

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE  
APPLIQUÉES

PAR  
HICHAM BENHADDAD

ALGORITHME D'UN PROBLÈME COMPLEXE DE CONFECTION  
D'HORAIRE DE TRAVAIL

NOVEMBRE 2021

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## Résumé

L'objectif principal de chaque entreprise privée est de maximiser ses profits et aussi d'offrir un service de qualité à moindre coût. L'entreprise cherche également à générer des horaires de travail de ses employés sur un horizon de planification donné qui couvrent la demande tout en respectant les réglementations et les conventions collectives de travail. Les problèmes de construction d'horaires de personnel ont donc été étudiés depuis plusieurs décennies en recherche opérationnelle. Généralement, les problèmes de confection des horaires de travail sont résolus sur de courtes périodes. Cependant, l'organisation étudiée dans ce mémoire, qui œuvre dans le secteur maritime, doit construire un horaire pour ses employés sur une durée d'un an. Un aussi long horizon de planification est inhabituel pour confectionner des horaires de travail. Ce problème de confection des horaires de travail est modélisé comme un problème de programmation linéaire. Il comporte des contraintes dures et des contraintes souples. Les contraintes dures visent notamment à satisfaire les séquences des jours de travail et de congé et le nombre minimum d'employé requis par jour. Les contraintes souples concernent le nombre de jours de travail requis pour chaque employé, par période ou le nombre maximum d'employés requis par jour. La confection d'un horaire est fastidieuse et consomme beaucoup de temps.

Ce projet de maîtrise a pour but de résoudre ce problème complexe de confection d'horaires de personnel sur un horizon de planification d'un an. Nous proposons une approche de résolution efficace, basée sur une décomposition du problème en trois sous périodes de longueur variée (hiver, été, automne). L'efficacité de notre approche de résolution a été démontrée par des tests réalisés à partir des données d'un problème réel.

Outre la bonne qualité des solutions, nous nous sommes également intéressés au temps d'exécution. Ainsi, en décomposant le problème, nous avons réussi à générer des solutions de bonne qualité en un temps de résolution raisonnable. Le problème de confection des horaires a été résolu à l'aide du logiciel CPLEX.

## Abstract

The main objective of any private company is to maximize its profits and also to offer a quality service at a lower cost. The company also seeks to plan work schedules of its employees over a given planning horizon satisfying the demand while respecting the regulations and the collective labor agreements. The problems of constructing staff schedules have been studied for several decades in operations research. Typically, scheduling problems are solved over short periods of time. However, the organization studied in this dissertation, which operates in the maritime sector, must construct a schedule for its employees over a one-year period. Such a long planning horizon is unusual for staff scheduling. This scheduling problem is modeled as a linear programming problem. It includes hard and soft constraints. The hard constraints include satisfying the sequence of days on and days off and the minimum number of employees required per day. The soft constraints concern the number of work days required for each employee per period or the maximum number of employees required per day. Scheduling is tedious and time consuming.

The goal of this master's project is to solve this complex problem of staff scheduling over a one-year planning horizon. We propose an efficient solution approach based on a decomposition of the problem into three sub-periods of various lengths (winter, summer, fall). The effectiveness of our solution approach has been demonstrated by tests performed on data from a real problem.

In addition to the good quality of the solutions, we were also interested in the execution time. Thus, by decomposing the problem, we were able to generate good quality

solutions in a reasonable computation time. The scheduling problem was solved using the CPLEX software.

## **Avant-propos**

Au terme de la rédaction de ce présent mémoire, je souhaiterais adresser mes remerciements à ceux qui de près ou de loin ont participé à sa réalisation. A cet effet, je tiens à adresser mes sincères remerciements à ma directrice de recherche Mme Viviane Gascon pour avoir accepté de diriger mon projet de recherche. Je souhaiterais également la remercier pour sa grande disponibilité et son appui. Merci infiniment pour les efforts consentis dans la lecture et des corrections de fond et de forme apportées, sans lesquelles ce document ne pourrait atteindre ce stade de clarté et de précision.

Mes remerciements vont également à l'endroit de mes parents qui m'ont toujours orienté, appuyé et motivé. Un grand merci aussi à mes frères et sœurs, mes collègues, mes amis et à tous ceux qui ont contribué de près ou de loin à la rédaction de ce mémoire.

# Table des matières

Résumé.....	i
Abstract .....	iii
Avant-propos.....	v
Liste des tableaux .....	vii
Liste des figures .....	viii
<b>CHAPITRE 1 : INTRODUCTION.....</b>	<b>1</b>
1. 1 Problématique .....	2
1. 2 Objectif de recherche .....	3
<b>CHAPITRE 2 : REVUE DE LITTERATURE .....</b>	<b>4</b>
2.1 Problèmes de confection d'horaires de travail du personnel .....	4
2.2 Classification selon le domaine d'application .....	5
2.2 Méthodes de résolution : méthodes exactes et heuristiques.....	9
2.2.1 Méthodes exactes .....	9
2.2.2 Méthodes heuristiques .....	10
<b>CHAPITRE 3 : DESCRIPTION DU PROBLÈME ET MODÉLISATION .....</b>	<b>12</b>
3.1 Description du problème.....	12
3.2 Modélisation mathématique.....	14
<b>CHAPITRE 4 : METHODES DE RESOLUTION ET EXPÉRIMENTATION.....</b>	<b>21</b>
4.1 Résolution par la méthode exacte .....	21
4.2 Approche de décomposition .....	22
4.3 Tests et résultats .....	34
<b>5- CONCLUSION.....</b>	<b>44</b>
5-1 Synthèse des travaux .....	44
5-2 Limites des solutions proposées .....	45
5-3 Améliorations futures .....	45
<b>BIBLIOGRAPHIE .....</b>	<b>46</b>
<b>Annexe A.....</b>	<b>50</b>
<b>Annexe B .....</b>	<b>52</b>
<b>Annexe C.....</b>	<b>55</b>
<b>Annexe D.....</b>	<b>58</b>



## Liste des tableaux

Tableau 3.1 : Exemple d'horaire avec six employés de catégorie différente.....	16
Tableau 4.1 : Valeur des paramètres.....	36
Tableau 4.2 : Résultats de T0.....	37
Tableau 4.3 : Résultats de T1.....	38
Tableau 4.4 : Résultats de T2.....	39
Tableau 4.5 : Résultats de T3.....	40
Tableau 4.6 : Résultats de T4.....	41
Tableau 4.7 : Synthèse des résultats .....	42

## Liste des figures

Figure 4-1 : Schéma du système de résolution par CPLEX .....	25
Figure 4.2 : Modèle ajusté pour l'obtention d'un horaire initial .....	28
Figure 4.3 : Modèle ajusté pour la résolution du problème de la période d'hiver .....	31
Figure 4.4 : Modèle ajusté pour la résolution du problème de la période d'été .....	33
Figure 4.5 : Modèle ajusté pour la résolution du problème de la période d'automne.....	35

# CHAPITRE 1 : INTRODUCTION

La confection d'horaires de travail constitue une problématique abordée dans de très nombreux domaines que ce soit dans les services portuaires, les compagnies aériennes, dans le milieu de la santé, la vente de détail, etc.

De nos jours, la recherche d'efficacité est toujours au centre de l'attention des organisations et les entreprises ont besoin d'outils pour gérer leur main-d'œuvre et augmenter leur productivité. Un des objectifs est alors de générer les horaires des employés sur un horizon de planification donné pour répondre à diverses exigences dont celles de la convention de travail. L'organisation doit s'assurer qu'il y a suffisamment d'employés en service chaque jour de l'année. Certaines règles et contraintes doivent être prises en compte, par exemple, le nombre de jours de travail requis pour chaque employé, les séquences des jours de travail et de congé, etc.

Une organisation sans but lucratif, œuvrant dans le secteur maritime, doit construire un horaire pour ses employés sur une durée d'un an. Ce long horizon de planification, inhabituel pour confectionner des horaires de travail, est spécifique à cette organisation et à son secteur d'activité. Le nombre d'employés requis varie selon la période de l'année. L'année est donc divisée en périodes (hiver, été, automne) de longueur variée durant lesquelles varient les besoins en nombre d'employés.

Plusieurs règles complexes régissent les conditions de travail des employés. Actuellement, l'organisation planifie les horaires de travail manuellement et cela exige de nombreuses semaines. Elle souhaite donc construire un horaire dans un délai beaucoup moindre que le temps requis actuellement (qui est de quelques semaines) et qui respecte toutes les règles de la convention collective en plus de son fonctionnement interne.

Ce mémoire a pour objectif de modéliser mathématiquement ce problème complexe et d'élaborer une approche de résolution efficace et rapide qui fournira un horaire de travail de bonne qualité satisfaisant les exigences du gestionnaire.

## 1. 1 Problématique

Afin de mieux comprendre la problématique traitée dans ce mémoire, il est important de décrire le contexte particulier de l'organisation étudiée pour la confection des horaires de travail de ses employés.

L'organisation doit construire un horaire sur un an pour permettre une meilleure planification pour l'utilisation des employés. Ces derniers n'ont pas à être présents sur place. En effet, c'est un peu comme s'ils étaient de garde. Ils doivent donc être disponibles les jours où ils travaillent.

Les employés sont divisés en deux catégories ( $A$ ,  $L$ ) et, selon leur catégorie, ils ne peuvent effectuer toutes les tâches. Par conséquent, l'horaire doit faire en sorte que le nombre requis de chaque catégorie d'employés soit présent chaque jour. Si le nombre d'employés planifiés à l'horaire est supérieur au nombre d'employés requis, une surcouverture est créée. Si le nombre d'employés planifiés à l'horaire est inférieur au nombre d'employés requis, une sous-couverture est créée. Il est plus important d'éviter de créer de la sous-couverture pour assurer la qualité de service requis. Cependant, il faut trouver le bon équilibre entre la charge de travail et les employés au travail.

L'année est divisée en périodes de longueur variée durant lesquelles les besoins en nombre d'employés peuvent varier aussi. Mais chaque employé doit travailler un nombre précis de jours durant chacune des périodes tout en respectant les séquences de jours de travail et de congé

L'horaire doit s'assurer que chaque employé travaille un nombre de jours consécutifs de travail et de congé qui se situe dans un intervalle donné. En effet, pour assurer une bonne qualité de vie aux employés, il faut éviter d'avoir des périodes de travail trop longues qui fatiguent et peuvent engendrer des absences et des maladies mentales ou physiques. Il importe alors que ces jours de travail soit suivis par un nombre de jours de congé adéquat. Par exemple, en hiver, chaque employé doit travailler entre 15 et 19 jours successifs, suivi par entre 11 et 15 jours de congé successifs. Puisque le nombre de jours successifs de travail et de congé varie selon la période de l'année, cela pose un

problème lorsque l'on passe d'une période à l'autre. En effet, les transitions entre les périodes rendent plus difficiles la satisfaction des exigences sur les séquences de jours de travail et de congé.

## **1. 2 Objectif de recherche**

Cette recherche a pour objectif, dans un premier temps, de modéliser mathématiquement le problème de confection des horaires des employés de l'organisation étudiée. En général, les problèmes de confection des horaires de travail doivent être résolus sur de courtes périodes de temps. La question à poser : est-il possible de générer un horaire de travail sur une longue période du temps, comme un an, en un temps de résolution raisonnable soit en quelques heures ? Une approche de résolution efficace, basée sur une décomposition du problème en trois sous périodes de longueur variée (hiver, été, automne), est élaborée pour obtenir des temps de calcul raisonnables.

Le chapitre 2 présente une revue de littérature qui aborde les problèmes de planification des horaires du personnel de façon générale et les principales méthodes de résolution de ces problèmes. Nous justifierons notre choix de la méthode de décomposition comme approche de résolution. Le problème et sa modélisation mathématique sont présentés dans le chapitre 3. Le chapitre 4 présente les approches de résolution (résolution du problème global et approche de décomposition), et les résultats des tests numériques effectués avec les deux approches. Au dernier chapitre, nous résumons les travaux réalisés tout au long du projet en énonçant un ensemble de remarques concernant l'application de ce type d'algorithmes et la résolution d'un problème de planification des horaires de travail, en précisant les améliorations à effectuer dans le futur.

## CHAPITRE 2 : REVUE DE LITTÉRATURE

Les problèmes de planification d'horaires de travail ont été largement étudiés dans la littérature. Les coûts de main-d'œuvre sont considérés comme une part importante des coûts directs d'une organisation. Il est nécessaire de planifier un nombre d'employés requis à l'heure afin d'éviter la sous-couverture et la sur-couverture. Il importe donc de planifier efficacement les horaires du personnel. Dans ce chapitre, nous présentons les travaux réalisés sur le problème de planification d'horaires du personnel dans un cadre général et nous explorons les études qui se rapprochent le plus de notre problématique. Ensuite, nous présentons les différentes méthodes de résolution : méthodes exactes et heuristiques.

### 2.1 Problèmes de confection d'horaires de travail du personnel

C'est depuis les années 50 que les problèmes de confection d'horaires de travail du personnel sont traités. Dès 1954, Edie (1954) avait tenté d'optimiser le fonctionnement des postes de péage de New York en réduisant le temps d'attente des véhicules. Edie (1954) proposa alors une heuristique pour trouver une solution au problème d'heure du personnel afin de réduire la congestion. En réponse, Dantzig (1954) précise le modèle de programmation linéaire en nombres entiers qui vise à déterminer le nombre d'agents de péage requis par quarts de travail, sans prendre en compte les pauses durant les quarts de travail, afin de répondre à la demande et minimiser le temps d'attente des véhicules aux postes de péage.

