarts pour le restant de l'horaire modifiable sont de longueur minimale, ils ne peuvent pas être réduits. Ainsi on ne propose aucun quart réduit et, par la suite, aucun quart allongé.

On constate qu'il y a un rapport clair entre le nombre de quarts proposés et le temps de résolution. En effet, lorsque le nombre de variables augmente dans le modèle, le temps de résolution augmente aussi. Cela justifie le développement des deux les deux méthodes heuristiques MH1 et MH2 qui permettent de sélectionner les quarts que nous proposons en amont afin de pouvoir diminuer le temps de résolution.

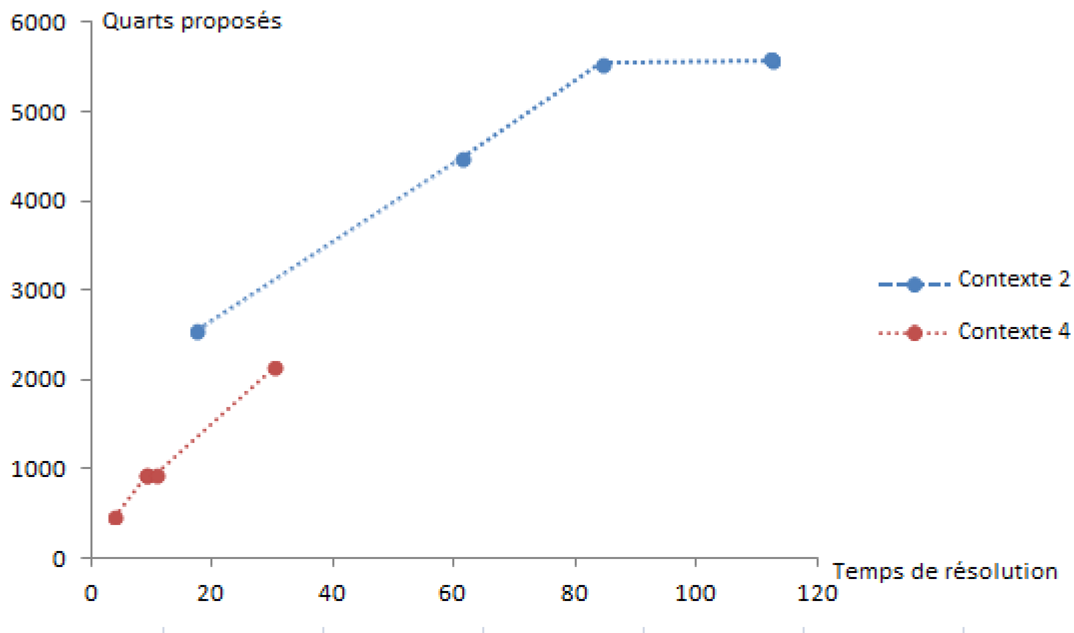

Figure 5.5 Quarts proposés versus temps de résolution (275 employés)

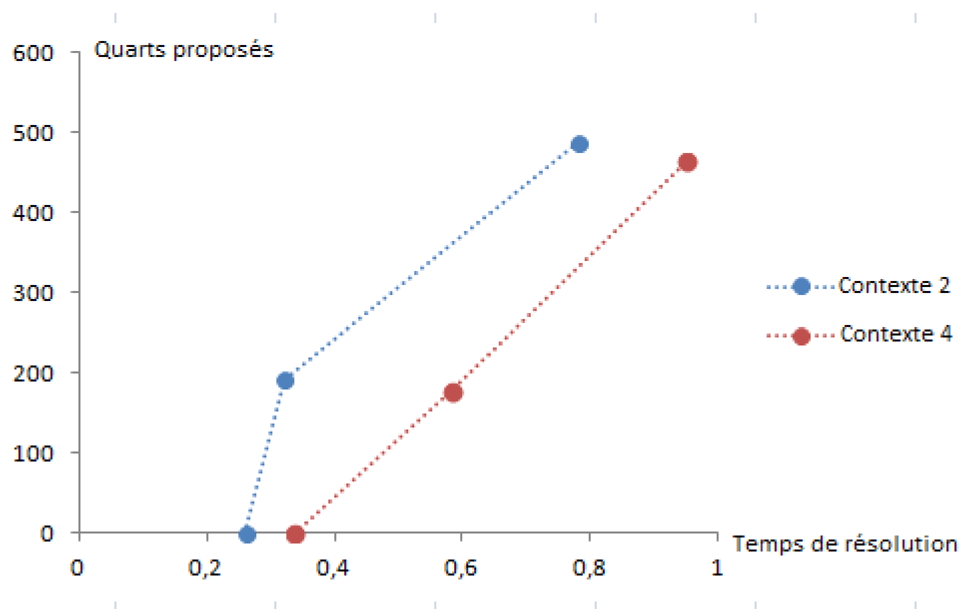

Figure 5.6 Quarts proposés versus temps de résolution (47 employés)

