POLYPUBLIE

Réduction de réseau pour le problème d'horaires de personnel aérien

Laurent Deirmendjian

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

Date	1999
Type	Mémoire ou thèse / Dissertation or Thesis Mémoire de maîtrise/Master's thesis
URL de PolyPublie **PolyPublie URL**	https://publications.polymtl.ca/8861/
Conditions d'utilisation **Terms of use**	© Tous droits réservés / All rights reserved
Référence **Citation**	Deirmendjian, L. (1999). Réduction de réseau pour le problème d'horaires de personnel aérien [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8861/

Détails de la thèse
Thesis details

Directeurs de recherche **Advisors**	François Soumis, & Beyime Tachefine
Programme **Program**	Non spécifié

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal
https://publications.polymtl.ca

UNIVERSITÉ DE MONTRÉAL

RÉDUCTION DE RÉSEAU POUR LE PROBLÈME D'HORAIRES DE PERSONNEL AÉRIEN

LAURENT DEIRMENDJIAN
DÉPARTEMENT DE MATHÉMATIQUES
ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
MAI 1999

© Laurent Deirmendjian, 1999.

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

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

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

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

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

0-612-46646-9

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

RÉDUCTION DE RÉSEAU POUR LE PROBLÈME D'HORAIRES DE PERSONNEL AÉRIEN

présenté par : DEIRMENDJIAN, Laurent

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

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

M. SMITH Benjamin T., Ph.D., président

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

M. TACHEFINE Beyime, Ph.D., membre et codirecteur de recherche

Mme. SANSO Brunilde, Ph.D., membre

A ma douce Marie

Remerciements

Je remercie mes deux directeurs de recherche, MM. François Soumis et Beyime Tachefine pour leur soutien professionnel et leur disponibilité.

Je tiens à remercier les membres et professionnels du GERAD qui m'ont aidé tout au long de mon travail, Norbert Lingaya, Anass Naghi, Daniel Villeneuve et particulièrement Sylvie Gélinas et Gérald Marquis de la compagnie Ad Opt Inc.

Enfin, je veux exprimer ma gratitude au personnel du GERAD, notamment, Francine Benoit, Carole Dufour, Nicole Paradis et Christine Picard.

Résumé

Ce mémoire décrit la réduction de réseau pour un problème d'horaires mensuels personnalisés pour les membres d'équipage (pilotes et agents de bord) en transport aérien. Le problème consiste à attribuer aux membres d'équipage des rotations, des repos, des vacances, des entraînements, etc., tout en considérant un ensemble de requêtes pondérées reflétant les préférences de chacun. L'attribution doit tenir compte de l'ancienneté de chaque employé. La construction de l'horaire à pointage maximal pour un employé ne doit jamais s'effectuer au détriment de l'horaire d'un employé plus senior.

Ce problème est formulé comme un problème non linéaire de réseaux multi-commodités, en nombre entiers et avec variables de ressources additionnelles. La méthode de résolution employée peut se résumer comme suit. Pour chaque employé du plus ancien au plus récent selon l'ordre d'ancienneté, un problème résiduel est résolu : étant donné un employé et un ensemble de rotations non affectées, la solution d'un problème linéaire en nombres entiers, appelé encore problème maître (PM), détermine l'horaire à pointage maximal de l'employé tout en s'assurant que les employés résiduels peuvent couvrir toutes les rotations non affectées. La résolution du problème résiduel s'effectue par une méthode d'évaluation et de séparation dans un arbre de branchement où à chaque nœud de branchement, un problème linéaire est résolu par la technique de génération de colonnes. Chaque colonne représente un *horaire*. Le sous-problème permettant de générer ces colonnes consiste à résoudre un problème de plus court chemin (PCC) avec contraintes sur un graphe associé à l'aide d'algorithmes spécialisés de programmation dynamique.

L'objectif de ce travail est d'améliorer les performances de l'algorithme du problème de plus court chemin avec contraintes (PCCRC) en définissant des techniques de réduction du réseau. La contribution de ce mémoire est d'avoir implanté et testé de nouvelles procédures permettant d'éliminer les tâches apparemment inintéressantes pour l'employé courant afin de réduire son réseau et ainsi diminuer le temps de calcul de la méthode. Ainsi, nous verrons au chapitre 4 les idées générales de réduction basées sur une classification des tâches à réaliser par les employés.

Abstract

This dissertation describes the reduction of networks for the Preferential Bidding Problem solved in the airline industry to construct personalized monthly schedules for pilots and officers. The problem consists in assigning to crew members pairings, days off, annual leaves, training periods, etc., while considering a set of weighted bids that reflect individual preferences. The assignment must be done under strict seniority restrictions : the construction of a maximum-score schedule for a particular crew member must never be done at the expense of a more senior employee.

The problem is formulated as an integer, nonlinear multi-commodity network flow problem with additional resource variables. The solution process is summarized as follows. For each employee, from the most senior to the most junior, a so-called residual problem is solved. Given an employee and a set of unassigned pairings, the solution to an integer linear program determines the employee's maximum-score schedule while taking into account all the remaining employees. The residual problem is solved by a column generation embedded in a branch&bound procedure. Each column represents a schedule. The subproblem generates columns using resource constrained shortest path problems which are solved by means of a specialized dynamic programming algorithm.

The goal of this research was to improve the performances of the algorithm of the Shortest Path Problem with Resource Constraints (PCCRC) by reducing the network. The contribution of this report was to implement and test procedures allowing to eliminate tasks apparently not interesting for an employee in order to

reduce his network and lower the time of calculation spenting in the method. We will see in the chapter 4, the general reduction ideas based on a classification of the tasks to be performed by the employees.

Table des matières

DÉDICACE iv

REMERCIEMENTS v

RÉSUMÉ vi

ABSTRACT viii

TABLE DES MATIÈRES ix

LISTE DES TABLEAUX xiv

LISTE DES FIGURES xvii

LISTE DES ABRÉVIATIONS xviii

LISTE DES ANNEXES xx

CHAPITRE 1 : Introduction **1**

CHAPITRE 2 : Le problème de la fabrication d'horaires mensuels **5**

2.1 Le problème de planification 5

2.2 La fabrication des horaires mensuels 8

2.3 La méthode respectant les préférences suivant l'ancienneté 9

CHAPITRE 3 : Revue de littérature 13

3.1 Le modèle classique 15

3.2 Les approches de résolution par génération de colonnes 17

3.3 Les sous-problèmes 23

3.4 La génération de colonnes appliquée à PBS 29

CHAPITRE 4 : Objectifs et méthodologie 35

4.1 Le logiciel PBS 36

4.1.1 Objectifs généraux 36

4.1.2 Nouveaux développements 37

4.1.3 Explications sur la construction du réseau 39

4.2 Techniques de réduction 42

4.2.1 Tâches mères et tâches filles 43

4.2.2 Stratégies de réduction 44

4.2.3 Implémentation 49

4.2.4 Les retombés 51

4.3 Étude probabiliste 54

CHAPITRE 5 : Expérimentation 58

5.1 Problème test 1 58

5.2 Problème test 2 71

5.3 Problème test 3 77

CHAPITRE 6 : Étude théorique : Calcul analytique de la taille du réseau 79

6.1 Première approche 81

6.2 Seconde approche 86

CHAPITRE 7 : Conclusion 93

BIBLIOGRAPHIE 95

ANNEXE 101

Liste des tableaux

5.1 Tableau exprimant des % suite à une réduction de tâches identiques . 59

5.2 Tableau exprimant des % suite à une réduction de tâches identiques . 66

5.3 Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques 67

5.4 Tableau exprimant le % moyen de tâches semblables enlevées 67

5.5 Tableau exprimant le % moyen d'arcs enlevés 68

5.6 Tableau du % moyen de la diminution du temps de calcul après réduction de tâches semblables 69

5.7 Tableau exprimant des % suite à une réduction de tâches identiques . 71

5.8 Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques 73

5.9 Tableau de % de la diminution de prolongations et d'étiquettes 73

5.10 Tableau exprimant des % suite à une réduction de tâches identiques . 74

5.11 Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques 74

5.12 Tableau exprimant le % moyen de tâches semblables enlevées 74

5.13 Tableau exprimant le % moyen d'arcs enlevés 75

5.14 Tableau du % moyen de la diminution du temps de calcul après réduction de tâches semblables 76

5.15 Tableau résumé des résultats 78

5.16 Tableau résumé du % moyen de la diminution du temps de calcul . . 78

Table des figures

2.1 *Étapes de la planification en transport aérien* 7

3.1 *Principaux modules de l'optimiser GENCOL* 22

4.1 *Exemple d'un horaire dans un sous-problème* 40

4.2 *Réseau d'un sous-problème* 41

4.3 *Tâches identiques* 44

4.4 *Tâches semblables* 45

4.5 *Réduction de tâches semblables* 46

4.6 *Exemple de problème* 47

4.7 *Exemple de réduction heuristique* 49

5.1 *Nombres de tâches restantes avant et après réduction des tâches identiques* 62