Baker (1976) a divisé les problèmes de confection d'horaires en trois classes. La première consiste à déterminer l'affectation des quarts de travail aux employés où un horaire quotidien est établi. La seconde concerne le problème de planification des journées de congé. Et enfin, la troisième classe est le jumelage des deux premières. C'est le problème de construction des cycles de travail. Ce problème consiste à déterminer simultanément les jours de travail et les jours de repos ainsi que les quarts de travail travaillés chaque jour par chaque employé.

Van den Bergh *et al.* (2013) proposent plusieurs étapes pour résoudre le problème de confection d'horaire de travail : détermination de la demande pour chaque tâche, planification des jours de travail et de congé de chaque employé, génération des quarts de travail, affectation des tâches aux quarts de travail et affectation des employés aux quarts de travail. Ces étapes sont en soi des problèmes variés abordés par plusieurs chercheurs (Kletzander et Musliu (2020), Bürgy *et al.* (2019)). Pour une revue de littérature plus détaillée sur les problèmes de génération d'horaires de travail, on peut se référer à Özder *et al.* (2020).

Dans ce mémoire, nous nous intéressons à la confection des horaires de travail d'employés dans un contexte particulier sur une période d'un an. La section suivante présente des problèmes d'horaires dans divers domaines étudiés dans la littérature ainsi que certains détails de leur modélisation de même que les méthodes de résolution utilisées pour ces problèmes.

## **2.2 Classification selon le domaine d'application**

La gestion des effectifs en personnel existe dans de nombreux domaines d'application, que ce soit dans les services portuaires, les compagnies aériennes, le milieu de la santé, la sécurité publique ou la vente au détail etc. Dans cette section, nous présentons brièvement les travaux déjà effectués sur le problème de confection des horaires de travail pour plusieurs domaines d'application.

### **Domaine de la santé**

Les applications au domaine de la santé sont nombreuses et se focalisent sur deux problématiques majeures : la confection d'horaires du personnel infirmier et des médecins. Les infirmières sont soumises à des conventions collectives, ce qui impose des règles assez strictes à respecter. Généralement, un des objectifs pour le problème de confection d'horaires du personnel infirmier est de minimiser le nombre de jours où il manque d'infirmières. Tandis que, pour les médecins, on vise plutôt à maximiser leurs préférences quant aux horaires de travail. Dans ce dernier cas, une revue de littérature détaillée a été réalisée par Erhard *et al.* (2018). Cependant, étant donnée la pénurie d'infirmières un peu partout dans le monde, plusieurs chercheurs se sont également

penchés sur la prise en compte des préférences des infirmières pour leurs horaires de travail.

Ferrand *et al.* (2011) traitent le problème de planification des horaires des médecins pour l'urgence d'un hôpital de Cincinnati. Ils construisent des horaires cycliques à l'aide d'un modèle de programmation linéaire en nombres entiers. Chaque journée est divisée en trois quarts de huit heures. Un médecin doit être présent durant chacun d'entre eux. Dans Mansini et Zanotti (2020), un problème de confection des horaires de chirurgiens est résolu par une heuristique basée sur une approche de recherche dans un voisinage élargi (large neighborhood search) afin de minimiser le temps supplémentaire travaillé.

Guo et Bard (2022) formulent le problème de confection des horaires des infirmières comme un problème de programmation linéaire en nombres entiers. Ils prennent en considération les préférences des infirmières, les séquences de jours consécutifs de travail et de congé en plus des contraintes habituelles. L'affectation des quarts de travail aux infirmières est faite dans une première étape. La solution est améliorée par une approche de génération de colonnes.

Nobile *et al.* (2021) ont étudié un problème d'horaires des infirmières de l'hôpital Pasargad en Iran. Un problème de programmation linéaire en nombres entiers mixte est proposé pour affecter les quarts de travail aux infirmières de manière à respecter le personnel hospitalier requis par quart. L'objectif de cette étude est d'optimiser le planning du personnel ayant les mêmes compétences afin de minimiser les coûts variables de l'hôpital.

Zanda *et al.* (2018) traitent un problème d'affectation des quarts de travail à des infirmières travaillant dans un hôpital universitaire en Italie. L'horizon de planification est d'un an tout comme le problème traité dans ce mémoire. Chaque jour est divisé en un certain nombre de quarts de travail qui doivent être attribués aux infirmières. Une série de contraintes doivent être satisfaites lors de la confection des horaires : la demande en personnel pour chaque quart de travail doit être satisfaite, les infirmières ne peuvent être affectées à plus d'un quart par jour, le nombre de jours de travail consécutifs est limité, etc. Le problème est modélisé comme un problème de programmation linéaire en nombres entiers linéaire. La fonction multi-objectif



minimise la pénurie d'infirmières, l'écart entre le nombre d'heures de travail requis par infirmière et le nombre d'heures réellement travaillées, et le nombre de jours de congé entre 2 jours de travail. L'approche de résolution est basée sur l'idée de diviser le problème en plusieurs sous problèmes (un par mois). Le sous-problème d'une période prend en compte la solution de la période précédente. Ce problème ne prend pas en considération les contraintes sur le nombre maximal et minimal de jours consécutifs de congé, ni le fait que la demande peut varier selon la période de l'année, ce qui a un impact sur le respect des jours consécutifs de travail et de congé.

### **Domaine de vente au détail**

Plusieurs travaux ont été réalisés sur la construction des horaires de travail dans le domaine de la vente au détail. Selon Yung *et al.* (2020), le milieu de la vente au détail est un environnement multi-activités, c'est-à-dire que les employés peuvent travailler sur plusieurs postes différents selon leurs qualifications. Le coût lié à la main d'œuvre est considéré comme une composante importante des coûts directs dans le secteur de la vente au détail. Yung *et al.* (2020) propose un modèle d'optimisation pour générer les horaires de travail d'employés d'une chaîne de magasins de détail de façon à maximiser les profits.

Froger (2015) propose, dans son mémoire, un modèle de programmation linéaire en nombres entiers qui vise à minimiser les coûts de main-d'œuvre pour une entreprise œuvrant dans la vente au détail, tout en satisfaisant la demande en personnel et les différentes contraintes de travail des employés. Le mémoire considère la mise à jour des horaires selon les modifications de la demande. Puis, en 2016, dans ce même contexte de mise à jour des quarts de travail déjà planifiés, Michon-Lacaze (2016) a intégré certains critères pour réagir face aux modifications de la demande ou autres lors de la planification des horaires. Bürgy *et al.* (2019) reprennent ce problème en incluant des aspects reliés aux incertitudes et la possibilité d'allonger les quarts de travail.

Parisio et Jones (2015) résolvent le problème de planification des horaires des employés dans les magasins de détail modélisé comme un problème stochastique de programmation linéaire en nombres entiers. Il s'agit de construire un horaire hebdomadaire pour un personnel polyvalent répondant à une demande incertaine du

personnel de vente. Les contraintes liées à la convention collective et aux qualifications des employés requises pour l'attribution des quarts de travail sont prises en considération.

### **Domaine du transport**

Les problèmes de confection des horaires des équipages ont été largement étudiés dans divers secteurs du transport tels que les compagnies aériennes (Wen *et al.* (2021), Zeighami et Soumis (2019)), maritimes (Wu *et al.* (2020), Lorenzo-Espejo *et al.* (2021), (Leggate *et al.* (2018))), ferroviaires (Heil *et al.* (2020)) et le transport urbain (Perumal *et al.* (2019), Lin et Hsu (2016), Kang *et al.* (2019)). L'horizon de planification est différent d'un secteur à l'autre. Les compagnies aériennes planifient généralement les horaires de leurs équipages sur une base mensuelle alors que les compagnies de bus utilisent souvent des horaires quotidiens ou hebdomadaires. Les compagnies maritimes établissent des horaires pour plusieurs mois à la fois.

Les compagnies aériennes recherchent continuellement des moyens de mieux utiliser leurs ressources pour augmenter leurs profits. Wen *et al.* (2021) présente un recensement de plusieurs modèles et algorithmes pour résoudre divers problèmes liés aux horaires des équipages et des pilotes dans le secteur aérien. Zeighami et Soumis (2019) présentent une extension du problème complexe de *crew pairing* qui consiste à déterminer l'équipage qui couvrira un ensemble de vols au moindre coût. Ils prennent en compte le nombre de pilotes disponibles, les périodes minimales de repos, etc. Day et Ryan [1997] proposent une méthode en deux étapes pour l'affectation des quarts de travail aux agents de bord sur une période de 14 jours. Ils génèrent toutes les séquences possibles de jours de repos et de travail pour les 14 jours. Une séquence est attribuée à chaque employé tout en s'assurant d'avoir assez d'employés pour couvrir les tâches.

Wu *et al.* (2020) étudient un problème de pilotage dans un port maritime. Il s'agit d'affecter des quarts de travail et les navires à piloter durant chaque quart de travail aux pilotes maritimes. Wu *et al.* (2020) ont développé un modèle de programmation linéaire mixte en nombres entiers (MIP) pour ce problème. L'objectif consiste à minimiser le coût encouru par les opérations de pilotage. Le problème est fortement NP-complet et un algorithme de branch-and-price (B&P) a été développé pour le résoudre.

Lorenzo-Espejo *et al.* (2021) traitent un problème de planification des jours de congé des pilotes maritimes. Le planning doit affecter, chaque jour, le nombre de pilotes nécessaires pour couvrir la demande quotidienne en personnel. Les pilotes peuvent travailler au plus un certain nombre de jours consécutifs. Deux modèles de programmation mathématique ont été présentés. Le premier modèle permet de construire des horaires avec deux longues périodes de vacances de durée limitée pour chaque pilote et minimise l'écart entre les charges de travail des pilotes. La possibilité de modifier la durée minimale des vacances permet aux gestionnaires de prendre en considération les préférences des pilotes. La formulation utilisée pour le second modèle permet de maximiser la durée des vacances des pilotes tout en minimisant les écarts entre les durées des vacances entre les pilotes.

Dans la prochaine section, nous nous intéressons aux différentes méthodes de résolution proposées dans la littérature.

## **2.2 Méthodes de résolution : méthodes exactes et heuristiques**

Les problèmes de planification d'horaires sont fortement combinatoires et appartiennent à la catégorie des problèmes dits NP-Complets, dont la résolution requiert l'utilisation de techniques appropriées. De nombreuses méthodes de résolution ont été développées en recherche opérationnelle. On peut classer ces méthodes en deux grandes catégories : les méthodes exactes et les méthodes approchées, encore appelées heuristiques.

### **2.2.1 Méthodes exactes**

Le principe essentiel d'une méthode exacte consiste généralement à énumérer, souvent de manière implicite, l'ensemble des solutions de l'espace de recherche. Ce type de méthodes rencontre généralement des difficultés dans les applications de taille importante. En effet, le temps de calcul nécessaire pour trouver une solution risque d'augmenter exponentiellement en fonction de la taille du problème. De façon pratique, seuls les problèmes de petites et moyennes tailles peuvent être résolus de façon optimale

par des algorithmes exacts. De plus, pour certains problèmes, la consommation de l'espace mémoire de ces algorithmes peut être très grande et parfois entraîner l'arrêt prématuré de l'application informatique. Parmi les algorithmes exacts ayant servi à résoudre des problèmes d'optimisation s'apparentant à la confection des horaires du personnel, on retrouve le branch-and-price (B&P) (Zamorano et Stolletz (2017), Boyer *et al.* (2014), Legrain *et al.* (2020), la génération de colonnes (Volland *et al.* (2017), Zeighami et Soumis (2019), Jens *et al.* (2011), Lin et Hsu (2016), les méthodes basées sur la relaxation lagrangienne (Hernandez *et al.* (2019)), la programmation par contraintes (Zhao et Li (2014)), etc. Par exemple, Legrain *et al.* (2020) s'intéresse au problème de confection des horaires des infirmières sur une période de 8 semaines en leur affectant des jours de travail et de repos. L'approche de branch-and-price est utilisée pour résoudre le problème où chaque colonne générée correspond à une suite de jours de travail. Dans Volland *et al.* (2017), la génération de colonnes a permis de résoudre à la fois un problème d'affectation des tâches et des quarts de travail à une nouvelle catégorie d'employés ayant la responsabilité de soulager le travail des infirmières. Ils prennent en charge certaines des tâches non médicales. Un problème de planification des horaires des chirurgies non urgentes et de leur affectation aux salles d'opérations est traité dans Zhao et Li (2014). Ils comparent les solutions et les temps de résolution obtenus par la résolution du modèle de programmation mixte non linéaire et la programmation par contraintes et démontrent que cette dernière est plus performante.

Le problème traité dans ce mémoire a pu être résolu, en partie, par une approche de résolution exacte. En effet, la décomposition du problème global en sous-problèmes a permis de résoudre chacun de façon exacte.

### **2.2.2 Méthodes heuristiques**

Les méthodes heuristiques permettent, quant à elles, d'obtenir rapidement une solution approchée, mais qui n'est pas toujours optimale. On peut également utiliser les heuristiques pour résoudre les problèmes de confection des horaires de travail dans le but de trouver une solution de qualité raisonnable, qui respecte les contraintes dures et,

le plus possible, les contraintes souples, dans des délais aussi courts que possible. Parmi ces méthodes, on trouve : la recherche locale, le recuit simulé, la méthode tabou, les algorithmes génétiques et l'optimisation par colonie de fourmis, etc.