5.3.3 Transformation des quarts

Dans les deux tableaux 5.11 et 5.12, pour les scénarios sc_5, on spécifie pour chaque jeu de données et pour chaque ensemble d'employés en surtemps ES et non en surtemps ENS, le nombre d'employés dont l'horaire a été modifié nbE_m, le nombre des quarts modifiés nbQ_m et la durée toale réduite red et allongée ext.

Tableau 5.11 Transformation de quarts (275 employés)

		ES			ENS		
Méthode	Contexte	nbE_m	nbQ_m	red(min)	nbE_m	nbQ_m	ext(min)
ME	Contexte 1	5	5	390	5	5	390
	Contexte 2	5	5	390	5	5	390
	Contexte 3	4	4	120	4	4	120
	Contexte 4	4	4	120	3	3	120
MH1	Contexte 1	5	5	390	5	5	390
	Contexte 2	5	5	390	3	5	390
	Contexte 3	2	2	60	2	2	60
	Contexte 4	2	2	60	2	2	60
Mh2	Contexte 1	5	5	390	5	5	390
	Contexte 2	5	5	390	5	5	390
	Contexte 3	2	2	60	2	2	60
	Contexte 4	3	3	90	3	3	90

Tableau 5.12 Transformation de quarts (47 employés)

		ES_m			ENS_m		
Méthode	Contexte	nbE_m	nbQ_m	red	nbE_m	nbQ_m	ext
ME	Contexte 1	2	4	240	3	3	240
	Contexte 2	2	4	240	3	4	240
	Contexte 3	2	3	210	3	3	210
	Contexte 4	2	3	210	2	3	210
MH1	Contexte 1	2	2	240	2	2	240
	Contexte 2	2	4	240	3	4	240
	Contexte 3	2	2	120	2	2	120
	Contexte 4	1	2	180	1	2	180
MH2	Contexte 1	2	4	240	3	3	240
	Contexte 2	2	4	240	3	4	240
	Contexte 3	1	2	180	1	2	180
	Contexte 4	2	3	210	3	3	210

On peut remarquer que pour chaque instance et chaque scénario, le nombre d'employés non en surtemps modifiés ne dépasse pas 5 employés à la fin de la ré-optimisation car, dans notre cas, on limite le nombre d'employés non en surtemps qui peuvent être modifiés à 5. D'autre part, on peut constater que la durée réduite est égale à la durée allongée pour tous les contextes. En effet, le temps réduit est toujours remplacé par un quart de même longueur.

5.4 Conclusion

En conclusion de l'analyse de ces résultats, nous pouvons observer que les méthodes heuristique MH1 et MH2 permettent d'accélérer le temps de calcul mais dégradent à l'occasion et légèrement la qualité de la solution. La méthode MH2 semble plus intéressante que MH1 car ses coûts sont moins élevés et le temps de résolution est accéléré par rapport à ME.
Les approches simultanée et séquentielle produisent des solutions ayant presque les mêmes coûts sauf que pour quelques instances, l'approche simultanée a donné une bien meilleure solution.
D'autre part, c'est à l'utilisateur de déterminer à quel point l'entreprise peut se permettre de faire des modifications dans l'horaire des employés. On a ainsi deux paramètres pour la flexibilité de la ré-optimisation : la valeur des pénalités et le nombre de transformations autorisées. Plus il y a de flexibilité, plus le coût de la solution finale sera faible.

CHAPITRE 6 CONCLUSION

Ce que nous souhaitions faire dans ce projet de maîtrise était de développer des méthodes efficaces et rapides qui permettent de mettre à jour l'horaire des employés qui tombent en surtemps pour éviter de payer ce surtemps. Nous clôturons ici par une synthèse des travaux (section 6.1), une présentation des limites des solutions proposées (section 6.2) et des perspectives d'améliorations futures (section 6.3).