5.2 *Temps de calcul des PCC avant et après réduction des tâches identiques* 63

5.3 *Exemple sur le principe de dominance* 65

5.4 *Nombres de tâches restantes après réduction de tâches semblables* . . 68

5.5 *Différence de qualité de la solution après un regroupement de 2 heures* 69

5.6 *Nombres de tâches restantes avant et après réduction des tâches identiques* 72

5.7 *Temps de calcul des PCC avant et après réduction des tâches identiques* 72

5.8 *Courbes du nombres de tâches restantes après réduction de tâches semblables* 75

5.9 *Courbes donnant la différence de qualité (coût) de la solution avec un regroupement de 2 heures* 76

5.10 *Courbes donnant la différence de qualité (coût) de la solution avec un regroupement de 12 heures* 77

6.1 Grille 82

6.2 *Distribution du nombre de cases ayant un même nombre de rotations* 83

6.3 *Distribution du % de tâches en fonction de leurs durées* 88

6.4 *Distribution du % de tâches par tranche d'heure et jour de départ* . . 89

6.5 *Distribution du % de tâches en fonction du jour de départ* 90

6.6 *Distribution du % de tâches par tranche d'heure de départ* 91

Liste des abréviations

CX/CY : Arcs de connections entre certaines tâches respectant les consommations de ressources

DO/FX : Arcs de congés et de repos

LD : Arcs reliant les arcs de repos aux rotations (liens bas)

LN : Arcs reliant les tâches commençant à la même heure (liens suivant)

LU : Arcs reliant les arcs de rotations aux arcs de repos (liens haut)

PCC : Plus Court Chemin

PL : Programmation Linéaire

PM : Problème Maître

PNT/PNC : Personnel Navigant Technique et Cabine

PBS : Preferential Bidding System

SC : Nœud Source

SDV : Service De Vol

SK : Nœud puits

SP : Sous Problème

SPPRC : Plus court chemin avec Contraintes de Ressources

SPPTW : Plus court chemin avec fenêtres de temps

TDV : Tronçon De Vol

Liste des annexes

Annexe A 99

Chapitre 1

Introduction

La gestion d'une flotte d'avions et de ses équipages pose des problèmes de **tournées** et d'**horaires** à la fois importants et difficiles à résoudre. Parmi ceux-ci, il y a la planification des itinéraires des avions, l'élaboration des *rotations* du personnel navigant, ainsi que leur affectation mensuelle au Personnel Navigant Technique (PNT) ou Cabine (PNC). Les deux derniers problèmes sont plus connus sous les noms de *Crew Pairing Problem* et *Monthly Crew Assignment Problem.*

Ces problèmes sont importants car l'objectif d'une compagnie consiste à organiser ses opérations de manière à maximiser ses profits, et des économies énormes peuvent être faites grâce à une organisation efficace. À titre d'exemple, les dépenses engendrées par les salaires des membres d'équipage peuvent atteindre annuellement des centaines de millions de dollars et même plus d'un milliard pour une grosse compagnie aérienne. Ainsi afin d'obtenir un bon ratio revenu/coût, la compagnie doit viser une utilisation intensive de ses avions et du personnel. Cet objectif s'inscrit dans le contexte d'une demande difficile à prévoir, en plus de problèmes météorologiques, d'absentéisme du personnel et de défaillances des équipements. Enfin ce sont des problèmes difficiles car, cela amène à résoudre des problèmes de grandes tailles avec de nombreuses variables et des conventions collectives comportant des contraintes très complexes à respecter.

Des méthodes manuelles ont d'abord été utilisées pour résoudre ces problèmes. Ces méthodes manuelles se sont avérées toutefois rapidement limitées au fur et à mesure que la taille des problèmes augmentait. Des méthodes approximatives (ou heuristiques) utilisant la programmation linéaire ont alors vu le jour. Étant donné la complexité des problèmes de la vie courante, des solutions optimales ne peuvent être obtenues qu'avec une approche d'optimisation mathématique. Plusieurs facteurs justifient fortement l'utilisation de méthodes d'optimisation pour résoudre les problèmes de **tournées** et d'**horaires** auxquels font face les transporteurs aériens. D'une part le temps de calcul se trouve largement réduit par rapport aux méthodes manuelles et d'autre part, la maniabilité actuelle des logiciels d'optimisation permet l'analyse efficace de différents scénarios, ce qui est très utile dans la négociation de conventions collectives ou simplement dans l'évaluation de l'impact du changement des règles et des contraintes. Des horaires d'avion et d'équipage de hautes qualités sont cruciaux dans le domaine du transport aérien où une compétition féroce et des marges de profit très minces sont la norme. D'ailleurs, Desrosiers *et al.* (1991) montrent que l'amélioration moyenne de la qualité des solutions à l'aide de logiciels d'optimisations récents permet des économies de l'ordre de 5% ce qui représente des dizaines de millions de dollars annuellement pour les grosses compagnies.

Ce rapport décrit la construction d'horaires mensuels personnalisés en tenant compte des préférences des membres d'équipage (pilotes et agents de bord). Ce problème consiste à affecter aux membres d'équipage, des rotations, des repos, des vacances, des entraînements etc., tout en considérant un ensemble de requêtes pondérées reflétant les préférences de chacun. L'affectation doit tenir compte de l'ancienneté de chaque employé : en effet la construction de l'horaire à pointage maximal pour un employé ne doit jamais s'effectuer au détriment de l'horaire d'un

employé ayant plus d'ancienneté. De ce projet de recherche et de développement est né un système informatisé de construction d'horaires personnalisés qui est en usage chez Air Canada depuis le mois de mai 1995, le système PBS 200 (**Preferential Bidding System**) .

Ce problème est formulé comme un problème non linéaire de réseau multi-commodités, en nombre entiers et avec variables de ressources additionnelles. La méthode de résolution employée peut se résumer comme suit. Pour chaque employé du plus ancien au plus récent selon l'ordre d'ancienneté, un problème résiduel est résolu : étant donné un employé et un ensemble de rotations non affectées, la solution d'un problème linéaire en nombres entiers, appelé encore problème maître (PM), détermine l'horaire à pointage maximal de l'employé tout en s'assurant que les employés résiduels peuvent couvrir toutes les rotations non affectées. La résolution du problème résiduel s'effectue par une méthode d'évaluation et de séparation dans un arbre de branchement où à chaque nœud de branchement, un problème linéaire est résolu par la technique de génération de colonnes. Chaque colonne représente un *horaire*. Le sous-problème permettant de générer ces colonnes consiste à résoudre un problème de plus court chemin (PCC) avec contraintes sur un graphe associé à l'aide d'algorithmes spécialisés de programmation dynamique.

L'objectif de ce travail est d'améliorer les performances de l'algorithme du PCC avec contraintes en définissant des techniques de réduction du réseau. La contribution de ce mémoire est d'avoir implanté et testé de nouvelles procédures permettant d'éliminer les tâches apparemment inintéressantes pour l'employé courant afin de réduire son réseau et ainsi diminuer le temps de calcul de la méthode.

Ainsi, dans une première partie, nous présenterons le problème de planification aérien et plus particulièrement celui de la fabrication des horaires mensuels. Puis nous verrons comment ces problèmes étaient traités dans le passé avant d'expliquer les méthodes actuelles de résolution de ces problèmes. Enfin, nous terminerons ce rapport avec une présentation des expérimentations entreprises pour réduire le réseau avec des jeux de données de petites tailles (100, 300 et 600 employés) ainsi que les résultats obtenus. Nous terminerons enfin ce rapport par l'extrapolation de nos résultats à l'aide d'études probabilistes en estimant la taille théorique des réseaux après réduction pour de plus gros problèmes (5000 employés) avant de conclure.

Chapitre 2

Le problème de la fabrication d'horaires mensuels

Introduction

Dans les pages qui suivent, nous allons nous intéresser au problème de planification qui se pose dans les compagnies aériennes.