Un problème de confection d'horaires pour les infirmières a été résolu par une approche de recherche locale dans Legrain *et al.* (2015). Une solution pour ce problème multi-objectif est obtenue beaucoup plus rapidement et respecte davantage les préférences des infirmières que la solution manuelle. Perumal *et al.* (2019) présente une méta-heuristique basée sur une méthode de recherche dans un voisinage élargi pour résoudre un problème d'horaires pour des chauffeurs d'autobus. La solution initiale est obtenue à l'aide d'un algorithme « glouton » et ils ont réussi à obtenir de bonnes solutions pour des problèmes de petites et moyenne tailles. Un problème d'horaires de techniciens combiné à l'affectation de routes pour assurer les services d'entretien est résolu par une méthode tabou dans Mathlouthi *et al.* (2021). Plusieurs contraintes sur les compétences requises des techniciens, la disponibilité des pièces ou la durée des journées de travail sont prises en considération. Dans Hernandez-Leandro *et al.* (2019), la relaxation lagrangienne est utilisée pour identifier des quarts de travail prometteurs comme première étape à l'intérieur d'une approche heuristique, afin d'affecter des tâches à des employés. Leur problème est considéré comme un problème de recouvrement (*set covering problem*).

Pour les problèmes de grande taille, les méthodes de résolution basées sur des heuristiques ou sur une combinaison d'approches semblent donner de meilleurs résultats pour la qualité des solutions et pour le temps de résolution. Dans ce mémoire, l'approche de résolution privilégiée comporte une combinaison d'une méthode exacte et d'une approche de décomposition.

## CHAPITRE 3 : DESCRIPTION DU PROBLÈME ET MODÉLISATION

Le problème de confection d'horaires de travail traité dans ce mémoire a pour particularité sa très grande taille puisqu'il s'agit de confectionner un horaire sur une période d'un an. Les besoins en nombre d'employés varient selon la période de l'année. Il importe également de satisfaire les contraintes sur le nombre minimum et maximum de jours de travail et de congé consécutifs. Le fait que le nombre d'employés requis change selon la période de l'année et qu'il faille satisfaire les contraintes sur les jours consécutifs de travail et de congé, génère une complexité accrue durant la transition d'une période à une autre.

### 3.1 Description du problème

Contrairement à ce qu'on retrouve généralement pour la confection des horaires de travail, l'organisation étudiée désire obtenir un horaire de travail sur un long horizon de planification, soit un an. L'année a été divisée en 3 périodes (hiver, été (incluant le printemps), automne) en fonction des besoins en personnel qui varient selon la période de l'année.

L'organisation a établi des besoins en personnel pour chaque jour. Cependant, en pratique et considérant les catégories d'employés et les diverses contraintes, il est presque impossible d'affecter le nombre exact d'employés requis de chaque catégorie, chaque jour. Un intervalle a donc été fixé quant au nombre minimal ( $min_{kp}$ ) et maximal ( $max_{kp}$ ) d'employés de chaque catégorie  $k$  devant être au travail chaque jour de la période  $p$ . Deux catégories génériques d'employés doivent être pris en considération :  $A$  et  $L$ . Une tâche ne peut pas être assignée à un employé qui n'est pas qualifié pour l'effectuer. Par exemple, un employé de la catégorie  $A$  ne peut effectuer le travail d'un employé de la catégorie  $L$ . Chaque employé travaille principalement dans sa catégorie d'origine pour effectuer son travail principal. Cependant, il peut être appelé



à travailler dans une autre catégorie s'il a les compétences requises et que le nombre d'employés d'une autre catégorie est insuffisant pour accomplir le travail.

En plus d'avoir droit à des périodes de vacances, chaque employé doit travailler entre  $cw_{min}$  et  $cw_{max}$  jours consécutifs suivi par entre  $co_{min}$  et  $co_{max}$  jours consécutifs de congé durant toute l'année. Ces exigences rendent le problème plus complexe notamment durant les transitions entre les périodes étant donné que les besoins en nombre d'employés varient selon la catégorie et la période de l'année.

	Jours												
	Hiver					Été					Automne		
Employé	1	2	3	4	5	6	7	8	9	10	11	12	13
1	A	A	A			A	A	A			A	A	A
2	L	L	L	L			L	L	L	L			L
3			A	A	A			A	A	A			A
4	A	A	A			A	A	A			A	A	A
5				L	L	L				L	L	L	L
6				A	A	A			A	A	A		

Tableau 3.1 : Exemple d'horaire fictif avec six employés de catégorie différente.

Le tableau 3.1 présente un exemple d'horaire fictif sur 13 jours et trois périodes. Les jours 1 à 5 représentent la période d'hiver, les jours 6 à 10 la période d'été et les jours 11 à 13, la période d'automne. Les cases en bleu indiquent que l'employé travaille. Chaque employé doit travailler entre 3 à 4 jours consécutifs suivis par 2 à 3 jours consécutifs de congé. Au moins deux employés de catégorie *A* et pas plus de trois sont requis chaque jour. Pour les employés de catégorie *L*, le nombre d'employés requis est entre 1 et 2 chaque jour. Dans cet exemple, l'employé 2 finit la période d'hiver par une journée de congé (jour 5). Il doit être en congé au moins le premier jour de la période d'été (jour 6). Les employés 2 et 3 finissent la période d'automne par un jour de travail (jour 13). Ils devraient donc commencer l'année suivante par au plus trois jours consécutifs de travail.

## 3.2 Modélisation mathématique

Le problème de confection des horaires de travail peut être modélisé mathématiquement comme un problème de programmation linéaire en nombres entiers avec variables binaires. Certaines des contraintes seront définies comme des contraintes souples (contraintes dont la violation est permise mais non souhaitable), ce qui introduira des objectifs additionnels à la fonction à optimiser.

Avant de présenter le modèle mathématique décrivant le problème étudié, nous introduisons les ensembles utilisés, les paramètres et les variables pour permettre la compréhension du modèle.

### Notation

#### Ensembles

$I$  : ensemble des employés.

$I_k$  : ensemble des employés de type  $k$ ,  $k = A, L$ .

$J_p$  : ensemble des jours de la période  $p = H$  (hiver),  $E$  (été),  $A$  (automne).

$J$  : ensemble des jours de l'année.

$P$  : ensemble des périodes,  $H$  (hiver),  $E$  (été),  $A$  (automne).

#### Paramètres

$min_{kp}$  : nombre minimum d'employés de catégorie  $k$  requis par jour à la période  $p$ ,  
 $k = A, L, p \in P$ .

$max_{kp}$  : nombre maximum d'employés de catégorie  $k$  requis par jour à la période  $p$ ,  
 $k = A, L, p \in P$ .

$wmin_{kp}$  : nombre minimum de jours de travail requis à la période  $p$  pour chaque employé de catégorie  $k$ ,  $k = A, L, p \in P$ .

$wmax_{kp}$  : nombre maximum de jours de travail requis à la période  $p$  pour chaque employé de catégorie  $k$ ,  $k = A, L, p \in P$ .

$cw_{min}$  : nombre minimum de jours consécutifs de travail pour chaque employé.

$cw_{max}$  : nombre maximum de jours consécutifs de travail pour chaque employé.

$co_{min}$  : nombre minimum de jours consécutifs de congé pour chaque employé.

$co_{max}$  : nombre maximum de jours consécutifs de congé pour chaque employé.



$M$  : une constante non négative suffisamment grande.

### Variables

$$x_{ij} = \begin{cases} 1 & \text{si l'employé } i \text{ travaille le jour } j, i \in I, j \in J \\ 0 & \text{sinon} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{si l'employé } i \in I \text{ est en congé durant au moins le premier} \\ & \text{jour de la période d'hiver} \\ 0 & \text{sinon} \end{cases}$$

$w_i, z_j$  : variables d'écart par rapport à la satisfaction des contraintes souples.

### Contraintes

Les contraintes du problème peuvent être modélisées par ce qui suit, suivies par la justification et l'explication de chacune.

$$\sum_{i \in I_k} x_{ij} \geq \min_{kp}, \quad j \in J_p, \quad k = A, L, p \in P. \quad (3.1)$$

$$\sum_{i \in I_k} x_{ij} - z_j \leq \max_{kp}, \quad j \in J_p, \quad k = A, L, p \in P. \quad (3.2)$$

$$\sum_{j \in J_p} x_{ij} \geq \min_{kp}, \quad i \in I. \quad (3.3)$$

$$\sum_{j \in J_p} x_{ij} - w_i \leq \max_{kp}, \quad i \in I. \quad (3.4)$$

$$x_{ij} - \sum_{k=1}^{t-1} x_{i(j+k)} + x_{i(j+t)} \geq -t + 2, \quad t = 2 \dots, cw_{\min}, \quad i \in I, j \in J. \quad (3.5)$$

$$\sum_{j=1}^{r+cw_{\max}} x_{ij} \leq cw_{\max}, \quad 1 \leq r \leq |J| - cw_{\max}, \quad i \in I. \quad (3.6)$$

$$-x_{ij} + \sum_{k=1}^{t-1} x_{i(j+k)} - x_{i(j+t)} \geq -1, t = 2.., co_{min}, i \in I, j \in \{1, \dots, |J| - t\} \quad (3.7)$$

$$\sum_{j=1}^{r+co_{max}} (1 - x_{ij}) \leq co_{max}, 1 \leq r \leq |J| - co_{max}, \quad i \in I \quad (3.8)$$

$$x_{ij} \in \{0,1\} \quad i \in I, j \in J. \quad (3.9)$$

Les contraintes dures, devant être satisfaites obligatoirement, se répartissent en trois catégories : nombre minimum d'employés requis par jour, nombre minimum de jours de travail requis par période, et séquences des jours de travail et de congé.

Les contraintes souples, qui peuvent être violées, se répartissent en deux catégories : nombre maximum d'employés requis par jour et nombre maximum de jours de travail requis par période.

L'organisation a fixé un intervalle pour le nombre d'employés requis par jour. Il doit donc y avoir entre  $min_{kp}$  et  $max_{kp}$  employés chaque jour. Le principe est le même pour le nombre de jours de travail et les séquences des jours de travail et de congé. Les contraintes (3.1) et (3.2) assurent d'avoir entre  $min_{kp}$  et  $max_{kp}$  employés respectivement chaque jour pour chaque période  $p$ . Cependant, la contrainte (3.2) est une contrainte souple puisqu'il est accepté d'avoir plus d'employés que le nombre maximum requis. Les contraintes (3.3) et (3.4) font que chaque employé  $i \in I_k$  travaille entre  $wmin_{kp}$  et  $wmax_{kp}$  jours respectivement pour chaque période  $p$ . Il est cependant accepté qu'un employé travaille plus que  $wmax_{kp}$  jours à une période  $p$  (contrainte (3.4)). Les variables d'écarts  $w_i$  et  $z_j$  sont donc présentes dans les contraintes souples (3.2) et (3.4) respectivement. Elles seront ajoutées à la fonction objectif qui vise à minimiser le nombre de contraintes violées.

La contrainte (3.5) permet d'avoir au minimum  $cw_{min}$  jours de travail consécutifs entre les jours de congé et ce, pour chaque employé  $i \in I$ . La contrainte (3.5) vise, en fait, à empêcher qu'il y ait un nombre insuffisant de jours consécutifs de travail et, par

conséquent, à ce que l'employé travaille au moins  $cw_{min}$  jours consécutifs. Illustrons le fonctionnement de la contrainte (3.5) à l'aide de l'exemple 1.

### Exemple 1 :

Supposons que l'employé  $i$  doit travailler au moins 3 jours consécutifs ( $cw_{min} = 3$ ).

Lorsque  $t = 2$ , la contrainte (3.5) s'écrit :  $x_{ij} - x_{i(j+1)} + x_{i(j+2)} \geq 0$ .

On ne peut donc avoir  $x_{ij} = x_{i(j+2)} = 0$  et  $x_{i(j+1)} = 1$  pour satisfaire la contrainte. Cela implique que l'employé  $i$  ne peut travailler un seul jour de travail entre 2 jours de congé.

Lorsque  $t = 3$ , la contrainte (3.5) s'écrit :  $x_{ij} - x_{i(j+1)} - x_{i(j+2)} + x_{i(j+3)} \geq -1$ .

On ne peut donc avoir  $x_{ij} = x_{i(j+3)} = 0$  et  $x_{i(j+1)} = x_{i(j+2)} = 1$  pour satisfaire la contrainte. Cela implique que l'employé  $i$  ne peut travailler deux jours consécutifs de travail entre 2 jours de congé.

La contrainte (3.5) force donc l'employé  $i$  à travailler au moins 3 jours consécutifs tel que requis.

La contrainte (3.6) oblige un employé à travailler au plus  $cw_{max}$  jours de travail consécutifs. La contrainte (3.7) permet d'avoir au minimum  $co_{min}$  jours de congé consécutifs entre les jours de travail et ce, pour chaque employé  $i \in I$ . La contrainte (3.7) vise, en fait, à empêcher qu'il y ait un nombre insuffisant de jours consécutifs de congé et, par conséquent, à ce que l'employé soit en congé au moins  $co_{min}$  jours consécutifs. Le fonctionnement des contrainte (3.6) et (3.7) sera illustré simultanément à l'aide de l'exemple suivant.