6.1 Synthèse des travaux

Nous avons proposé de résoudre le problème de mise à jour d'horaire d'employés qui tombent en surtemps de manière exacte et de manière heuristique en développant deux méthodes heuristiques différentes. Pour pouvoir mettre nos solutions en application et pouvoir les tester, nous avons généré des solutions initiales comprenant des employés en surtemps. Puis nous avons proposé plusieurs étapes dans notre démarche :

— Génération des quarts modifiés : Pour la mise à jour, nous proposons plusieurs quarts transformés qui engendrent des variables au modèle du logiciel d'optimisation. Nous avons donc défini différentes transformations autorisées possibles pour chacun des employés en surtemps et non en surtemps dont l'horaire peut être modifié.

— Ajout des pénalités : Nous avons adopté la même structure des coûts utilisée par le logiciel lors de l'optimisation initiale. Nous y avons ajouté des pénalités de modification des quarts pour éviter une grande modification de l'horaire.

Nous avons développé et comparé les trois méthodes suivantes :

— Méthode exacte ME : Consiste à résoudre un programme linéaire en nombre entiers en proposant tous les quarts modifiés possibles pour tous les employés avec horaire modifiable. Cette méthode renvoie les solutions de meilleure qualité avec les coûts les plus faibles.

— MH1 et MH2 : Deux méthodes heuristiques en deux phases qui visent à accélérer le temps de résolution et qui réduisent le nombre des quarts proposés en les sélectionnant en amont. La différence entre ces deux méthodes se présente dans la proposition des quarts pour les employés non en surtemps dans la deuxième phase. Pour MH1, on propose des quarts transformés seulement pour les employés non en surtemps modifiés dans la première phase. Par contre pour MH2, on ne propose des quarts transformés aux employés non en surtemps que sur les jours modifiés dans la première phase. Avec ces deux heuristiques, on est arrivé à accélérer le temps de résolution mais on a obtenu des solutions avec des coûts légèrement plus élevés.

Pour chaque méthode développé, nous avons étudié deux approches :

— Approche séquentielle : Dans cette approche, on traite les employés en surtemps l'un après l'autre. Cela permet des fois d'améliorer le temps de résolution. Dans un contexte opérationnel et avec un petit nombre d'employés, cela peut être pratique pour un gestionnaire de personnel qui désire approuver chaque modification.
— Approche simultanée : Dans l'approche simultanée, on traite les employés en surtemps tous à la fois. Cette approche permet d'avoir des coûts plus faibles que ceux renvoyés par l'approche séquentielle.

Après avoir testé nos méthodes sur des petits jeux de données (47 employés) et des plus gros jeux de données (275 employés) et en analysant les résultats obtenues, nous avons pu dégager que ME est la meilleure en terme de qualité de solution et MH1 et MH2 sont plus rapides. MH1 est aussi plus rapide que MH2. En contrepartie, MH2 dégrade moins la qualité de la solution.

6.2 Limitations de la solution proposée

Pour obtenir le coût le plus faible possible, il faut obtenir l'horaire le plus modifié. Cependant, le fait de modifier un horaire a aussi un coût indirect sur l'entreprise. En effet, les employés pourraient être insatisfaits et moins productifs si on change souvent leur horaire. Il faut alors trouver un compromis entre le coût de la solution proposée et le nombre de modifications apportées aux horaires.

D'autre part, les solutions des deux heuristiques MH1 et MH2 montrent que pour réussir à diminuer le temps de résolution, il est difficile de garder une bonne qualité de la solution au niveau de coût. Il est donc nécessaire de faire un arbitrage entre le coût et le temps de résolution.

Dans un troisième temps, nous pouvons mettre l'accent sur le fait que la mise à jour d'horaire des employés en surtemps ne peut pas se faire s'il n'y as pas d'autres employés non en surtemps qui travaillent la même activité et dans la même période et c'était le cas avec l'instance de 17 employés. Ici, il pourrait être intéressant de revoir le modèle de planification afin de favoriser le calcul de solutions planifiées qui facilitent le remplacement des employés durant les périodes susceptibles aux perturbations de la demande.

La dernière limitation est qu'on n'a pas testé nos méthodes dans un contexte réel. En effet, nous avons généré nous-même des solutions initiales avec du surtemps. Il serait donc intéressant de faire tester nos méthodes par un utilisateur réel et dans un contexte réel. En plus, travailler sur des instances de plus grande taille pourrait également influencer l'efficacité des méthodes développées.