A la section 2.1, nous introduirons les différentes étapes de ce processus de planification.
A la section 2.2, nous nous intéresserons plus particulièrement à la fabrication des horaires mensuels et aux différentes méthodes pour leurs constructions.
Finalement à la section 2.3, nous présenterons la méthode de construction d'horaires mensuels respectant les préférences suivant l'ancienneté appelée **Preferential Bidding System**

2.1 Le problème de planification

Le lecteur pourra se reporter à l'annexe pour les descriptions et terminologies utilisées dans le reste de ce volume.

Dans une compagnie, les décisions se prennent en deux temps : les décisions de planification sont prises à l'avance et les décisions d'opération sont prises en temps réel le jour des opérations. Le processus de planification quant à lui se décompose en quatre étapes qui doivent être faites séquentiellement.

1. **Horaire des vols** ou *flight scheduling* : Cette première phase consiste à déterminer tous les segments de vol à offrir pendant une période donnée, compte tenu des prévisions de la demande, de la concurrence et des intervalles de temps de décollage et d'attérissage alloués à la compagnie dans différents aéroports. Ce problème incombe en majeur partie au département de marketing.
2. **Itinéraire des avions** ou *aircraft routing* : Cette seconde phase consiste à assigner des avions spécifiques aux segments de vol et déterminer l'horaire de maintenance. Tous les segments de vol offerts doivent être couverts exactement une fois par un appareil. Dans ce cas, il est évident que cela dépend des horaires des vols, de la flotte de la compagnie, de la capacité et de la vitesse des différents types d'avion, ainsi que des diverses contraintes d'entretien des appareils.
3. **Rotations d'équipage** ou *crew pairing* : Après avoir déterminés les itinéraires et les horaires des avions, les planificateurs doivent établir un ensemble de rotations. Cela consiste à déterminer les ensembles de segments de vol qui seront effectués par un même équipage. L'information nécessaire à la résolution de ce problème requiert certains résultats de l'étape précédente, en plus des règles des conventions collectives et des lois gouvernementales concernant le transport. Encore une fois, tous les segments de vol doivent être couverts une

seul fois par les rotations L'objectif est de construire un ensemble de rotations d'équipage à coût minimum permettant de couvrir tous les vols à effectuer pour un horizon donné.

4. **Horaires mensuels** ou *monthly assignment* : Une fois les rotations construites, cette quatrième phase consiste à affecter les rotations au personnel navigant, en construisant des blocs généralement d'une durée d'un mois. L'objectif de ce problème est de construire des blocs qui maximisent le bien être du personnel tout en maintenant une utilisation efficace de celui-ci.

   L'information requise pour résoudre ce type de problème est l'ensemble des rotations déterminées précédemment ainsi que les conventions de travail.

La figure 2.1 résume le processus séquentiel de planification en transport aérien.

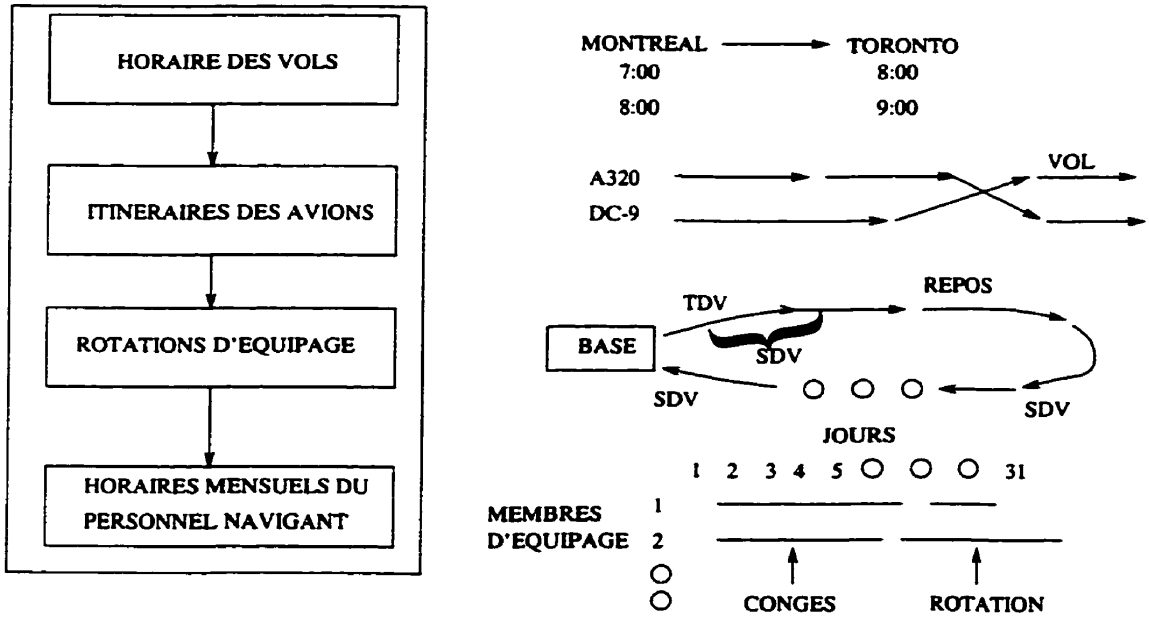


Figure 2.1 – *Étapes de la planification en transport aérien*

2.2 La fabrication des horaires mensuels

Il y a trois différentes méthodes pour construire des horaires mensuels, la méthode anonyme, la méthode personnalisée et la méthode respectant les préférences suivant l'ordre d'ancienneté.

1. Dans la méthode anonyme dite **Bidline**, des horaires anonymes couvrant chacune des rotations sont construits dans un premier temps, puis les employés choisissent séquentiellement du plus senior au plus junior leurs horaires préférés.
2. Dans la méthode personnalisée dite **Rostering**, la construction d'horaires mensuels personnalisés, doit permettre d'affecter des rotations, des périodes de repos aux membres d'équipages en prenant en compte une liste d'activités pré-assignées. Une activité pré-assignée est une activité telle qu'une période d'entraînement, des vacances, un rendez-vous médical, une activité de transition (activité qui a commencé le mois précédent et finit au début du mois courant), une activité de réserve. Ces activités doivent être nécessairement incluses à un moment donné dans l'horaire mensuel d'un employé donné.
3. Dans la méthode respectant les préférences suivant l'ancienneté dite **Preferential Bidding**, des horaires personnalisés sont construits comme dans la méthode personnalisée, mais en prenant également en compte un ensemble de demandes pondérées, appelées "bids", reflétant les préférences de chaque membre d'équipage de même qu'un ensemble d'activités pré-assignées devant être inclus dans son horaire. Dans la suite de ce mémoire cette méthode sera appelée **PBS** à cause de son nom anglais **P**referencial **B**idding **S**ystem.

Par exemple, un employé peut indiquer ses préférences pour une rotation spécifique, une période de repos, un fin de semaine libre, des départs après 8 heures du matin, etc... Une demande peut aussi concerner tout un horaire; par exemple un pilote peut préférer un horaire avec au minimum 74 heures de vol. Chez Air Canada, près de 75 demandes différentes peuvent être faites.

Ainsi chaque employé associe un poids à chacun de ses souhaits, et ces poids sont utilisés pour évaluer le pointage de ses horaires potentiels. L'objectif du logiciel PBS est d'assigner, par ordre d'ancienneté stricte, le meilleur horaire personnalisé afin de couvrir toutes les rotations qui doivent être effectuées durant le mois et de construire un horaire avec le pointage maximal pour chaque employé.

Comme nous l'expliquerons plus en détails dans les sections suivantes, la solution doit permettre d'assigner le meilleur horaire à chaque employé en respectant les préférences de chacun et l'ancienneté. Le système PBS traite le problème en résolvant pour chaque employé, un problème d'optimisation impliquant aussi les employés moins seniors. Le temps de calcul croit donc rapidement avec le nombre d'employés. Les quelques méthodes présentées dans la littérature jusqu'à présent étaient heuristiques.

2.3 La méthode respectant les préférences suivant l'ancienneté

Le système **PBS** qui est actuellement en place chez Air Canada, est utilisé pour construire les horaires mensuels pour les PNT. Étant donné que chaque PNT est qualifié pour voler sur un seul type d'avion, les problèmes sont divisés en autant

d'instances qu'il y a de types d'appareils. De plus les problèmes sont divisés suivant les bases de la compagnie. Ainsi un problème implique un groupe d'employés affectés à une même base et remplissant la même fonction à bord d'un type d'avion particulier. Avec ce partitionnement, généralement la taille des problèmes se trouve entre 20 et 200 employés pour certaines compagnies. La version PBS 200 utilisée pour résoudre ces problèmes de petites et moyennes tailles est suffisante.