### Exemple 2 :

Supposons que l'employé  $i$  doit travailler au plus 4 jours consécutifs ( $cw_{max} = 4$ ) et doit être en congé au moins 3 jours consécutifs ( $co_{min} = 3$ ) sur une période de  $|J| = 7$  jours.  $r$  varie alors de 1 à  $|J| - cw_{max} = 7 - 4 = 3$ .

Lorsque  $r = 1$ , la contrainte (3.6) s'écrit :  $x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i5} \leq 4$ .

Donc, au plus, 4 des variables présentes dans la contrainte peuvent prendre la valeur 1. Théoriquement, on pourrait avoir  $x_{i1} = x_{i2} = x_{i4} = x_{i5} = 1$  et  $x_{i3} = 0$ , c'est -à-dire une seule journée de congé entre des jours de travail. Cependant, les contraintes (3.6) combinées aux contraintes (3.7) sur le nombre minimum de jours consécutifs de congé empêchent une telle solution.

En effet, lorsque  $t = 2$  et  $j = 1, 2$ , les contraintes (3.7) s'écrivent :

$$-x_{i1} + x_{i2} - x_{i3} \geq -1 \text{ et } -x_{i2} + x_{i3} - x_{i4} \geq -1.$$

On ne peut donc avoir  $x_{i1} = x_{i3} = 1$  et  $x_{i2} = 0$  par la première contrainte (3.7). Par la deuxième contrainte (3.7), on ne peut avoir  $x_{i2} = x_{i4} = 1$  et  $x_{i3} = 0$ , soit une seule journée de congé entre les jours de travail.

Lorsque  $r = 2$ , la contrainte (3.6) s'écrit :  $x_{i1} + x_{i2} + x_{i3} + x_{i4} + x_{i5} + x_{i6} \leq 4$ .

Au plus, 4 des variables présentes dans la contrainte peuvent prendre la valeur 1. Théoriquement, on pourrait avoir :  $x_{i1} = x_{i2} = x_{i5} = 1$  et  $x_{i3} = x_{i4} = 0$ , c'est-à-dire deux journées de congé entre des jours de travail. Cependant, les contraintes (3.6) combinées aux contraintes (3.7) sur le nombre minimum de jours consécutifs de congé empêchent une telle solution.

En effet, lorsque  $t = 3$  et  $j = 1, 2$ , les contraintes (3.7) s'écrivent :

$$-x_{i1} + x_{i2} + x_{i3} - x_{i4} \geq -1 \text{ et } -x_{i2} + x_{i3} + x_{i4} - x_{i5} \geq -1.$$

On ne peut donc avoir  $x_{i1} = x_{i4} = 1$  et  $x_{i2} = x_{i3} = 0$  par la première contrainte (3.7). Par la deuxième contrainte (3.7), on ne peut avoir  $x_{i2} = x_{i5} = 1$  et  $x_{i3} = x_{i4} = 0$ , soit deux journées de congé entre les jours de travail. L'employé  $i$  ne peut donc être en congé deux jours consécutifs entre 2 jours de travail. La contrainte (3.7) force donc l'employé  $i$  à être en congé au moins 3 jours consécutifs.

La contrainte (3.8) oblige un employé à être en congé durant au plus  $co_{max}$  jours de congé consécutifs. Illustrons le fonctionnement de la contrainte (3.8) à l'aide de l'exemple 4.

#### Exemple 4 :

Supposons que l'employé  $i$  doit être en congé au plus 4 jours consécutifs ( $co_{max} = 4$ )

et que  $|J| = 7$ .  $r$  varie de 1 à  $|J| - co_{max} = 7 - 4 = 3$

Lorsque  $r = 1$ , la contrainte (3.8) s'écrit :

$$(1 - x_{i1}) + (1 - x_{i2}) + (1 - x_{i3}) + (1 - x_{i4}) + (1 - x_{i5}) \leq 4.$$

Au plus, 4 des variables présentes dans la contrainte peuvent prendre la valeur 0.

Théoriquement, on pourrait avoir  $x_{i1} = x_{i2} = x_{i4} = x_{i5} = 0$  et  $x_{i3} = 1$ , c'est-à-dire une

seule journée de travail entre des jours de congé. Cependant, les contraintes (3.8) combinées aux contraintes (3.5) sur le nombre minimum de jours consécutifs de travail empêchent une telle solution.

Lorsque  $r = 2$ , la contrainte (3.8) s'écrit :

$$(1-x_{i1}) + (1-x_{i2}) + (1-x_{i3}) + (1-x_{i4}) + (1-x_{i5}) + (1-x_{i6}) \leq 4.$$

Au plus, 4 des variables présentes dans la contrainte peuvent prendre la valeur 0. Théoriquement, on pourrait avoir  $x_{i1} = x_{i2} = x_{i5} = 0$  et  $x_{i3} = x_{i4} = 1$ , c'est-à-dire deux jours de travail entre des jours de congé. Cependant, les contraintes (3.8) combinées aux contraintes (3.5) sur le nombre minimum de jours consécutifs de travail empêchent également une telle solution.

### Fonction objectif

$$\text{Min } \alpha \sum_{j \in J} z_j + \beta \sum_{i \in I} w_i$$

La fonction objectif consiste à minimiser la somme des écarts par rapport à la satisfaction des contraintes souples. Chaque contrainte peut être stricte ou souple. Les contraintes strictes ne peuvent pas être violées, alors que les contraintes souples peuvent l'être. Une pénalité est associée à chaque contrainte souple. Les pénalités vont croître proportionnellement avec la déviation par rapport à la contrainte. Considérant que toutes les contraintes souples n'ont pas la même importance, des poids différents peuvent être associés à chacun des écarts de ces contraintes.  $\alpha$  et  $\beta$  sont des constantes entières représentant le poids associé aux écarts des contraintes sur le nombre d'employés requis par jour (contrainte (3.2)) et les contraintes sur le nombre de jours de travail par période (contrainte (3.4)) respectivement.

Prenons, par exemple, la contrainte (3.4) qui impose au maximum 48 jours de travail pour la période d'hiver et une pénalité  $\beta$  de 10. Le coût dans la fonction objectif sera de 10 fois le nombre de jours de travail à la période d'hiver excédant les 48 jours requis. Dans ce cas,  $w_i \neq 0$ .

La pénalité est ajoutée à la fonction objectif autant de fois qu'il y a de jours de travail en surplus des 48 jours requis et ce, par période. Si un employé a travaillé 49 jours (un jour de plus que le nombre maximal permis) durant la période d'hiver et 39 jours du 1 juillet au 31 août (un jour de plus que le nombre maximal permis), une pénalité totale de  $10(1) + 10(1) = 20$  sera ajoutée à la fonction objectif. Le même principe s'applique pour les contraintes (3.2), qui limitent le nombre d'employés travaillant chaque jour, et la pénalité  $\alpha$ .

Le modèle global permet de générer un horaire personnalisé des employés. Ce modèle permet d'avoir un horaire optimal, par rapport aux contraintes d'origine, lorsque toutes les variables d'écart sont nulles à l'optimum.

Ce modèle fait qu'à la fin de l'année (fin de l'horizon de planification), le nombre de jours consécutifs de travail et de congé peut ne pas être respecté. Cela ne pose pas un problème puisque ces jours seront pris en considération pour la confection des horaires de travail de l'année suivante.

## CHAPITRE 4 : METHODES DE RESOLUTION ET EXPERIMENTATION

Le problème à résoudre est un problème de programmation linéaire en nombres entiers de grande taille. Les méthodes de résolution privilégiées pour ce type de problème sont principalement des méthodes heuristiques. Puisque le problème a été divisé en plusieurs sous-problèmes, selon les périodes, nous avons testé une méthode exacte et une approche de décomposition.

### 4.1 Résolution par la méthode exacte

Afin de résoudre le problème décrit à la section 3.2, nous avons utilisé le solveur CPLEX 12.10. La figure 4.1 présente le schéma du système de résolution par CPLEX.

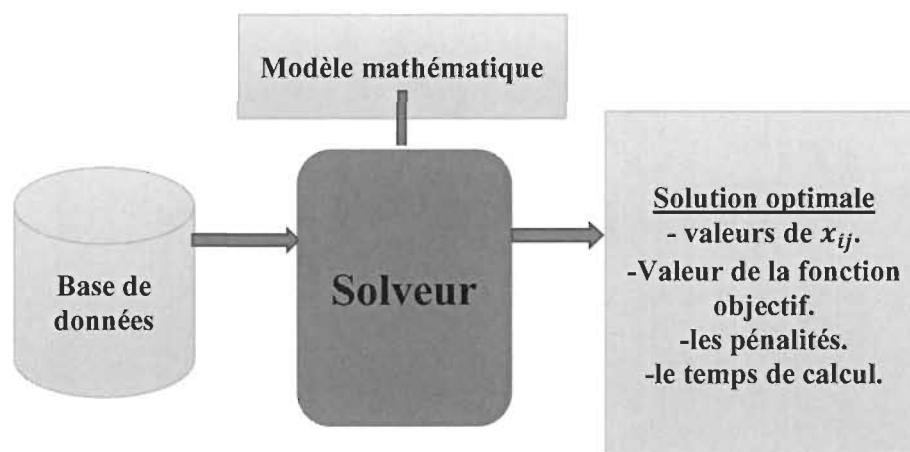


Figure 4-1 : Schéma du système de résolution par CPLEX.

Pour que CPLEX fonctionne correctement, deux fichiers sont nécessaires. Le premier comprend le modèle mathématique basé sur la syntaxe OPL de CPLEX. Le second est le fichier de données qui contient les informations relatives aux employés et aux

exigences de l'organisation. Les problèmes deviennent impossibles à résoudre de façon exacte en un temps raisonnable lorsque la taille devient trop grande.

Des tests numériques ont été réalisés sur CPLEX afin de résoudre le problème global, sur une période d'un an. La solution optimale a été obtenue après plus de quatre heures de temps d'exécution, ce qui n'est pas nécessairement anormal pour un problème de cette taille. La taille de problème a donc un impact important sur la consommation de l'espace mémoire. Il a donc été possible d'obtenir une solution optimale mais en un temps de résolution élevé.

Par conséquent, une approche de résolution basée sur la décomposition du problème en autant de sous-problèmes que de périodes (approche de décomposition) a été développée. Cette approche permet de diminuer considérablement les temps de calcul.

## **4.2 Approche de décomposition**

Pour résoudre notre problème de planification d'horaires en un temps raisonnable, il faut réduire la taille du modèle mathématique. Les contraintes sur le nombre de jours de travail requis pour chaque période et chaque employé, les séquences de jours de travail et de congé et le nombre d'employés requis par jour sont reliées entre elles à travers les périodes. Afin de décomposer le problème global en sous-problèmes associés à chacune des périodes, le modèle mathématique a dû être modifié.

### **4.2.1 Processus de décomposition**

Dans l'organisation étudiée, certaines périodes de l'année sont plus occupées que d'autres. Par exemple, les services sont plus en demande début décembre. Il faut donc ajuster le nombre d'employés présents en conséquence. C'est la raison pour laquelle, l'année est divisée en périodes de longueur variée, chacune requérant un nombre précis d'employés par jour. Par exemple, durant la période d'été (du 1 juillet au 31 août), il doit y avoir au moins 30 employés chaque jour alors que, du 9 au 21 décembre (13 jours), il doit y avoir au moins 43 employés chaque jour. Considérant les contraintes sur le nombre de jours de travail et de congé consécutifs, la transition entre les périodes ajoute à la difficulté de respecter le nombre requis d'employés par jour.



Les trois périodes considérées par l'organisation étudiée sont :

- **Hiver** : du 1 janvier au 30 mars (89 jours)
- **Été** : du 31 mars au 31 août (154 jours)
- **Automne** : du 1 septembre au 31 décembre (122 jours).

Ces trois périodes permettent de générer des sous problèmes pouvant être résolus indépendamment en un temps de résolution raisonnable. Cependant, pour passer d'une période à l'autre, la solution de la période précédente sera prise en considération, notamment pour respecter les contraintes sur les jours consécutifs de travail et de congé.

#### **4.2.2 Résolution du problème pour la période de l'hiver**

Le premier problème à résoudre est celui de la période d'hiver. En principe, il faudrait utiliser la solution de la fin de l'année précédente (fin de l'horizon de planification de la période d'automne de l'année précédente) dans le but de respecter les contraintes du nombre de jours consécutifs de travail et de congé. Comme les données de l'année précédente n'étaient pas disponibles, on a décidé de partir à zéro (construire un horaire partiel comme solution de départ). Cependant, pour le passage de la période d'hiver à la période d'été et de l'été à l'automne, l'approche de résolution sera celle qui aurait été utilisée si l'horaire de l'année précédente avait été disponible. Il n'y a donc pas perte de généralités.

##### **Génération d'un horaire initial**

Un horaire initial est généré pour les 20 premiers jours de la période d'hiver pour l'ensemble des 55 employés. Le nombre de jours choisi pour l'horaire initial permet à la fois de couvrir un nombre adéquat de jours et de faciliter la transition pour la suite de la période d'hiver. L'horaire initial sert à déterminer ensuite l'horaire pour le reste de l'année. En ce sens, il peut avoir un certain impact sur l'horaire global, notamment en lien avec les contraintes fixant le nombre de jours consécutifs de congé et de travail.