6.3 Améliorations futures

Une des améliorations futures pourrait être de développer une troisième heuristique plus performante en faisant des recherches plus poussées sur les quarts utilisés dans la solution optimale afin de pouvoir réduire le nombre des quarts proposés. Cela permettrait peut-être d'accélérer le temps de résolution tout en gardant la qualité de la solution.
Pour répondre à la troisième limitation, et afin de se rapprocher de la réalité, une des améliorations pourrait également être de faire des recherches dans un contexte multi-activités. Ainsi les employés non en surtemps pourraient changer d'activité pour remplacer un employé en surtemps. On aura donc plus de flexibilité dans la modification de l'horaire et une plus grande possibilité d'éviter le surtemps.
Prendre en compte le placement des pauses et les préférences des employés pourraient aussi être deux pistes de recherche intéressantes. Enfin, il pourrait être intéressant de faire plus de tests avec des pénalités différentes. Cela aiderait l'utilisateur à pouvoir définir les coûts et les valeurs des pénalités.

RÉFÉRENCES

T. Aykin, “Optimal shift scheduling with multiple break windows”, *Management Science*, vol. 42, no. 4, pp. 591–602, Avril 1996.

J. F. Bard et H. W. Purnomo, “Preference scheduling for nurses using column generation”, *European Journal of Operational Research*, vol. 164, no. 2, pp. 510–534, 2005.

S. E. Bechtold et L. W. Jacobs, “Implicit modeling of flexible break assignments in optimal shift scheduling”, *Management Science*, vol. 36, no. 11, p. 1339–1351, Novembre 1990.

J. V. D. Bergh, J. Belien, P. D. Bruecker, E. Demeulemeester, et L. D. Boeck, “Personnel scheduling : A literature review”, *European Journal of Operational Research*, vol. 226, no. 3, p. 367–385, 2013.

M. Bouchard, “Coloration de graphes et attribution d’activités dans des quarts de travail”, Thèse de doctorat, École Polytechnique de Montréal, 2008.

M.-C. Côté, B. Gendron, et L.-M. Rousseau, “Modeling the regular constraint with integer programming”, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, vol. 4510, p. 29–43, 2007.

M.-C. Côté, B. Gendron, C.-G. Quimper, et L.-M. R. (2011a), “Formal languages for integer programming modeling of shift scheduling problems.” *Constraints, 16 (1)*, pp. 54–76, 2011.

M.-C. Côté, B. Gendron, C.-G. Quimper, et L.-M. R. (2011b), “Grammar-based integer programming models for multiactivity shift scheduling”, *Management Science, 57 (1)*, pp. 151–163, 2011.

S. Dahmen et M. Rekik, “Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic”, *Journal of Scheduling*, vol. 18, no. 2, p. 207–223, April 2015.

G. B. Dantzig, “A comment on edie’s "traffic delays at toll booths”, *Journal of the Operations Research Society of America*, vol. 2, no. 3, pp. 339–341, Août 1954.

S. Demassey, G. Pesant, et L.-M. Rousseau, “Constraint programming based column generation for employee timetabling”, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 217–227, 2005.

L. C. Edie, “Traffic delays at toll booths”, *Journal of the Operations Research Society of America*, vol. 2, no. 2, p. 107–138, Mai 1954.

A. Ernst, H. Jiang, M. Krishnamoorthy, et D. Sier, “Staff scheduling and rostering : A review of applications, methods and models”, *European Journal of Operational Research*, vol. 153, p. 3–27, 2004.

C. Froger, “Mise à jour des horaires de personnel travaillant sur des quarts”, Mémoire de maîtrise, École Polytechnique de Montréal, 2015.

R. Hassani, G. Desaulniers, et I. E. Hallaoui, “Real-time personnel re-scheduling after a minor disruption”, *Cahiers de GERAD*, April 2017.

Q. Lequy, G. Desaulniers, et M. Solomon, “A two-stage heuristic for multi-activity and task assignment to work shifts”, *Computers & Industrial Engineering*, vol. 63, no. 4, p. 831–841, December 2012.

H. Michon-Lacaze, “Élaboration de quarts de travail robustes aux perturbations de courtes durées”, Mémoire de maîtrise, École Polytechnique de Montréal, 2015.

M. Moz et M. Pato, “An integer multicommodity flow model applied to the rerostering of nurse schedules”, *Annals of Operations Research*, vol. 119, pp. 286–301, 2003.