La solution doit également satisfaire aux règlements des conventions collectives qui gouvernent la composition de chaque horaire, et à deux types de contraintes, des contraintes locales à respecter dans le sous problème ainsi que des contraintes globales à respecter dans le problème maître. Ces règles pouvant être modifiées à l'occasion, elles sont modélisées sous forme de paramètres.

1. Contraintes locales :

   Les contraintes locales régissent la construction des horaires individuels. Voici une courte description de certaines de ces règles. Les paramètres présents dans ces règles changent selon la compagnie. Ceux présentés dans la description, sont les paramètres actuellement utilisés chez Air Canada. Ces paramètres peuvent être modifiés et des valeurs différentes sont utilisées par d'autres compagnies .

   Crédits de vol ou *Flight credits* : Les crédits de vol sont les heures de vols et les crédits pour des activités comme les périodes d'entraînement, réunions, vacances annuelles, etc... La somme de ces crédits doit se situer entre 70 et 78 heures.

Jours travaillés ou *Working days* : Le nombre de jours travaillés consécutifs entre deux périodes de repos est au plus de 7 jours.

Jours de congé ou *Test periods* : Un employé doit avoir au moins 10 jours non travaillés pendant le mois.

Indice de fatigue ou *Fatigue factor* : Le facteur de fatigue est évalué en utilisant un score qui augmente avec chaque rotation effectuée et qui décroît avec les périodes de repos ; il ne doit pas être supérieur à 1500 au début de n'importe quelle rotation et ne doit jamais excéder 3000 au cours d'un mois.

Il y a 4 types d'horaires : normal, court, vide et ouvert. Un horaire qui satisfait toutes les contraintes et qui a au moins 72 heures de crédits de vol, est appelé un horaire normal (en anglais normal schedule). Un horaire réalisable pour lequel, les crédits de vol sont entre 70 et 72 heures est appelé un horaire court (en anglais low time). Si la construction de l'horaire d'un employé ne peut pas respecter une contrainte, cet employé est temporairement assigné à un horaire vide (en anglais empty schedule), et devient un membre de l'équipe de réserve. Finalement, un horaire ouvert (en anglais open time) est un horaire sans titulaire qui peut être utilisé pour couvrir une séquence de rotations non affectées qui ne peuvent être assignées à l'horaire d'aucun employé. Les rotations incluent dans les horaires ouverts, sont utilisées par la suite pour construire les horaires des membres de l'équipe en réserve.

2. Contraintes globales :

   Les contraintes globales concernent la composition de l'ensemble des horaires construits. Elles sont regroupées en 5 catégories.

Partition ou *Set Partitionning* : Chaque rotation doit être attribuée à un et un seul employé.

Choix des horaires ou *Schedule assignment* : A chaque employé, on doit attribuer un horaire normal, court ou vide.

Horaires courts ou *Low time schedules* : La solution ne peut contenir qu'au plus 2 horaires courts.

Horaires ouverts ou *Open time schedules* : La solution ne peut contenir qu'au plus 2 horaires ouverts.

Durée totale des horaires ouverts ou *Open time duration* : Les rotations non-couvertes ne durent pas plus de 120 heures.

La séparation entre les contraintes locales et globales est un prélude au processus de résolution basé sur la génération de colonnes. En effet, le problème de coordination appelé problème maître (PM) conserve toutes les contraintes globales sur l'ensemble des horaires alors que les générateurs d'horaires (les sous-problèmes) tiennent compte des contraintes locales. Des détails sur l'approche de résolution seront fournis dans les sections suivantes.

Chapitre 3

Revue de littérature

Introduction

Dans les pages qui suivent, nous allons présenter une revue de littérature des méthodes utilisées pour résoudre les problèmes de plus courts chemins avec contraintes, ainsi que les modèles utilisés pour résoudre les problèmes de construction d'horaires.

A la section 3.1, nous introduirons le modèle classique du problème de partitionnement d'ensemble.
A la section 3.2, nous nous intéresserons plus particulièrement aux approches de résolution par génération de colonnes.
A la section 3.3, nous présenterons une revue de littérature des articles portant sur les problèmes de plus courts chemins avec contraintes.
Finalement à la section 3.4, nous présenterons la méthode de génération de colonnes appliquée au problème PBS.

Dans le passé, les problèmes de planification d'horaires d'équipages dans un contexte aérien étaient résolus à l'aide de méthodes manuelles ou approximatives basées sur des modèles de programmation linéaire en nombres entiers. Les méthodes manuelles impliquaient un processus d'essais et erreurs, dont les résultats d'une période étaient souvent adaptés pour résoudre la période suivante. Bien que les

méthodes heuristiques reposaient sur l'usage des ordinateurs, elles pouvaient nécessiter également l'intervention d'un grand nombre d'employés aussi bien pour trouver une solution initiale que pour rendre réalisable des résultats produits sans respect de l'ensemble des contraintes. Une autre approche a été dans un premier temps de rendre réalisable une solution initiale puis de l'améliorer en utilisant des procédures d'échanges jusqu'à l'obtention d'une solution optimale. En procédant de la sorte il était facile de converger rapidement vers la solution optimale pour de petits problèmes, l'optimalité pouvant être atteinte en testant les résultats de toutes les solutions possibles, mais pour de gros problèmes, les solutions ne pouvaient être toutes explorées vu leur grand nombre.

Ces méthodes d'échanges ont naturellement permis de réduire le nombre d'avions et de personnels requis pour couvrir un horaire, cependant elles n'atteignaient pas leurs buts ultimes d'optimiser l'utilisation des avions, des équipages, du temps et du capital. Ainsi, la procédure manuelle exigeait du temps et d'énormes efforts pour un résultat souvent discutable. Les méthodes heuristiques quant à elles pouvaient trouver des solutions en moins de temps, mais encore sous-optimales.

Étant donné la complexité des problèmes réels, les solutions ont longtemps été sous-optimales. L'approche proposée génère une solution optimale qui peut facilement s'adapter à une modification des données d'entrées, comme une modification de l'horaire initiale des avions, des conventions collectives ou du nombre d'avions. L'usage de méthodes manuelles étaient en effet mal adaptées à des modifications étant donné le temps relativement long pour trouver ne serait-ce qu'une solution initiale. Or ces genres de perturbations surgissent très fréquemment non seulement au cours de la planification mais aussi durant le jour d'opération. D'autres

modèles d'optimisation ont été développés pour réoptimiser les horaires d'équipage dans ces situations (voir Mirela Stojkovic 1998). Cependant, les solutions optimales de modèles mathématiques sont encore peu utilisées en pratique et des solutions sous-optimales produites manuellement sont souvent retenues.

3.1 Le modèle classique

La formulation classique utilisée pour la construction de rotations d'équipage est de type partitionnement d'ensemble. Les valeurs des paramètres mentionnées ici sont ceux de la compagnie Air Canada. D'autres valeurs sont utilisées ailleurs.

Soit R, indicé par r, l'ensemble des horaires mensuels réalisables représentés dans le programme linéaire (PL), r représente un horaire.

Soit $R(k)$, l'ensemble des horaires que l'on a bâti pour le pilote k, avec $1 \leq k \leq m$, m étant le nombre d'employé.

Soit $R(O/T)$, l'ensemble des horaires ouverts (en anglais open time).

Soit C, le coût de chacun des horaires dépendant de l'ancienneté et des préférences accordées à une rotation particulière.

Soit P, indexé par p , l'ensemble des rotations à couvrir, p représente alors une rotation.

Soit w, le nombre d'horaire court (en anglais low time) qui peuvent être construit

avec $w = 2$

Soit $l_r = 1$ si l'horaire r est un horaire court et 0 sinon.

Soit d_r, la durée d'un horaire r si c'est un horaire ouvert (en anglais open time) et 0 autrement.

Définissons par c_r le coût d'un horaire et par a_{pr} le paramètre binaire valant 1 si, l'horaire r couvre la rotation p, et 0 sinon.

Finalement, définissons X_r la variable de décision binaire associée à l'horaire r ; X_r prend la valeur 1 si l'horaire r est affectée à un employé, et 0 sinon.

$$(IP_k) \min_{X_r} \sum_{r \in R} c_r X_r \tag{3.1}$$

$$s.\,c.: \quad \sum_{r \in R} a_{pr} X_r = 1, \ \forall p \in P \tag{3.2}$$

$$\sum_{r \in R(k)} X_r = 1, \ \forall\, k \textit{ en devoir ou en réserve} \tag{3.3}$$

$$\sum_{r \in R} l_r X_r \ \leq w \tag{3.4}$$

$$\sum_{r \in R(O/T)} X_r \ \leq 2 \tag{3.5}$$

$$\sum_{r \in R(O/T)} d_r X_r \ \leq 120 \tag{3.6}$$

$$X_r \in \{0,1\}, \ \forall r \in R \tag{3.7}$$

La fonction objectif (3.1) vise la minimisation du coût des horaires pour couvrir l'ensemble des rotations. Les contraintes (3.2) assurent que les rotations soient toutes couvertes exactement une fois par l'horaire d'un employé. Les contraintes (3.3) sont

les contraintes dites de "cadres", c'est-à-dire chaque employé est affecté à un horaire, qu'il soit en *devoir* ou en *réserve*. Les contraintes (3.4) représentent les contraintes sur les *low time schedules*, (3.5) sur les *open time schedules* et (3.6) sur les *open time duration*. Les contraintes d'intégrités sur les variables de décision sont imposées en (3.7).