Les contraintes sur le nombre minimal et maximal de jours de travail requis par période, par employé, (contraintes (3.3) et (3.4)), n'ont pas été incluses pour ces 20 jours du

début de la période d'hiver. Ces contraintes impliquent que chaque employé doit travailler entre 35 et 48 jours durant la période d'hiver qui est d'une durée de 89 jours. En considérant uniquement les premiers 20 jours, il faudrait préciser un peu artificiellement un nombre de jours de travail requis qui ne reflèterait pas la réalité.

Un horaire de travail pour les 20 premiers jours de l'hiver a été obtenu à l'aide de CPLEX. Il servira de solution initiale pour la deuxième phase de la résolution du problème de la période d'hiver, ce qui permettra de réduire le temps d'exécution.

Le modèle mathématique a été modifié comme suit (figure 4.2) pour l'obtention de la solution de départ.

Soient  $J = \{1, \dots, 20\}$  (ensemble des jours du début de la période d'hiver),  $I = \{1, \dots, 55\}$  (ensemble des employés).

$$\text{Min } \alpha \sum_{j \in J} z_j$$

$$\sum_{i \in I_k} x_{ij} \geq \min_{kp}, \quad j \in \{1, \dots, 20\}, \quad k = A, L, p$$

$$\in P. \quad (4.1)$$

$$\sum_{i \in I_k} x_{ij} - z_j \leq \max_{kp}, \quad j \in \{1, \dots, 20\}, \quad k = A, L, p$$

$$\in P. \quad (4.2)$$

$$x_{ij} - \sum_{k=1}^{t-1} x_{i(j+k)} + x_{i(j+t)} \geq -t + 2, \quad t = 2 \dots, cw_{\min},$$

$$i \in I, j \in \{1, \dots, 20\} \quad (4.5)$$

$$\sum_{j=1}^{r+cw_{\max}} x_{ij} \leq cw_{\max}, \quad 1 \leq r \leq |\{1, \dots, 20\}| - cw_{\max}, \quad i \in I. \quad (4.6)$$

$$-x_{ij} + \sum_{k=1}^{t-1} x_{i(j+k)} - x_{i(j+t)} \geq -1, \quad t = 2 \dots, co_{\min},$$

$$i \in I, j \in \{1, \dots, 20 - t\}. \quad (4.7)$$

$$\sum_{j=1}^{r+co_{\max}} (1 - x_{ij}) \leq co_{\max}, \quad 1 \leq r \leq |\{1, \dots, 20\}| - co_{\max}, \quad i \in I. \quad (4.8)$$

$$\sum_{j=1}^{cw_{\min}} x_{ij} \geq cw_{\min} - M * y_i, \quad \forall i \in I \quad (4.9)$$

$$\sum_{j=1}^{co_{\min}} x_{ij} \leq 0 + M * (1 - y_i), \quad \forall i \in I \quad (4.10)$$

$$x_{ij} \in \{0,1\} \quad i \in I, j \in J \quad (4.11)$$

Figure 4.2 : Modèle ajusté pour l'obtention d'un horaire initial.

La fonction objectif, dans ce cas, ne prend plus en considération les pénalités sur les contraintes concernant le nombre minimal et maximal de jours de travail par employé puisqu'on a supprimé les contraintes (3.3) et (3.4). Elle minimise donc seulement les pénalités associées aux contraintes sur le nombre maximum d'employés requis par jour.

Afin de s'assurer que chaque employé travaille au moins  $cw_{min}$  jours de travail consécutifs et est en congé durant au moins  $co_{min}$  jours consécutifs et ce, pour le début de la période d'hiver, des modifications ont été apportées au modèle mathématique. Sans ces modifications, les contraintes (4.5) et (4.7) n'auraient pu être réalisables. Les contraintes (4.9) et (4.10) et la variable  $y_i$  ont été ajoutées afin de prendre en considération l'horaire du début de la période d'hiver pour générer l'horaire de l'ensemble de cette période.

Un employé sera soit en congé, soit au travail au début de la période d'hiver. La variable  $y_i$  permet donc de considérer ces deux possibilités.

$$y_i = \begin{cases} 1 & \text{si l'employé } i \in I \text{ est en congé durant au moins le premier} \\ & \text{jour de la période d'hiver} \\ 0 & \text{sinon} \end{cases}$$

Si l'employé  $i$  travaille durant les premiers jours de la période d'hiver,  $y_i = 0$ , la contrainte (4.9),  $\sum_{j=1}^{cw_{min}} x_{ij} \geq cw_{min} - M * y_i$  devient

$$\sum_{j=1}^{cw_{min}} x_{ij} \geq cw_{min}, \quad \forall i \in I$$

Cette contrainte force l'employé  $i \in I$  à travailler au minimum  $cw_{min}$  jours consécutifs.

Si l'employé  $i$  est en congé durant les premiers jours de la période d'hiver,  $y_i = 1$ , la contrainte (4.10),  $\sum_{j=1}^{co_{min}} x_{ij} \leq 0 + M * (1 - y_i)$ , devient

$$\sum_{j=1}^{co_{min}} x_{ij} \leq 0, \quad \forall i \in I$$

Cette contrainte force l'employé  $i \in I$  à être en congé au minimum  $co_{min}$  jours consécutifs.

L'ajout de ces deux contraintes combinées aux contraintes (4.5) et (4.7) assurent le respect du nombre minimal du nombre de jours consécutifs de travail et de congé.

### Résolution du sous-problème de la période d'hiver

La période d'hiver comprend les jours 1 à 89. L'horaire initial couvre les 20 premiers jours de la période d'hiver. Le modèle mathématique doit être ajusté pour prendre en considération la solution de l'horaire initial (pour les 20 premiers jours) et pour que certaines contraintes ne soient appliquées que pour les jours 21 à 89. Les modifications au modèle mathématique sont en caractères gras dans la figure 4.3. Les  $x_{ij}^1$  correspondent à la solution de l'horaire initial soit pour les jours 1 à 20 pour chaque employé.

$$\text{Min } \alpha \sum_{j \in J} z_j + \beta \sum_{i \in I} w_i$$

$$x_{ij} = x_{ij}^1, i \in I, j \in \{1, \dots, 20\} \quad (4.12)$$

$$\sum_{i \in I_k} x_{ij} \geq \min_{kp}, \quad j \in \{21, \dots, 89\}, k = A, L, p \in P. \quad (4.13)$$

$$\sum_{i \in I_k} x_{ij} - z_j \leq \max_{kp}, \quad j \in \{21, \dots, 89\}, k = A, L, p \in P. \quad (4.14)$$

$$\sum_{j \in \{1, \dots, 89\}} x_{ij} \geq \min_{kp}, \quad i \in I. \quad (4.15)$$

$$\sum_{j \in \{1, \dots, 89\}} x_{ij} - w_i \leq \max_{kp}, \quad i \in I. \quad (4.16)$$

$$x_{ij} - \sum_{k=1}^{t-1} x_{i(j+k)} + x_{i(j+t)} \geq -t + 2, t = 2 \dots, cw_{\min}, i \in I, j \in \{1, \dots, 89\} \quad (4.17)$$

$$\sum_{j=1}^{r+cw_{\max}} x_{ij} \leq cw_{\max}, \quad 1 \leq r \leq |\{1, \dots, 89\}| - cw_{\max}, \quad i \in I \quad (4.18)$$

$$-x_{ij} + \sum_{k=1}^{t-1} x_{i(j+k)} - x_{i(j+t)} \geq -1, t = 2 \dots, co_{\min}, i \in I, j \in |\{1, \dots, 89\}| - t \quad (4.19)$$

$$\sum_{j=1}^{r+co_{\max}} (1 - x_{ij}) \leq co_{\max}, 1 \leq r \leq |\{1, \dots, 89\}| - co_{\max}, \quad i \in I \quad (4.20)$$

$$x_{ij} \in \{0,1\} \quad i \in I, j \in \{1, \dots, 89\}. \quad (4.21)$$

Figure 4.3 : Modèle ajusté pour la résolution du problème de la période d'hiver.

La contrainte (4.12) fixe les valeurs des  $x_{ij}$  correspondant à la solution des 20 premiers jours.

Pour les contraintes (4.13) et (4.14) sur le nombre minimal et maximal d'employés requis par jour, on considère seulement les jours 21 à 89 puisqu'on a déjà la solution des jours 1 à 20. Cela permet de réduire le temps de calcul. Pour les contraintes (4.15) à (4.20), on doit prendre en considération les 89 jours de la période d'hiver afin de respecter les contraintes sur le nombre de jours de travail de la période d'hiver et les séquences des jours de travail et de congé pour chaque employé mais en fixant les valeurs des  $x_{ij}$  en fonction de la solution initiale soit pour les 20 premiers jours.

#### **4.2.3 Résolution du problème pour la période de l'été**

Pour construire l'horaire de la période d'été, on doit prendre en considération l'horaire de travail de la fin de la période de l'hiver, dans le but de respecter l'ensemble des contraintes du modèle global sur les séquences des jours de travail et de congé. Par exemple, si l'employé finit la période d'hiver avec 5 jours consécutifs de travail, il devra continuer à travailler au moins 10 jours consécutifs et au plus 14 pour respecter l'intervalle des jours consécutifs de travail qui est entre 15 et 19.

La période de transition entre l'hiver et l'été doit inclure la solution d'un certain nombre de jours de la fin de l'hiver, mais pas tous les jours de l'hiver. La solution des 20 derniers jours de l'hiver suffit pour passer à la résolution du sous-problème de la période d'été. En effet, on doit prendre au moins 19 jours puisque le nombre maximum de jours consécutifs de travail est égal à 19. En prenant 20 jours, on s'assure d'avoir le nombre minimal requis tout en ayant une marge d'un jour pour la transition. Par exemple, si on considérait les 14 derniers jours de l'hiver et qu'un employé finissait la période de l'hiver avec plus de 14 jours de travail consécutifs (disons 17 jours), l'employé devrait travailler au moins 1 jour et au plus 5 jours consécutifs pour respecter l'intervalle des jours consécutifs de travail qui est entre 15 et 19. En réalité, comme il a travaillé 17 jours consécutifs, cela pourrait l'amener à travailler jusqu'à 22 jours consécutifs, alors que l'horaire de début de l'été devrait le faire travailler au plus 2 jours consécutifs. Donc, en considérant 14 jours au lieu de 20, les contraintes sur le nombre de jours consécutifs de travail et de congé pourraient ne pas être respectées avec un

nombre insuffisant de jours pour la solution de la fin de la période d'hiver. Donc, seules ces contraintes doivent être adaptées puisque les contraintes sur le nombre de jours de travail de la période d'été et le nombre d'employés requis par jour pour cette période ne dépendent pas de la solution de la période d'hiver.

Les jours de la fin de la période d'hiver considérés pour le sous-problème de la période d'été sont les jours 70 à 89 (les 20 derniers jours de l'hiver). Les jours 90 à 243 (tel que fixés par l'organisation à l'étude) sont les jours de la période d'été.

Le modèle mathématique doit être ajusté pour prendre en considération la solution de l'horaire de la fin de la période d'hiver (20 derniers jours) et pour que certaines contraintes ne soient appliquées que pour les jours 90 à 243. Les modifications au modèle mathématique sont en rouge dans la figure 4.4. Les  $x_{ij}^2$  correspondent à la solution de la fin de la période d'hiver soit pour les jours 70 à 89 et chaque employé.



$$\text{Min } \alpha \sum_{j \in \{90, \dots, 243\}} z_j + \beta \sum_{i \in I} w_i$$

$$x_{ij} = x_{ij}^2, i \in I, j \in \{70, \dots, 89\} \quad (4.22)$$

$$\sum_{i \in I_k} x_{ij} \geq \min_{kp}, \quad j \in \{90, \dots, 243\}, k = A, L, p \in P. \quad (4.23)$$

$$\sum_{i \in I_k} x_{ij} - z_j \leq \max_{kp}, \quad j \in \{90, \dots, 243\}, k = A, L, p \in P. \quad (4.24)$$

$$\sum_{j \in \{90, \dots, 243\}} x_{ij} \geq \min_{kp}, \quad i \in I. \quad (4.25)$$

$$\sum_{j \in \{90, \dots, 243\}} x_{ij} - w_i \leq \max_{kp}, \quad i \in I. \quad (4.26)$$

$$x_{ij} - \sum_{k=1}^{t-1} x_{i(j+k)} + x_{i(j+t)} \geq -t + 2, \quad t = 2 \dots, cw_{\min},$$

$$i \in I, j \in \{70, \dots, 243\} \quad (4.27)$$

$$\sum_{j=1}^{r+cw_{\max}} x_{ij} \leq cw_{\max}, \quad 1 \leq r \leq |\{70, \dots, 243\}| - cw_{\max}, \quad i \in I. \quad (4.28)$$

$$-x_{ij} + \sum_{k=1}^{t-1} x_{i(j+k)} - x_{i(j+t)} \geq -1, t = 2 \dots, co_{\min}, i \in I,$$

$$j \in |\{70, \dots, 243\}| - t. \quad (4.29)$$

$$\sum_{j=1}^{r+co_{\max}} (1 - x_{ij}) \leq co_{\max}, 1 \leq r \leq |\{70, \dots, 243\}| - co_{\max}, \quad i \in I. \quad (4.30)$$

$$x_{ij} \in \{0,1\} \quad i \in I, j \in \{70, \dots, 243\}. \quad (4.31)$$

Figure 4.4 : Modèle ajusté pour la résolution du problème de la période d'été.