Z. Omari, “Attribution des activités aux employés travaillant sur des quarts”, Mémoire de maîtrise, École Polytechnique de Montréal, 2002.

M. Rekik, J.-F. Cordeau, et F. Soumis, “Implicit shift scheduling with multiple breaks and work stretch duration restrictions”, *Journal of Scheduling*, vol. 13, no. 1, p. 49–75, Février 2010.

H. E. Sakkout et M. Wallace, “Probe backtrack search for minimal perturbation in dynamic scheduling”, *Constraints 4(5)*, pp. 359–388, 2000.

M. Segal, “The operator-scheduling problem : A network-flow approach”, *Operations Research*, vol. 22, no. 4, p. 808–823, Juillet-Août 1974.

E. Vatri, “Intégration de la génération de quarts de travail et de l’affectation d’activités”, Mémoire de maîtrise, École Polytechnique de Montréal, 2001.

ANNEXE A STRUCTURE DE COÛTS

La structure des coûts pour l'optimisation est celle utilisée par le logiciel d'optimisation actuel. On a trois types de coûts :

- **Les coûts de la main d'oeuvre** Les coûts pour l'entreprise d'embaucher un employé pour une période.
- **Les coûts des quarts anonymes** Les coûts d'utiliser un quart anonyme pour couvrir une période.
- **Les coûts de la sur-couverture** Ces coûts pénalisent le fait qu'une période soit en sur-couverture.

On souhaite éviter le plus possible la sur-couverture et les quarts anonymes. Leurs coûts sont donc plus élevés.

Coût de la main d'oeuvre Le coût de chaque employé se calcule à l'aide d'une fonction linéaire par morceaux sur le nombre total de périodes travaillées dans la semaine. Le coût par période est donné par :

1. Huit premières heures d'affectation dans la semaine = 75$/période
2. Plus de 8 et jusqu'à 16 heures d'affectation dans la semaine = 86,1$/période
3. Plus de 16 et jusqu'à 24 heures d'affectation dans la semaine = 99$/période
4. Plus de 24 et jusqu'à 32 heures d'affectation dans la semaine = 113,7$/période
5. Plus de 32 et jusqu'à 40 heures d'affectation dans la semaine = 130,65$/période
6. Chaque période après 40 heures d'affectation dans la semaine = 150$/période

Augmenter les coûts en fonction du temps travaillé dans la semaine permet d'assurer une équité parmi les employés par rapport au nombre d'heures travaillées dans la semaine. En effet, avoir deux employés travaillant chacun 8h coûte moins cher qu'avoir un seul employé travaillant 16h.
Il est à noter que les coûts ci-haut ne correspondent pas aux salaires versées aux employés. Ils proviennent tout de même d'un client de Kronos qui les avait ajustés pour obtenir des solutions satisfaisantes.

Coût des quarts anonymes Pour chaque période de l'horizon, il y a un coût linéaire par morceaux pour chaque quart anonyme couvrant cette période. Pour une période donnée on a donc :

1. Premier quart utilisé = 1500$
2. Deuxième quart utilisé = 1723,05$
3. Troisième quart utilisé = 1979,25$
4. Quatrième quart utilisé = 2273,55$
5. Cinquième quart utilisé = 2611,65$
6. Sixième et septième quart utilisé = 3000$
7. Huitième et neuvième quart utilisé = 4846,2$
8. Dixième et onzième quart utilisé = 7828,5$
9. Douzième et treizième quart utilisé = 12646,05$
10. Quatorzième et quinzième quart utilisé =20428,5$
11. Chaque quart supplémentaire utilisé à partir du seizième quart utilisé =33000$

Coût de sur-couverture Il y a un coût linéaire par morceaux pour chaque période de sur-couverture d'une activité. Pour une période et une activité données on a donc :

1. Premier employé en sur-couverture = 1500$
2. Deuxième employé en sur-couverture = 1889,85$
3. Troisième employé en sur-couverture = 2381,1$
4. Quatrième, cinquième et sixième employés en sur-couverture = 3000$
5. Septième employé en sur-couverture = 3519,9$
6. Huitième et neuvième employés en sur-couverture = 3779,7$
7. Dixième employé en sur-couverture = 4102,8$
8. Onzième, douzième et treizième employés en sur-couverture = 4762,2$
9. Chaque employé supplémentaire en sur-couverture à partir du quatorzième employé en sur-couverture = 6000$.