3.2 Les approches de résolution par génération de colonnes

Cette approche permet de résoudre les problèmes de construction de rotations (pairing), d'horaires dans le cas anonyme (Bidline), dans le cas personnalisé (rostering) et est utilisée à l'intérieur de procédures plus complexes pour résoudre le problème PBS.

L'objectif est de trouver un ensemble d'horaires valides à coûts minimums permettant de couvrir toutes les rotations exactement une fois. Les horaires correspondent à des chemins dans un réseau, où se déplacent des équipages. Ces horaires sont dits admissibles ou valides lorsqu'ils respectent la réglementation appropriée. Dans le contexte de routage de véhicules, les chemins sont des itinéraires valides à effectuer par les camions et les tâches à couvrir, sont les clients à desservir (voir Desrochers, Desrosiers et Solomon 1992) ou encore, dans le contexte des chauffeurs d'autobus, les chemins sont des horaires pour les chauffeurs et les tâches à couvrir sont des portions de trajet d'autobus (voir Desrochers et Soumis 1988).

Les approches de résolution par génération de colonnes consistent à résoudre

itérativement, d'une part, un problème d'optimisation appelé sous-problème servant à générer au besoin des horaires prometteurs et, d'autre part, la relaxation linéaire du modèle (3.1)-(3.7) restreint à un sous-ensemble d'horaires, appelé problème maître restreint. La différence entre ces approches et les méthodes heuristiques proposées dans Ball et Roberts(1985), Gershkoff(1989) et Graves *et al.*(1993) pour le problème de rotations, réside dans le fait que, dans les approches par génération de colonnes, le générateur d'horaires est guidé par l'information duale correspondant à la solution courante du problème maître restreint pour produire des horaires qui permettent d'améliorer la valeur courante de la fonction objectif. De plus, l'optimalité de la solution courante peut être validée par la résolution de sous-problème, lorsque ce dernier ne permet plus de générer des horaires de coût marginal négatif. Ainsi la méthode de génération de colonnes produit la solution optimale en considérant implicitement tous les horaires plutôt que de produire une rotation à partir d'un sous ensemble d'horaires générés à priori.

Le logiciel utilisé pour résoudre ce problème par génération de colonnes s'appelle GENCOL. Ce logiciel fait appel à un algorithme de séparation et d'évaluation progressive pour trouver une solution en nombres entiers. À chaque noeud de l'arbre de branchement, une borne inférieur est calculée, à l'aide de l'approche de génération de colonnes, en résolvant la relaxation linéaire de la formulation (3.1)-(3.7). Pour la plupart des problèmes réels, la méthode du simplexe est pratiquement inutilisable pour résoudre directement cette relaxation linéaire à cause du très grand nombre de variables que ces problèmes contiennent.

La méthode de génération de colonnes permet d'éviter l'énumération explicite de toutes les variables en décomposant la relaxation linéaire, c'est-à-dire le problème

maître, en deux modules : un problème maître restreint et un sous-problème (qui lui-même peut être souvent séparé en plusieurs sous-problèmes). Le problème maître restreint est identique au problème maître à l'exception qu'il considère seulement un petit sous-ensemble de variables. Son rôle est de déterminer une solution optimale pour ce sous-ensemble de variables et de fournir au sous-problème les variables duales correspondant à cette solution.

Dans GENCOL, le problème maître restreint est résolu par la méthode du simplexe, primale ou duale selon les applications. Le sous-problème se formule comme un problème de plus court chemin avec variables de ressource. Son rôle consiste à générer des variables de coût marginal négatif (dans le cadre d'une minimisation) pour le problème maître. Chacune de ces variables est associée à un horaire valide dans notre contexte. Le sous-problème est résolu par divers algorithmes de programmation dynamique, selon le nombre de ressources et ses particularités. L'article synthèse de Desrosiers *et al.* (1995) en mentionne plus de dix versions plus ou moins spécialisées.

Pour mieux visualiser les problèmes d'horaires, ils sont représentés sous forme de réseaux à l'aide d'un générateur de réseaux. Ces réseaux sont une série de nœuds et d'arcs ou les chemins dans ces réseaux représentent les horaires.

Les chemins sont générés par les sous-problèmes et correspondent aussi bien à des variables dans le problème maître qu'aux colonnes de la matrice dans la formulation du problème de partitionnement d'ensemble. Une fois que ces chemins sont validés, ils serviront comme donnée d'entrée pour le problème maître.

Le processus de résolution de GENCOL procède de façon itérative résolvant tantôt le problème maître tantôt le sous-problème. À chaque itération, le problème maître est résolu. Les contraintes globales sont conservées et prises en compte dans le problème maître tandis que les contraintes locales dans le sous-problème. Les contraintes sont généralement modélisées dans le sous-problème sous forme de ressources. Une ressource est une commodité consommée à chaque fois qu'un arc est emprunté. La consommation de ressource se fait sur les arcs et la validation des restrictions de ressource sur les nœuds. Dès lors, une chemin est dit valide si la consommation de ressource cumulée le long du chemin respecte les restrictions à chaque nœud. L'information concernant la consommation cumulée de ressource est placée dans une étiquette.

Les variables duales associées à la solution optimale courante sont transmises au sous-problème afin de lui permettre de trouver le chemin valide ayant le plus petit coût marginal. Le sous-problème est ensuite résolu. Si le coût marginal de la solution du sous-problème est négatif, alors une ou plusieurs variables associées aux chemins de coût marginal négatif sont ajoutées au sous-ensemble courant du problème maître restreint . Les horaires sont générés par chacun des sous-problèmes, un par employé. Le problème maître restreint est alors de nouveau résolu. Le processus itératif se termine lorsque le coût marginal de la solution du sous-problème est non-négatif. En effet, cela démontre qu'aucun chemin non considéré ne peut contribuer à l'amélioration de la solution optimale courante du problème maître restreint, prouvant ainsi l'optimalité de cette dernière pour le problème maître.

Des fonctions jouant le rôle de filtres s'assurent de la validité des chemins, du prolongement des étiquettes dans le réseau du PCC et du calcul du coût d'un chemin.

Il existe deux types de filtres, les filtres internes dans le sous-problème et externes dans le problème maître. Le fonction principale du filtre interne est le prolongement des étiquettes contenant les informations relatives aux consommations de ressources au fur et à mesure de la construction des chemins. Les filtres internes reçoivent des étiquettes et l'information concernant l'arc suivant. Il crée alors une nouvelle étiquette et la valide si la consommation de ressource jusqu'à ce point se trouve dans les limites permises. Dans le cas contraire le nouveau chemin est invalide et rejeté.

Les chemins valides qui ont été générés par le sous-problème servent de donnée d'entrée pour le filtre externe. Il s'assure qu'aucune contrainte ne soit violée. Les données de sortie de ce filtre serviront comme colonne dans le problème maître. La structure générale des modules de GENCOL est illustrée à la figure 3.1.