La contrainte (4.22) fixe les valeurs des  $x_{ij}$  correspondant à la solution des 20 derniers jours de la période d'hiver.

Les contraintes (4.23) à (4.26) sur le nombre d'employés requis par jour et le nombre de jours de travail requis durant l'été, respectivement, ne dépendent pas de la solution de la période de l'hiver. On peut donc faire varier l'indice  $j$  de 90 à 243 jours (seulement la période de l'été).

Pour les contraintes (4.27) à (4.30), les jours considérés incluent les 20 derniers jours de la période de l'hiver ( $j \in \{70, \dots, 243\}$ ).

#### **4.2.4 Résolution du problème pour la période d'automne**

Les jours de la fin de la période d'été considérés pour le sous-problème de la période d'automne sont les jours 224 à 243. Les jours 244 à 365 sont les jours de la période d'automne.

Le modèle mathématique doit être ajusté pour prendre en considération la solution de l'horaire de la fin de période d'été (20 derniers jours) et pour que certaines contraintes ne soient appliquées que pour les jours 244 à 365. Les modifications au modèle mathématique sont en caractères gras dans la figure 4.5. Les  $x_{ij}^3$  correspondent à la solution de la fin de la période d'été soit pour les jours 224 à 243 et chaque employé.

$$\text{Min } \alpha \sum_{j \in \{244, \dots, 365\}} z_j + \beta \sum_{i \in I} w_i$$

$$x_{ij} = x_{ij}^3, i \in I, j \in \{224, \dots, 243\} \quad (4.32)$$

$$\sum_{i \in I_k} x_{ij} \geq \min_{kp}, \quad j \in \{244, \dots, 365\}, k = A, L, p \in P. \quad (4.33)$$

$$\sum_{i \in I_k} x_{ij} - z_j \leq \max_{kp}, \quad j \in \{244, \dots, 365\}, k = A, L, p \in P. \quad (4.34)$$

$$\sum_{j \in \{244, \dots, 365\}} x_{ij} \geq \min_{kp}, \quad i \in I. \quad (4.35)$$

$$\sum_{j \in \{244, \dots, 365\}} x_{ij} - w_i \leq \max_{kp}, \quad i \in I. \quad (4.36)$$

$$x_{ij} - \sum_{k=1}^{t-1} x_{i(j+k)} + x_{i(j+t)} \geq -t + 2, t = 2 \dots, cw_{\min},$$

$$i \in I, j \in \{224, \dots, 365\}. \quad (4.37)$$

$$\sum_{j=1}^{r+cw_{\max}} x_{ij} \leq cw_{\max}, \quad 1 \leq r \leq |\{224, \dots, 365\}| - cw_{\max}, \quad i \in I. \quad (4.38)$$

$$-x_{ij} + \sum_{k=1}^{t-1} x_{i(j+k)} - x_{i(j+t)} \geq -1 \quad t = 2 \dots, co_{\min}, i \in I, j$$

$$\in |\{224, \dots, 365\}| - t. \quad (4.39)$$

$$\sum_{j=1}^{r+co_{\max}} (1 - x_{ij}) \leq co_{\max}, \quad 1 \leq r \leq |\{224, \dots, 365\}| - co_{\max}, \quad i \in I. \quad (4.40)$$

$$x_{ij} \in \{0,1\} \quad i \in I, j \in \{224, \dots, 365\}. \quad (4.41)$$

Figure 4.5 : Modèle ajusté pour la résolution du problème de la période d'automne.

La contrainte (4.32) fixe les valeurs des  $x_{ij}$  correspondant à la solution des 20 derniers jours de la période d'été.

Les contraintes (4.33) à (4.36) sur le nombre d'employés requis par jour et le nombre de jours de travail requis durant l'automne, respectivement, ne dépendent pas de la solution de la période de l'été. On peut donc faire varier l'indice  $j$  de 244 à 365 jours (les jours de la période de l'automne). Pour les contraintes (4.37) à (4.40), les jours considérés incluent les 20 derniers jours de la période d'été ( $j \in \{224, \dots, 365\}$ ).

Les programmes pour le logiciel CPLEX pour chaque période sont présentés en détail dans les annexes A à D.

## 4.3 Tests et résultats

Dans cette section, nous présentons les résultats obtenus suite à la résolution du problème d'optimisation.

Les données et les tableaux des résultats sont présentés dans un premier temps. Par la suite, les résultats et l'analyse des tests numériques sont détaillés. Cela permet de mesurer la performance de l'approche de décomposition pour l'obtention des horaires de travail.

### 4.3.1 Données et résultats

Pour chaque période, le nombre de jours, le nombre requis de jours de travail, de jours consécutifs de travail et de congé ainsi que le nombre d'employés sont présentés dans le tableau 4.1. Rappelons que les employés sont divisés en 2 catégories, A et L. Chaque jour, un nombre minimal et maximal d'employés requis est défini. Cet intervalle varie selon les périodes de l'année. Au moins 3 employés de catégorie L et au plus 7 doivent être présents chaque jour pour toutes les périodes. Certaines périodes de l'année sont plus occupées que d'autres et exigent la présence d'un plus grand nombre d'employés. En général, le nombre d'employés requis par jour est égal à 36. Cependant, durant la période d'été (du 1 juillet au 31 août), il doit y avoir au moins 25 employés et au plus

33 chaque jour, alors que du 9 au 21 décembre (période de Noël), il doit y avoir au moins 36 employés et au plus 43 chaque jour.

	Nombre requis							
	Employés		Jours de travail		Jours consécutifs de travail		Jours consécutifs de congé	
Période	Min	Max	Min	Max	Min	Max	Min	Max
Hiver (89 jours)	25	36	44	47	15	19	11	15
Été (154 jours)								
31 mars au 30 avril	25	36	10	18	15	19	11	15
1 mai au 30 juin	25	36	30	37	15	19	11	15
1 juillet au 31 août	30	33	35	38	15	19	11	15
Automne (122 jours)								
	25	36	60	73	15	19	11	15
9 au 21 décembre	36	43						

Tableau 4.1 : Valeur des paramètres

### 4.3.2 Tests numériques

Plusieurs tests numériques ont été réalisés afin d'évaluer la sensibilité du modèle aux variations des valeurs de certains paramètres, notamment les pénalités sur le nombre de jours de travail requis par période ainsi que le nombre d'employés requis par jour. Nous avons considéré un test sur la méthode exacte qui est la résolution du problème global et 4 possibilités de la méthode de décomposition.

- T0 : Méthode exacte : pénalités égales ( $\alpha = \beta = 1$ ) (résolution du problème global).
- T1 : Méthode de décomposition : pénalités égales ( $\alpha = \beta = 1$ ).
- T2 : Méthode de décomposition : division de la période d'été en deux sous périodes avec des pénalités égales ( $\alpha = \beta = 1$ ).
- T3 : Méthode de décomposition : augmentation de la valeur de la pénalité  $\alpha$  de 1 à 10 et maintien de la valeur de la pénalité  $\beta$  à 1 ( $\alpha = 10, \beta = 1$ ).
- T4 : Méthode de décomposition : maintien de la valeur de la pénalité  $\alpha$  à 1 et augmentation de la valeur de la pénalité  $\beta$  de 1 à 10 ( $\alpha = 1, \beta = 10$ ).

À des fins de comparaison, le temps d'exécution, la valeur de la fonction objectif et le nombre de contraintes insatisfaites sont répertoriés pour chaque test.

Les résultats des tests T2, T3 et T4 sont comparés au test T1. Donc, si l'un des tests génère une solution meilleure que celle de T1, alors la variation du paramètre relatif à ce test est prometteuse. Si les solutions de deux tests sont identiques, on compare leur temps d'exécution.

Pour résoudre le problème de programmation mathématique en nombre entiers, nous avons utilisé le solveur CPLEX 12.10.0.0 IBM ILOG CPLEX Optimization Studio : 12.10. Le code a été développé en OPL de CPLEX. Tous les résultats des tests présentés dans cette section ont été réalisés sur un ordinateur tournant sur un processeur Intel (R) Core (TM) i9 - 9900K, une fréquence d'horloge de 3.60GHz et 16GO de RAM, Windows 64bits comme système d'exploitation.

#### **4.3.2.1 Résultats**

Cette section présente les résultats des tests numériques. Les tableaux comprennent la période, une valeur qui précise si la solution est optimale, le temps d'exécution en secondes, le nombre de contraintes insatisfaites, la valeur de la fonction objectif, le nombre de variables, le nombre de contraintes et le nombre d'itérations lors de la résolution.

### **T0 Méthode exacte**

Le problème global a été résolu à l'aide de CPLEX ce qui correspond à la méthode exacte. Les résultats se trouvent dans le tableau 4.2.

<b>T0 Méthode exacte</b>	
Solution optimale	Oui
Temps d'exécution (secondes)	16140
Nombre de contraintes insatisfaites	1
Valeur de la fonction objectif	1
Nombre de variables	21280
Nombre de contraintes	512757
Nombre d'itérations	6643777

Tableau 4.2 : Résultats de T0

Le problème global, pour toute l'année, a donc pu être résolu de façon optimale avec une seule contrainte insatisfaite (un des employés doit travailler 49 jours pendant la période d'hiver alors que la contrainte initiale imposait au plus 48 jours de travail.). Cependant, le temps d'exécution est trop long, CPLEX a obtenu une solution après 4 heures et 29 minutes. Cela s'explique par le fait que la taille du problème est très grande.

### **T1 Méthode de décomposition : pénalités égales ( $\alpha = \beta = 1$ )**

Le tableau 4.3 présente les résultats du test T1 (les pénalités sur le nombre de jours de travail requis par période et le nombre des employés requis par jour sont égales) avec décomposition du problème global en sous-problèmes.

<b>T1 Méthode de décomposition : pénalités égales (<math>\alpha = \beta = 1</math>)</b>			
	<u>Période</u>		
	<u>Hiver</u>	<u>Été</u>	<u>Automne</u>
Solution optimale	Oui	Oui	Oui
Temps d'exécution (secondes)	27	1270	111
Nombre de contraintes insatisfaites	1	0	0
Valeur de la fonction objectif	1	0	0
Nombre de variables	5128	10028	8433
Nombre de contraintes	116766	238766	192645
Nombre d'itérations	133434	832520	117479

Tableau 4.3 : Résultats de T1

Une solution optimale a été obtenue pour chaque période en un temps raisonnable. L'expérimentation la plus longue est pour la période d'été. Cela s'explique par le fait que la taille du problème pour la période d'été est plus grande. Le temps d'exécution total est 1408 secondes soit environ 23 minutes, ce qui est une nette amélioration par rapport au temps obtenu avec le test T0. Une seule contrainte n'est pas satisfaite durant la période d'hiver. En effet, un des employés doit travailler 49 jours pendant la période d'hiver alors que la contrainte initiale imposait au plus 48 jours de travail.

#### **T2 Méthode de décomposition : division de la période d'été en deux sous périodes avec des pénalités égales ( $\alpha = \beta = 1$ )**

Le temps d'exécution du problème de la période d'été pour T1 est plus élevé que ceux des périodes d'hiver et d'automne. Pour réduire ce temps, on a divisé la période d'été en deux sous périodes.



<b>T2 Méthode de décomposition : division de la période d'été en deux sous périodes avec des pénalités égales (<math>\alpha = \beta = 1</math>)</b>				
	<b>Période</b>			
	Hiver	Été		Automne
		Jours 90 à 181	Jours 182 à 243	
Solution optimale	Oui	Oui	Oui	Oui
Temps d'exécution (secondes)	27	56	6	283
Nombre de contraintes non satisfaites	1	0	0	0
Valeur de la fonction objectif	1	0	0	0
Nombre de variables	5128	6494	4784	8433
Nombre de contraintes	116766	149748	106728	192645
Nombre d'itérations	133434	98174	4769	392021

Tableau 4.4 : Résultats de T2

D'après le tableau 4.4, on peut faire les mêmes constatations que celles tirées du test T1. De plus, même si le nombre de contraintes insatisfaites n'a pas changé, on arrive tout de même à obtenir la solution beaucoup plus rapidement et ce, avec un nombre d'itérations plus petit. Le temps total de résolution est 372 secondes soit 6,2 minutes. L'amélioration du temps de résolution est notable pour le sous-problème de la période d'été avec 62 secondes au total. Cependant, celui du problème de l'automne a augmenté. Cela s'explique par le fait que le problème d'une période donnée dépend de la solution de la fin de la période précédente.

**T3 Méthode de décomposition : Augmentation de la valeur de pénalité  $\alpha$  de 1 à 10 et maintien de la valeur de pénalité  $\beta$  à 1 ( $\alpha=10, \beta=1$ )**

Rappelons que la pénalité  $\alpha$  s'applique aux contraintes qui limitent le nombre d'employés requis par jour à un nombre maximum lorsqu'elles ne sont pas satisfaites. La pénalité  $\beta$  s'applique aux contraintes qui imposent à chaque employé un nombre maximal de jours de travail pour chaque période lorsqu'elles ne sont pas satisfaites. Ces deux contraintes sont, en effet, considérées comme des contraintes souples. Pour les tests T0 à T2, les valeurs des pénalités  $\alpha$  et  $\beta$  étaient égales. Les tests T3 et T4 permettent de vérifier l'impact sur la solution de faire varier les valeurs de ces pénalités. Le tableau 4.5 présente les résultats du test T3.

Il est à noter que des tests pourraient être faits avec d'autres valeurs de  $\alpha$  et  $\beta$ . Pour les tests T3 et T4, les tests visaient à vérifier l'impact de ces paramètres quand l'écart entre leurs valeurs est élevé.

<b>T3 Méthode de décomposition : Augmentation de la valeur de pénalité <math>\alpha</math> de 1 à 10 et maintien de la valeur de pénalité <math>\beta</math> à 1 (<math>\alpha = 10, \beta = 1</math>)</b>			
	<u>Période</u>		
	<u>Hiver</u>	<u>Été</u>	<u>Automne</u>
Solution optimale	Oui	Oui	Oui
Temps d'exécution (secondes)	75	2093	123
Nombre de contraintes non satisfaites	1	0	0
Valeur de la fonction objectif	1	0	0
Nombre de variables	5128	10028	8433
Nombre de contraintes	116766	238766	192645
Nombre d'itérations	294329	1127332	119376

Tableau 4.5 : Résultats de T3

Le tableau 4.5 indique que le problème admet une solution optimale en un temps raisonnable avec une seule contrainte insatisfaite (la solution n'a pas changé par rapport à celle du test T1). La contrainte insatisfaite est en lien avec un employé qui travaille un jour de plus que le maximum de jours de travail permis durant la période d'hiver. Le temps total de résolution est 2291 secondes soit 38 minutes, ce qui représente une

augmentation de 15 minutes par rapport à celui de T1. En augmentant la valeur de la pénalité  $\alpha$  sur le nombre maximum d'employés requis par jour et en maintenant celle de  $\beta$ , sur le nombre maximum de jours de travail par employé durant chaque période, le problème devient plus contraignant. La recherche d'une solution afin de satisfaire les contraintes sur lesquelles les pénalités s'appliquent prend alors plus de temps. Cela peut expliquer le temps d'exécution plus élevé que celui du test T1.

**T4 Méthode de décomposition : Maintien de la valeur de pénalité  $\alpha$  à 1 et augmentation de la valeur de pénalité  $\beta$  de 1 à 10 ( $\alpha=1$ ,  $\beta=10$ )**

Pour le test T4, c'est la valeur de la pénalité  $\beta$ , sur la contrainte (3.4) qui exige un nombre maximal de jours de travail par employé pour chaque période, qui a été augmenté. Une seule de ces contraintes n'a pas été satisfaite. Comme la pénalité est de 10, la valeur de la fonction objectif est donc de 10 au lieu de 1 comme pour le test T3. Il semble donc, selon les résultats des tests, que l'augmentation de la valeur des pénalités  $\alpha$  ou  $\beta$  n'a pas d'influence sur la solution. Le temps de résolution total est 1557 secondes soit près de 26 minutes. Ce temps est moindre que celui de T3 (38 minutes) mais demeure un peu plus élevé que celui de T1 (23 minutes). L'augmentation des valeurs des pénalités a donc un impact négatif puisqu'elle augmente le temps d'exécution sans améliorer la solution.

<b>T4 Méthode de décomposition : Maintien de la valeur de pénalité <math>\alpha</math> à 1 et augmentation de la valeur de pénalité <math>\beta</math> de 1 à 10 (<math>\alpha=1, \beta=10</math>)</b>			
	<u>Période</u>		
	<u>Hiver</u>	<u>Été</u>	<u>Automne</u>
Solution optimale	Oui	Oui	Oui
Temps d'exécution (secondes)	101	1232	224
Nombre de contraintes non satisfaites	1	0	0
Valeur de la fonction objectif	10	0	0
Nombre de variables	5128	10028	8433
Nombre de contraintes	116766	238766	192645
Nombre d'itérations	794391	745021	332907

Tableau 4.6 : Résultats de T4

#### 4.3.5 Synthèse des résultats

Le tableau 4.7 résume les résultats des différents tests en présentant la solution obtenue pour toute l'année.

	<b>Tests</b>				
	T0	T1	T2	T3	T4
Solution optimale	Oui	Oui	Oui	Oui	Oui
Nombre de contraintes non satisfaites	1	1	1	1	1
Temps de résolution (secondes)	16140	1408	372	2291	1557
Nombre d'itérations	6643777	1083433	628398	1541037	1872319

Tableau 4.7 : Synthèse des résultats

Il est important de signaler que le nombre d'itérations et le temps de résolution sont des indicateurs importants. En effet, si la solution ne change pas d'un test à l'autre, il importe de constater si on atteint la même solution mais en un nombre d'itérations moindre. Cela se révèle un point assez important quant à la vitesse de convergence de notre algorithme.

On remarque dans le tableau 4.7 que, pour tous les tests de la méthode de décomposition, il y a une amélioration du temps d'exécution si on le compare à celui de la résolution du problème global. Cependant, le nombre de contraintes non satisfaites est le même pour tous les tests, peu importe la valeur des pénalités.

Le temps d'exécution et le nombre d'itérations pour T2 sont les plus petits par rapport à la résolution du problème global et les autres tests de la méthode de décomposition. Cela s'explique par le fait que plus on réduit la taille de l'horizon de planification, plus on obtiendra une solution optimale rapidement. On peut conclure que la méthode exacte et la méthode de décomposition réussissent à trouver une solution optimale. Mais, la méthode de décomposition est plus efficace en ce qui concerne le temps d'exécution comparé à celui de la méthode exacte.

## 5- CONCLUSION

### 5-1 Synthèse des travaux

Le problème de génération d'horaires de travail a été étudié par plusieurs chercheurs. Dans la revue de littérature, nous avons vu plusieurs techniques pour résoudre ce problème. Cependant, ces techniques permettaient de résoudre les problèmes de confection d'horaires de travail sur des horizons de planification de quelques jours à quelques semaines. Aucune méthode n'avait été présentée pour résoudre le problème de grande taille sur une période d'un an. Dans ce mémoire, nous avons vu qu'il était possible de résoudre un problème de grande taille sur un horizon de planification d'un an.

Le modèle mathématique présenté dans ce mémoire est un modèle de programmation linéaire avec variables binaires qui indiquent si un employé travaille ou est en congé un jour donné.

Dans la phase d'expérimentation, CPLEX a pu résoudre le problème global de grande taille mais avec un long temps d'exécution (4 heures et 21 minutes). La méthode de décomposition du problème global en sous problèmes avec des périodes de longueur variée a permis de résoudre chacun très rapidement. Les temps de résolution varient entre 372 secondes et 2291 secondes.

Finalement, nous avons observé que la complexité du problème n'est pas liée au nombre de contraintes ni au nombre de variables, mais plutôt au nombre de jours (horizon de planification). De plus, le fait de varier les valeurs des pénalités n'a pas réellement eu d'impact sur la qualité de la solution du problème.

## **5-2 Limites des solutions proposées**

Une première limitation associée au problème global est sa complexité qui grossit avec le nombre de jours et le nombre d'employés. Dans les tests que nous avons effectués, le nombre d'employés n'a jamais dépassé 55. Dans des grandes entreprises, ce nombre peut atteindre plus de 1000 employés. La méthode de décomposition pourrait avoir de la difficulté à traiter un problème avec un si grand nombre d'employés. Le nombre d'employés peut donc devenir une limitation à ce problème.

La deuxième limitation est au niveau des intervalles des paramètres du modèle mathématique du problème global tels que pour les séquences des jours de travail et de congé. Plus les intervalles sont petits, plus il sera difficile pour le solveur de trouver une solution optimale et peut-être même une solution réalisable.

## **5-3 Améliorations futures**

Pour résoudre plus facilement notre problème d'horaire du personnel qui est de grande taille sur un horizon de planification d'un an, on a assoupli les contraintes du modèle global par rapport à celles de départ. Ça pourrait être intéressant de considérer des contraintes plus strictes, il serait alors plus difficile d'obtenir une solution réalisable en un temps raisonnable. Dans ce cas, on pourrait utiliser une méthode heuristique ou la méthode de programmation par contraintes qui permettrait peut-être de résoudre le problème plus facilement sans diviser l'horizon de planification du problème en sous périodes. Cela permettrait peut-être d'accélérer le temps de résolution tout en gardant la qualité de la solution. 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.

## BIBLIOGRAPHIE

K. R. Baker. Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, vol. 27, no. 1, pp.155–167, 1976.

V. Boyer, B. Gendron et L.M. Rousseau. A branch-and-price algorithm for the multi-activity multi-task shift scheduling problem. *Journal of Scheduling* volume 17, pp, 185-197, 2014.

R. Bürgy, H. M. Lacaze et G. Desaulniers. Employee scheduling with short demand perturbations and extensible shifts. *Omega*, 89, pp.177–192, 2019.

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, 1954.

P. R. Day et D. M. Ryan. Flight attendant rostering for short-haul airline operations. *Operations Research*, vol. 45, no. 5, pp. 649–661, 1997.

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

M. Erhard, J. Schoenfelder, A. Fugener et J. O. Brunner. State of the art in physician scheduling. *European Journal of Operational Research*, 265, pp. 1–18, 2018.

Y. Ferrand, M. Magazine, U. S. Rao, et T. F. Glass. Building Cyclic Schedules for Emergency Department Physicians. *Interfaces*, vol. 41, pp. 521-533, 2011.

C. Froger. Mise à jour des horaires de personnel travaillant sur des quarts. *Mémoire de Maîtrise, École Polytechnique de Montréal*, 2015.

J. Guo et J. F. Bard. A column generation-based algorithm for midterm nurse scheduling with specialized constraints, preference considerations, and overtime. *Computers and Operations Research*, 138, pp.1-26, à paraître 2022.



J. Heil, Kirsten Hoffmann et U. Buscher. Railway crew scheduling: Models, methods and applications. *European Journal of Operational Research*, 283, pp. 405–425, 2020.

N. A. Hernández-Leandro, V. Boyer, M. Angélica Salazar-Aguilar et L. Rousseau. A matheuristic based on Lagrangian relaxation for the multi-activity shift scheduling problem. *European Journal of Operational Research*, 272, pp. 859-867, 2019.

O. Jens, Brunner et Gunther M, Edenharter, Long term staff scheduling of physicians with different experience levels in hospitals using column generation, *Health Care Management Science*, 14, pp.189-202, 2011.

L. Kang, S. Chen and Q. Meng. Bus and driver scheduling with mealtime windows for a single public bus route. *Transportation Research Part C*, 101, pp.145–160, 2019.

L. Kletzander et N. Musliu. Solving the general employee scheduling problem. *Computers and Operations Research*, 113, pp.1-13, 2020.

A. Leggate, S. Sucu, K. Akartunal et R. van der Meer. Modelling crew scheduling in offshore supply vessels. *Journal of the Operational Research Society*, 69 :6, pp. 959-970, 2018.

A. Legrain · H. Bouarab et N. Lahrichi. The Nurse Scheduling Problem in Real-Life. *J Med Syst*, 39 :160, pp.1-11, 2015.

A. Legrain, J. Omer et S. Rosat. A rotation-based branch-and-price approach for the nurse scheduling problem. *Mathematical Programming Computation*, 12, pp.417- 450, 2020.

D. Lin et C. Hsu. A column generation algorithm for the bus driver scheduling problem. *Journal of Advanced Transformation*, 50, pp.1598–1615, 2016.

A. Lorenzo-Espejo, J. Munuzuri, L. Onieva et P. Cortés. Scheduling consecutive days off: A case study of maritime pilots. *Computers & Industrial Engineering*, 155, pp.1-18, 2021.

- R. Mansini et R. Zanotti. Optimizing the physician scheduling problem in a large hospital ward. *Journal of Scheduling*, 23, pp.337–361, 2020.
- I. Mathlouthi, M. Gendreau et J.Y. Potvin. A metaheuristic based on tabu search for solving a technician routing and scheduling problem. *Computers and Operations Research*, 125, pp.1-8, 2021.
- H. Michon-Lacaze. Élaboration de quarts de travail robustes aux perturbations de courtes durées. Mémoire de maîtrise, École Polytechnique de Montréal, 2016.
- A. H. Nobile, S. M. E. Sharifnia et L. E. Cardenas-Barron. Mixed integer linear programming problem for personnel multi-day shift scheduling: A case study in an Iran hospital. *Alexandria Engineering Journal*, 61, pp.419-426, 2021.
- E. H. Özder E. Özcan et T. Eren. A Systematic Literature Review for Personnel Scheduling Problems. *International Journal of Information Technology & Decision Making*, Vol. 19, No. 06, pp. 1695-1735, 2020.
- A. Parisio, C. N. Jones. A two-stage stochastic programming approach to employee scheduling in retail outlets with uncertain demand. *Omega*, 53, pp.97–103, 2015.
- S.S.G. Perumal, J. Larsen, R. M. Lusby, M. Riis et K. S. Sørensen. A matheuristic for the driver scheduling problem with staff cars. *European Journal of Operational Research*, 275, pp.280-294, 2019.
- J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester et L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, vol. 226, no. 3, pp. 367–385, 2013.
- J. Volland, A. Fügener et Jens O. Brunner. A column generation approach for the integrated shift and task scheduling problem of logistics assistants in hospitals. *European Journal of Operational Research*, 260, pp. 316-334, 2017.