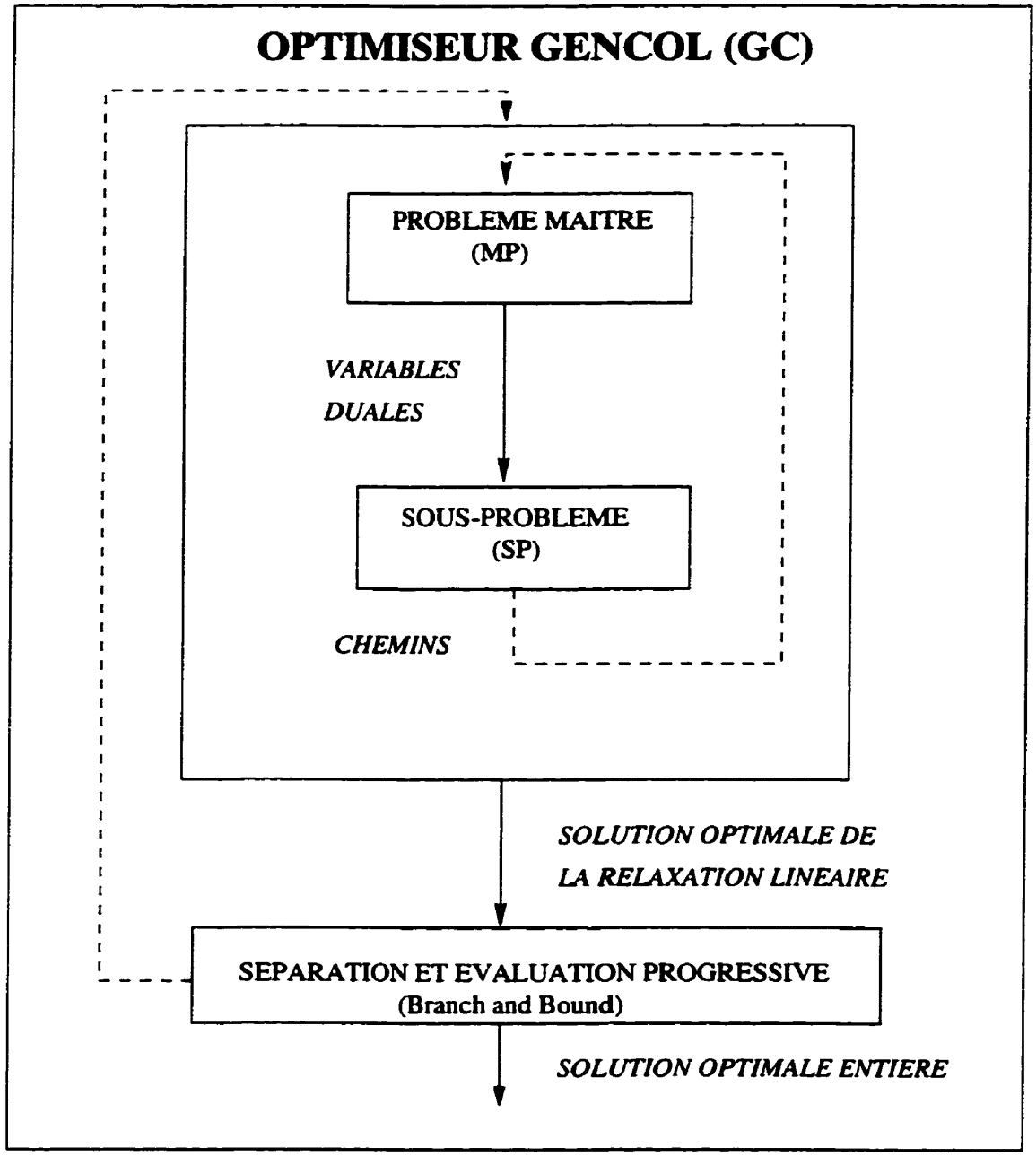


Figure 3.1 – *Principaux modules de l'optimiser GENCOL*

3.3 Les sous-problèmes

Au cours d'une itération, c'est-à-dire lors de la résolution du problème résiduel (IP_k), il y a un sous-problème pour chaque employé. Chaque sous-problème possède une structure de réseau. Les ressources utilisées permettent de modéliser certaines restrictions qui n'ont pu être traité explicitement lors de la construction du réseau. Dans le cas qui fera l'objet de ce rapport, les rotations sont représentées dans le réseau par des arcs. Plusieurs ressources sont utilisées : le nombre de crédits de vols, le nombre de jours non travaillé et le nombre de jours de travails consécutifs. La consommation des ressources est faite sur les arcs et les limites sont imposées sur les nœuds. C'est d'ailleurs sur les nœuds que se fait la validation des horaires. La validation est une vérification qui assure que l'intervalle de faisabilité (restriction de la ressource) ne soit pas violé. La validation se fait en comparant une valeur cumulée de la ressource avec l'intervalle de faisabilité. Un horaire est donc valide s'il ne viole pas les bornes de ressources modélisant la réglementation du personnel navigant.

Un chemin P_i est dit réalisable s'il correspond à un horaire légal. Le problème de plus court chemin avec contraintes de ressources consiste à trouver un chemin réalisable de coût minimum allant de l'origine o à la destination d. Ce problème a été introduit par Desrochers [1986] comme une généralisation multi-dimensionnelle du problème de plus court chemin avec fenêtres de temps (SPPTW). Le problème de plus court chemin avec contraintes de ressources (SPPRC) apparaît comme un sous-problème de la décomposition de Dantzig-Wolfe (génération de colonnes), utilisé pour résoudre les problèmes de tournées de véhicules, problèmes de planification de flotte d'avions, problèmes d'horaires d'équipages et bien d'autres problèmes d'ho-

raires et tournées.

Plusieurs synthèses des applications et méthodes de résolution du problème de plus court chemin avec contraintes de ressources ou ses variantes ont été décrites dans la littérature (Minoux 1975, Handler et Zang 1980, Aneja et al. 1983, Desrosiers et al. 1983, Jaffe 1984, Desrochers 1986, Desrochers et Soumis 1988a,b, Maculan et Salles 1991, Desrosiers et al. 1995, Desaulniers et Villeneuve 1997).

Dans chaque sous-problème, un problème de plus court chemin avec contraintes de ressources est résolu sur un graphe acyclique. Pour e = k,...,m, soit $G_e^k = (V_e^k, A_e^k)$ le graphe d'un employé e, où V_e^k est l'ensemble des nœuds tandis que A_e^k est l'ensemble des arcs. L'ensemble des nœuds V_e^k est composé de $N_e^k \cup \{o, d\}$ où N_e^k est l'ensemble des nœuds qui peuvent être visités du nœud source noté o, au nœud puits noté d. Ces deux derniers nœuds représentent respectivement le début et la fin d'une période d'un mois. Les arcs de A_e^k correspondent aux activités résiduelles du mois comme des rotations, des périodes de repos et des périodes d'entraînement.

À l'itération k, chaque horaire pour l'employé e peut être représenté par un chemin $o - d$ dans G_e^k. L'inverse n'est pas vrai sachant qu'un chemin ne respecte pas forcément toutes le contraintes locales essentielles pour un horaire légal. On désigne par R l'ensemble des ressources utilisées pour modéliser les contraintes locales de PBS. Le nombre de crédits de vols, le nombre de jours non travaillés et le nombre de jours de travails consécutifs sont des exemples de contraintes de ressources considérées dans le problème. Soit t_{ij}^r la valeur représentant les unités de ressources r consommées sur l'arc (i, j), pour tout $r \in R$. Une contrainte d'intervalle $[l_i^r, u_i^r]$ est associée avec chaque ressource $r \in R$ et avec chaque nœud $i \in V_e^k$.

Le processus de génération des colonnes admissibles (fabrication d'horaires), se fait dans le sous-problème de chaque employé. Dans la description qui suit, nous ne faisons pas intervenir d'indice e pour chaque employé afin de ne pas alourdir la notation.

Les variables du sous problème sont de deux types : les variables de flots classiques $X_{ij}, (i,j) \in A$, définies sur les arcs du réseau, et les variables de ressources $T_i^r, i \in R$ et $r \in R$ définies sur les nœuds du réseau, pour chacune des ressources de l'ensemble R. La variable de flot X_{ij} prend la valeur 1 si l'arc $(i,j) \in A$ est utilisé par le chemin de l'origine o à la destination d, et 0 sinon. La variable de ressource $T_i^r, i \in R$ et $r \in R$ est restreinte à l'intervalle de valeur $[\, l_i^r, u_i^r \,]$; elle prend des valeurs entières si les fonctions qui la définissent imposent de telles valeurs. Les fonctions de prolongation $f_{ij}^r(T_i^r)$, définies pour chaque arc $(i,j) \in A$ et chaque ressource $r \in R$, permettent de calculer les variables de ressource à chaque noeud j compte tenu de l'information au nœud i.

Le coût marginal d'une colonne est fonction du chemin emprunté, de la valeur des variables de ressource et de l'information duale provenant du problème restreint. Sa forme générale est donc $c(\mathbf{X},\mathbf{T};\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\epsilon},\boldsymbol{\lambda})$, où $\mathbf{X} = (X_{ij}, (i,j) \in A)$ et $\mathbf{T} = (T_i^r, i \in R \text{ et } r \in R)$, alors que $\boldsymbol{\alpha} = (\alpha_p, p \in P)$, $\boldsymbol{\beta} = (\beta_k, 1 \leq k \leq m)$, $\boldsymbol{\delta}$, $\boldsymbol{\epsilon}$,$\boldsymbol{\lambda}$ sont les vecteurs de variables duales associées respectivement aux contraintes (3-2), (3-3), (3-4), (3-5) et (3-6).