- X.Wen, X. Sun, Y. Sun et X. Yue. Airline crew scheduling: Models and algorithms. *Transportation Research Part E*, 149, pp. 1-18, 2021.
- L. Wu, S. Jia et S. Wang. Pilotage planning in seaports. *European Journal of Operational Research*, 287, pp. 90-105, 2020.
- D. Yung, V. Bucarey, M. Olivares et M. Christensen. Labor Planning and Shift Scheduling in Retail Stores Using Customer Traffic Data. Université Libre de Bruxelles, Inria Lille-Nord Europe, Villeneuve d'Ascq, France, Université de Chile, pp. 1-34, 2020.
- E Zamorano et R. Stolletz. Branch-and-price approaches for the Multiperiod Technician Routing and Scheduling Problem. *European Journal of Operational Research*, 257, pp. 55–68, 2017.
- S. Zanda, P. Zuddasb et C. Seatzu. Long term nurse scheduling via a decision support system based on linear integer programming: A case study at the University Hospital in Cagliari. *Computers & Industrial Engineering*, 126, pp. 337-347, 2018.
- V.Zeighami et F. Soumis. Combining Benders' Decomposition and Column Generation for Integrated Crew Pairing and Personalized Crew Assignment Problems. *Transportation Science*, Vol. 53, No. 5, pp. 1479–1499, 2019.
- Z. Zhao et X. Li. Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming. *Operations Research for Health Care* 3, pp. 160-167, 2014.

## Annexe A

Obtention d'un horaire initial

```
* OPL 12.10.0.0 Model
* Author: benhaddad
* Creation Date: 6 Nov. 2020 at 19:47:52
*****/
// Définition des ensembles
range I= 1..55;    //Ensemble des employés
range Ia= 1..43;   //Ensemble des employés de type A
range Il= 44..55;  //Ensemble des employés l
range J=1..20;     //Ensemble des jours de l'année
range R=1..1;      // 1<= r <= |J|-cwmax
range R1=1..5;     // 1<= r <= |J|-comax

// Déclaration des variables de décisions
dvar boolean x[I][J];
dvar boolean y[I];

// Déclaration des variables d'écart par rapport à la
satisfaction des contraintes souples.

dvar int+ z[J];
dvar int+ z1[J];

int M=200; //une constante suffisamment grande

// Fonction objectif
minimize
sum (j in J) z[j]+sum (j in J) z1[j];

subject to {

// Nombre minimum d'employés requis par jour.

forall (j in J)
sum (i in I) x[i][j]>=25;

// Nombre maximum d'employés requis par jour.

forall (j in J)
sum (i in I) x[i][j]- z[j]<=36;
```

```

// Nombre minimum d'employés requis par jour de catégorie L.
(resp nombre max)

forall (j in J)
sum (i in I1) x[i][j]>=3;

forall (j in J)
sum (i in I1) x[i][j]-z1[j]<=7;

// Chaque pilote doit travailler au moins 15 JOURS consécutifs

forall (i in I)
forall (t in 2..15)
forall (j in 1..20-t)
x[i][j]-sum (k in 1..t-1) x[i][j+k]+x[i][j+t]>=-t+2;
forall (i in I)
sum (j in 1..15) x[i][j] >= 15-M*y[i];

// Chaque pilote ne doit pas travailler plus de 19 jours
consécutifs

forall (i in I)
forall (r in R)
sum (j in r..r+19) x[i][j] <=19;

// Chaque pilote devrait être en congé au moins 11 jours
consécutifs

forall (i in I)
forall (t in 2..11)
forall (j in 1..20-t)
-x[i][j] + sum (k in 1..t-1) x[i][j+k]-x[i][j+t] >=-1;
forall (i in I)
sum (j in 1..11) x[i][j] <= 0 + M*(1-y[i]);

// Chaque pilote devrait être en congé pendant un maximum de
15 jours consécutifs

forall (i in I)
forall (r1 in R1)
sum (j in r1..r1+15)(1- x[i][j]) <=15;

}

```

## Annexe B

Résolution du problème pour la période de l'hiver

```
/******  
* OPL 12.10.0.0 Model  
* Author: benhaddad  
* Creation Date: 6 Nov. 2020 at 19:47:52  
*****/  
range I= 1..55;    //ensemble des employés  
range Ia= 1..43;   //ensemble des employés de type A  
range Il= 44..55;  //ensemble des employés de type L  
range J=1..89;     //Ensemble des jours de l'année  
range R=1..70;     // 1<= r <= |J|-cwmax  
range R1=1..74;    // 1<= r <= |J|-comax  
  
int a[1..55][1..20] =...; // solution de départ  
  
// Déclaration des variables de décisions  
  
dvar boolean x[I][J];  
  
// Déclaration des variables d'écart par rapport à la  
satisfaction des contraintes souples.  
  
dvar int+ w[I];  
dvar int+ w1[I];  
dvar int+ z[J];  
dvar int+ z1[J];  
  
// Fonction objectif  
minimize  
sum (j in J) z[j]+sum (j in J) z1[j]+sum (i in I) w[i]+  
sum (i in I) w1[i];  
  
subject to {  
  
//solution de depart  
forall (i in I)  
forall (j in 1..20)  
x[i][j]==a[i][j];  
  
//nombre minimum d'employés requis par jour  
forall (j in 21..89)  
sum (i in I) x[i][j]>=25;  
  
//nombre maximum d'employés requis par jour.
```

```

forall (j in 21..89)
sum (i in I) x[i][j]- z[j]<=36;

//nombre minimum d'employés requis par jour de catégorie L.
(resp nombre max).

forall (j in J)
sum (i in I1) x[i][j]>=3;

forall (j in J)
sum (i in I1) x[i][j]-z1[j]<=7;

//nombre maximum de jour de travail durant l'hiver
forall (i in I)
sum (j in J) x[i][j]+w1[i]>=44;

forall (i in I)
sum (j in J) x[i][j]-w[i]<=48;

//Chaque pilote doit travailler au moins 15 JOURS consécutifs

forall (i in I)
forall (t in 2..15)
forall (j in 1..89-t)
x[i][j]-sum (k in 1..t-1)x[i][j+k]+x[i][j+t]-z[i][j]>=-t+2;

//Chaque pilote ne doit pas travailler plus de 19 jours
consécutifs

forall (i in I)
forall (r in R)
sum (j in r..r+19) x[i][j] <=19;

// Chaque pilote devrait être en congé au moins 11 jours
consécutifs

forall (i in I)
forall (t in 2..11)
forall (j in 1..89-t)
-x[i][j]+sum(k in 1..t-1)x[i][j+k]-x[i][j+t] >=-1;

```

```
// Chaque pilote devrait être en congé pendant un maximum de  
15 jours consécutifs
```

```
forall (i in I)  
forall (r1 in R1)  
sum(j in r1..r1+15)(1- x[i][j]) <=15;  
}
```



## Annexe C

Résolution du problème pour la période de l'été

```

/*****
 * OPL 12.10.0.0 Model
 * Author: benhaddad
 * Creation Date: 11 Nov. 2020 at 23:36:14
 *****/
range I= 1..55;      //ensemble des employés
range Ia= 1..43;     //ensemble des employés A
range Il= 44..55;    //ensemble des employés l
range J=70..243;     //Ensemble des jours de l'année
range R=70..243-19;  // 1<= r <= |J|-cymax
range Rl=70..243-15; // 1<= r <= |J|-comax

int b[1..55][70..89] =...; // solution fin de période d'hiver

// Déclaration des variables de décisions
dvar boolean x[I][J];

// Déclaration des variables d'écart par rapport à la
satisfaction des contraintes souples.

dvar int+ w[I];
dvar int+ w1[I];
dvar int+ z[J];
dvar int+ z1[J];

// Fonction objectif

sum (j in J) z[j]+sum (j in J) z1[j]+sum (i in I) w[i];

subject to {

//solution de départ (période de transition)

forall (i in I)
forall (j in 70..89)
x[i][j]==b[i][j];

//nombre minimum d'employés requis par jour.

forall (j in 90..243)
sum (i in I) x[i][j]>=25;
```

```

forall (j in 182..243)
sum (i in I) x[i][j]>=30;

//nombre maximum d'employés requis par jour.

forall (j in 90..181)
sum (i in I) x[i][j]-z[j]<=36;

//nombre maximum d'employés requis par jour. (1 juillet au 31
août).

forall (j in 182..243)
sum (i in I) x[i][j]-z[j]<=33;

//nombre minimum d'employés requis par jour de catégorie L.
(resp nombre max)

forall (j in 90..243)
sum (i in I1) x[i][j]>=3;

forall (j in 90..243)
sum (i in I1) x[i][j]- z1[j]<=7;

//nombre de jour de travail requis par chaque employé pendant
le mois d'avril

forall (i in I)
sum (j in 90..120) x[i][j]>=10;
forall (i in I)
sum (j in 90..120) x[i][j]- w[i]<=18;

//nombre minimum de jour de travail requis pour chaque employé
(de 1 juillet au 31 août)

forall (i in I)
sum (j in 182..243) x[i][j]>=35;

//nombre maximum de jour de travail requis pour chaque employé
(de 1 juillet au 31 août)

forall (i in I)
sum (j in 182..243) x[i][j]-w[i]<=38;

```

```

//Chaque pilote doit travailler au moins 15 JOURS consécutifs

forall (i in I)
forall (t in 2..15)
forall (j in 70..243-t)

    x[i][j]-sum (k in 1..t-1) x[i][j+k]+x[i][j+t]-v[i][j]>=-t+2;

//Chaque pilote ne doit pas travailler plus de 19 jours
consécutifs

forall (i in I)
forall (r in 70..243-19)
sum (j in r..r+19) x[i][j]-p[i][r]<=19;

//Chaque pilote devrait être en congé pendant au moins 11
jours consécutifs

forall (i in I)
forall (t in 2..11)
forall (j in 70..243-t)

    -x[i][j]+sum (k in 1..t-1) x[i][j+k]-x[i][j+t]-z[i][j]>=-1;

// Chaque pilote devrait être en congé pendant un maximum de
15 jours consécutifs

forall (i in I)

forall (r1 in 70..243-15)

sum (j in r1..r1+15)(1- x[i][j])-f[i][r1] <=15;

}

```

## Annexe D

Résolution du problème pour la période d'automne

```

/*****
 * OPL 12.10.0.0 Model
 * Author: benhaddad
 * Creation Date: 4 avr. 2021 at 18:16:53
 *****/
range I= 1..66;           //ensemble des employés
range Ia= 1..55;          //ensemble des employés A
range Il= 56..66;         //ensemble des employés l
range J=224..365;         //ensemble des jours de l'année
range R=224..365-19;      // 1<= r <= |J|-cwmax
range Rl=224..365-15;     // 1<= r <= |J|-comax

int q[1..66][224..243] =...; // solution fin de période d'été.

// Déclaration des variables de décisions
dvar boolean x[I][J];

// Déclaration des variables d'écart par rapport à la
satisfaction des contraintes souples.

dvar int+ w[I];
dvar int+ z[J];
dvar int+ z1[J];
dvar int+ z2[J];
dvar int+ z3[J];

minimize
sum (j in J) z[j]+sum (j in J) z1[j]+sum (j in J) z2[j]
+sum (j in J) z3[j] + sum (i in I) w[i];

subject to {

forall (i in I)
forall (j in 224..243)
x[i][j]==q[i][j];

//nombre minimum d'employés requis par jour.

forall (j in 244..342)
sum (i in Ia) x[i][j]>=25;
```

```

forall (j in 356..365)
sum (i in Ia) x[i][j]>=25;

//nombre minimum d'employés requis par jour durant la période
de Noël.

forall (j in 343..355)
sum (i in I) x[i][j]+z[j]>=36;

//nombre maximum d'employés requis par jour

forall (j in 244..342)
sum (i in I) x[i][j]- z1[j]<=36;

forall (j in 356..365)
sum (i in I) x[i][j]-z2[j]<=36;

//nombre minimum d'employés requis par jour de catégorie L
(resp nombre max).

forall (j in 244..365)
sum (i in Il) x[i][j]>=3;

forall (j in 244..365)
sum (i in Il) x[i][j]- z3[j]<=7;

//nombre de jour de travail requis par chaque employé

forall (i in I)
sum (j in 244..365) x[i][j] >= 60;

forall (i in I)
sum (j in 244..365) x[i][j]-w[i]<=73;

//Chaque pilote doit travailler au moins 15 JOURS consécutifs.

forall (i in I)
forall (t in 2..15)
forall (j in 224..365-t)
x[i][j]-sum (k in 1..t-1) x[i][j+k]+x[i][j+t]>=-t+2;

//Chaque pilote ne doit pas travailler plus de 19 jours
consécutifs.

forall (i in I)

```

```

forall (r in 224.. 365-19)
sum (j in r..r+19) x[i][j]<=19;

//Chaque pilote devrait être en congé pendant au moins 11
jours consécutifs

forall (i in I)
forall (t in 2..11)
forall (j in 224..365-t)
-x[i][j] + sum (k in 1..t-1) x[i][j+k]-x[i][j+t]>=-1;

// Chaque pilote devrait être en congé pendant un maximum de
15 jours consécutifs

forall (i in I)
forall (r1 in 224..365-15)
sum (j in r1..r1+15) (1- x[i][j]) <=15;

}

```