La formulation du sous-problème se pose alors somme suit :

$$\min \quad c(\mathbf{X}, \mathbf{T}; \alpha, \beta, \delta, \epsilon, \lambda) \tag{3.8}$$

$$s.\ c.: \quad \sum_{j;(o,j)\in A} X_{oj} = \sum_{j;(j,d)\in A} X_{jd} = 1 \tag{3.9}$$

$$\sum_{j;(i,j)\in A} X_{ij} = \sum_{j;(j,i)\in A} X_{ji} = 0\ ;\ \forall i \in N \tag{3.10}$$

$$X_{ij}(f^r_{ij}(T^r_i) - T^r_j) \leq 0\ ,\ \forall r \in R,\ \forall (i,j) \in A \tag{3.11}$$

$$l^r_i \leq T^r_i \leq u^r_i\ ,\ \forall r \in R,\ \forall i \in V \tag{3.12}$$

$$X_{ij} \in \{0,1\}\ ,\ \forall (i,j) \in A \tag{3.13}$$

La fonction objectif (3.8) vise à minimiser la fonction de coût. Les relations (3.9)-(3.10) sont les contraintes classiques de chemin, de l'origine o vers la destination d. Les relations (3.11) décrivent la compatibilité entre les variables de flot et les variables de ressource. Ce sont des contraintes non linéaires semblables aux équations (non linéaires) de complémentarité en programmation linéaire. Les contraintes (3.12) indiquent le domaine des variables de ressource et de flot et (3.13) les contraintes d'intégrités des variables.

Dans notre cas, le graphe est acyclique et les nœuds sont classés dans un ordre topologique. Pour résoudre le problème de plus court chemin avec contraintes de ressources, Desrochers et Soumis (1988a) proposent un algorithme de programmation dynamique pseudo-polynomial. L'algorithme de programmation dynamique traite séquentiellement les nœuds de la source jusqu'à la destination. En chaque nœud, il génère des étiquettes en prolongeant les chemins correspondant aux étiquettes efficaces présentes aux nœuds prédécesseurs. Les prolongements qui fournissent des

chemins illégaux sont ensuite supprimés. Puis il applique la règle de dominance pour éliminer tous les chemins correspondants à des étiquettes non efficaces.

Soit $F(S,i,t)$ le coût minimum du chemin allant de l'origine o au nœud i, $i \in N \cup \{d\}$, passant par tous les nœud de $S \subseteq N \cup \{d\}$ et arrivant au nœud i au temps t ou plus tard. Le calcul du coût $F(S,i,t)$ se fait en résolvant les relations de récurrence suivantes :

$F(0,o,l_o) = 0$

$F(S,j,t) = \min_{(i,j)\in A} \{F(S-\{j\},i,t') + c_{ij} \mid i \in S-\{j\},\ t' \leq t - t_{ij},\ l_i \leq t' \leq u_i\}$,

$\forall S \subseteq N \cup \{d\},\ j \in S,\ l_j \leq t \leq u_j$.

La solution optimale est donnée par

$\min_{S\subseteq N\cup\{d\}} \min_{a_d \leq t \leq b_d} F(S,d,t)$.

Ainsi l'algorithme de programmation dynamique procède en trois grandes étapes. En chaque nœud $j \in N$, il effectue les opérations suivantes :

1 - prolongation du chemin (génération des étiquettes),

Ainsi dans un algorithme de programmation dynamique de type "pulling", la génération des étiquettes se fait comme suit. Pour un nœud j donné, des étiquettes sont crées en prolongeant celles réalisables et présentes aux nœuds i, tels que $(i,j) \in A$. Plus précisemment, une nouvelle étiquette (T_j, C_j) donnée par

$T_j = max\{l_j, T_i + t_{ij}\}$

$C_j = C_i + c_{ij}$

est créée au nœud j si $T_i + t_{ij} \leq u_j$

2 - filtrage (test de réalisabilité),

On s'assure dans cette étape que le chemin est légal à chaque nœud, c'est-à-dire que l'on vérifie que les consommations de ressource des variables de ressource restent à l'intérieur des intervalles de valeur $[\ l_i^r, u_i^r\ ]$ en chaque nœud. Les prolongements qui fournissent des chemins illégaux sont ensuite supprimés.

3 - dominance (élimination des étiquettes non efficaces).

Soit $P_i^{(k)}$ le $k$ème chemin $P_i^{(k)}$ de l'origine o au nœud i. On lui associe une étiquette $(T_i^{(k)}, C_i^{(k)}) = (T_i^{1(k)}, T_i^{2(k)}, ..T_i^{n(k)}, C_i^{(k)})$ représentant son état des ressources et son coût.

Soit $(T^{(1)}, C^{(1)})$ et $(T^{(2)}, C^{(2)})$, deux étiquettes associées à deux chemins réalisables de o à i. On dit que $(T^{(1)}, C^{(1)})$ domine $(T^{(2)}, C^{(2)})$ et on note $(T^{(1)}, C^{(1)}) \preceq (T^{(2)}, C^{(2)})$, si et seulement si $C^{(1)} \leq C^{(2)}$ et $T^{(1)} \leq T^{(2)}\ \ \forall (r \in R)$.

Une étiquette associée à un chemin réalisable de o à i, est dite efficace si il n'existe pas d'étiquette en i qui la domine. Un chemin est dit efficace s'il est associé à une étiquette efficace.

La relation $\preceq$ étant une relation d'ordre partiel, le nombre d'étiquettes efficaces à traiter augmente d'une façon exponentielle en fonction du nombre de ressources. Ce qui rend la procédure de prolongation très ardue. La taille de l'espace de recherche peut être réduite, en augmentant l'élimination de la procédure de dominance. Cela peut se faire par exemple, en réduisant la dimension de l'espace des ressources, ou bien en définissant de nouvelles règles de dominance ou de nouvelles fonctions de prolongation, voir Nagih (1999).

3.4 La génération de colonnes appliquée à PBS

La génération de colonnes telle que décrite ci-dessus est utilisée pour les deux modes de construction des horaires mensuels : la construction d'horaires anonymes et d'horaires personnalisés. Pour la méthode PBS, avec préférences et contraintes d'ancienneté, il faut construire un horaire de pointage maximal pour chaque employé de telle sorte que cette optimisation ne soit jamais faite au détriment d'un employé plus senior. Or comme Gamache *et al.* (1998) l'ont montré, cela présente une difficulté. En effet, dans une formulation directe de la fonction économique, les poids doivent être choisis de façon à pouvoir différencier les requêtes sélectionnées par un employé ainsi que refléter l'ancienneté des employés. Étant donné le nombre élevé de requêtes possibles et afin de pouvoir les distinguer entre elles dans la formulation de l'objectif, un variable de 32 bits est nécessaire pour différencier les requêtes choisies par un seul employé. Ainsi, avec une fonction objectif exprimée à l'aide d'une variable de 32 bits, la valeur de cet objectif, reflétant les préférences d'un employé est un entier décimal qui peut être aussi grand que 2^{32} soit $\simeq 4 * 10^9$.
Ainsi pour exprimer l'ancienneté entre deux employés lors de la résolution du problème, la fonction objectif nécessite deux variables (mots) d'ordinateur de 32 bits chacun (un pour chaque employé), soit 64 bits, c'est-à-dire un nombre proche de $2^{64} \simeq 10^9$. En généralisant à m employés et afin d'exprimer les différences entre les employés, la fonction économique peut prendre des valeurs proche de 2^{m*32} pour respecter les contraintes d'ancienneté entre les m employés. Ainsi ce nombre élevé, nous pousse à recourir à une approche séquentielle afin de résoudre ce type de problème.

En raison des contraintes d'ancienneté qui impose la construction d'un horaire

de l'employé le plus senior au plus junior, il semble tout indiquer d'adopter une approche de résolution séquentielle. Cependant, la recherche de l'horaire optimal pour un seul employé est en soi difficile. D'une part, on cherche à satisfaire au maximum cet employé, d'autre part, il faut un ensemble d'horaire couvrant toutes les rotations.

Il semble que dans la littérature, il n'existe que deux articles pertinents traitant de la construction d'horaires mensuels personnalisés dans le domaine du transport aérien. Moore, Evans et Ngo (1978) pour Qantas, et Byrne (1988) pour CP Air ont proposé des heuristiques pour construire séquentiellement un bon horaire pour chaque employé, du plus senior au plus junior en prenant en compte une liste de requêtes pondérées pour chaque employé. La méthode proposée est la suivante. Pour un employé donné, l'activité légale la plus désirée est assignée en premier à l'horaire appelé l'horaire "squelette" c'est-à-dire en construction. Le processus continue avec l'activité légale suivante la plus attrayante. Pour valider l'horaire en cours, on vérifie si, pour chaque jour du mois, il est encore possible de couvrir toutes les rotations résiduelles avec les employés restants, c'est-à-dire si le nombre d'employé est égal au nombre de rotations. On arrête l'ajout de nouvelles activités à l'horaire squelette lorsqu'un horaire admissible a été construit. Cet horaire est alors attribué à l'employé traité et toutes les rotations qui ont été utilisées, sont retirées de l'ensemble des rotations résiduelles. Lorsque tous les horaires ont été construits, des échanges d'activités entre les horaires sont permis de façon à améliorer la solution d'ensemble.

L'approche précédente présente quelques inconvénients. Par exemple, en juxtaposant les activités les plus désirées les unes à la suite des autres, on ne garantit

pas que l'horaire squelette soit optimal ni même admissible. Dans ce dernier cas, il faudra faire marche arrière pour cet horaire en construction. De plus, puisque la validation est incomplète, le processus d'optimisation pour l'ensemble des employés peut lui aussi devoir faire marche arrière. Même si une solution partielle est optimale, il pourrait exister d'autres horaires optimaux qui permettraient de construire de meilleurs horaires pour les employés junior.

La méthode exacte utilisée actuellement dans le logiciel PBS utilise également un processus séquentiel basée sur l'ancienneté. Pour chaque employé, du plus senior au plus junior, un problème résiduel est résolu. Pour un employé donné et un ensemble d'horaire déjà affecté, nous formulons un programme linéaire en nombres entiers pour couvrir les rotations résiduelles en prenant en compte tous les employés restants. Le problème résiduel est résolu par une méthode de génération de colonnes intégrée à un processus de séparation et d'évolution progressive. La solution de ce problème détermine une solution optimale pour cet employé tout en assurant qu'il existe des horaires admissibles pour tous les employés restant.

Ainsi, l'approche de résolution de Gamache *et al.* (1998) utilisée dans PBS, consiste à résoudre plusieurs fois le problème, autant de fois qu'il y a d'employés, en utilisant le principe de génération de colonnes. Lorsqu'une rotation fait partie de l'horaire d'un employé, elle n'est plus disponible pour les employés suivants. Pour modéliser ceci, deux approches sont utilisées. La première est heuristique et consiste à fixer dans le problème maître la colonne représentant l'horaire pour le premier employé. Ainsi, toutes les rotations faisant partie de cette horaire sont alors retirées de l'ensemble des rotations, restant à être affectées et ne sont plus disponibles pour le reste des employés. La seconde approche consiste à nettoyer les réseaux au fur et

à mesure de la résolution. C'est-à-dire réduire du réseau de l'employé suivant, les rotations formant un horaire au pointage identique (aux yeux de la fonction objectif) à celui de l'horaire optimal ayant été déterminé pour l'employé précédent. En effet, la maximisation de la satisfaction pour l'employé senior, peut ne pas être optimale pour l'ensemble des employés. Dès lors, l'élaboration d'une solution optimale pour tous les employés n'est pas chose facile. Les solutions entières sont obtenues en utilisant des coupes originales très efficaces.

Nous verrons au chapitre 4, que plusieurs versions du logiciel PBS existent ou sont en développement :

1 : PBS 200 pour 200 employés

C'est la version de base qui est actuellement commercialisée.

2 : PBS 500 pour 500 employés

C'est une version en cours de développement qui relaxe une partie des contraintes de couverture de tâches "faciles" (jusqu'à 60%) pour accélérer la résolution du problème. Une contrainte est dite "facile" si il n'y a pas de rareté de ressources pour cette tâche, c'est-à-dire qu'elle est largement désirée. Ces contraintes sont réintroduites dynamiquement au besoin si elles ne sont pas satisfaites. Ce prototype a permis d'obtenir d'aussi bonnes solutions que la version de base sur quelques instances d'Air Transat.

3 : PBS 5000 pour 5000 employés et plus

Cette version est en cours de développement. Dans cette approche, il n'y a plus de problème maître c'est-à-dire plus de programme linéaire, seuls les PCC de chaque réseau sont calculés. Les contraintes locales sont respectées dans les sous-problèmes lors de la construction d'un horaire. GENCOL n'intervenant

plus dans ce modèle, la réalisabilité de la solution, c'est-à-dire toutes les rotations couvertes et un horaire réalisable pour chaque employé, est obtenue par l'ajout de "compteurs" permettant d'assurer une solution réalisable pour chaque employé. Cette notion de " compteurs" sera largement expliquée au chapitre suivant.

Ces versions de PBS se distinguent essentiellement par leur méthode de résolution, permettant de trouver un horaire pour un nombre d'employés croissant. Cette évolution du logiciel pour traiter les plus grands problèmes réduit l'effort consacré à la résolution du PM, mais le travail de résolution du PCC reste le même. L'objectif de ce mémoire qui consiste à accélérer le PCC, est donc pertinent pour toutes ces versions.

Des stratégies d'accélération consistantt à résudre le PCC de façons heuristiques durant les premières itérations de la génération de colonnes et à ne pas résoudre tous les sous-problèmes à chaque itération de la génération de colonnes, ont été présenté par Gamache (1995). Ces stratégies sont déjà incluses dans la version de base du logiciel, mais ceci n'est pas suffisant, d'où la justification d'une méthode de réduction telle que présentée dans ce mémoire.

D'autre part, des méthodes d'accélération ont été testées au travers de la parallélisation des sous-problèmes, voir Pires (1997). Ces techniques sont pertinentes et permettent d'accélérer la résolution des problèmes en utilisant plusieurs procésseurs. Il faut dans notre contexte, résoudre des problèmes pour toutes les flottes et tous les types de personnel. Ainsi, il faudrait un très grand nombre de procésseurs si on utilisait plusieurs procésseurs pour chaque problème.
Les compagnies aériennes ont généralement moins de machines (procésseurs) que

de problèms à résoudre. La stratégie la plus efficace est donc d'utiliser une seule machine par problème.

Chapitre 4

Objectifs et méthodologie

Introduction

Avant de présenter les objectifs de ce mémoire, nous allons présenter le système PBS, ainsi que les nouveaux développements. Ceci permettra de situer les objectifs de ce mémoire dans un plan de travail impliquant d'autres chercheurs.

A la section 4.1, nous présenterons les objectifs généraux du logiciel PBS, ainsi que les nouveaux développements qui ont été implémentés dans la version PBS 500.
A la section 4.2, nous détaillerons les techniques de réduction utilisées pour réduire le réseau.
Finalement à la section 4.3, nous introduirons une étude de type probabiliste afin d'estimer la taille que pourrait avoir le réseau d'un problème de grande taille (5000 employés) après utilisation des techniques de réduction présentées à la section 4.2.

4.1 Le logiciel PBS

4.1.1 Objectifs généraux

Le système **PBS** développé par Ad Opt Technologies est un système commercial qui crée des horaires mensuels individuels pour du personnel navigant cabine et technique. Ce système présentement en utilisation chez Air Transat, Air Canada, TWA etc... se nomme *Altitude PBS 200* et prend en considération les préférences de chaque employé. L'architecture du produit se compose de trois principaux modules :

- **Bidder Graphical Interface :** Interface graphique utilisée par les membres d'équipage pour exprimer leurs préférences pour l'horaire du prochain mois au travers de plusieurs options.
- **Planner Graphical Interface :** Interface graphique utilisée par les planificateurs pour résoudre les problèmes de construction d'horaires mensuels.
- **Optimiseur :** Composante qui produit les horaires mensuels. L'optimiseur accommode d'une manière optimale les préférences de chaque membre d'équipage suivant les rotations à couvrir et les règlements à respecter. Enfin il produit des horaires pour les membres d'équipages actifs et ceux en réserves.

Un système PBS optimal doit construire un bloc (horaire) pour chaque employé dans l'ordre d'ancienneté strict et avoir les propriétés suivantes :

1 - Considérer tous les blocs possibles pour l'employé courant et choisir un des meilleurs.

2 - S'il y a plusieurs meilleurs blocs pour l'employé courant, choisir celui qui

permettra de maximiser la satisfaction des employés suivants.

3 - Un bloc est possible pour l'employé courant seulement si les tâches résiduelles peuvent être couvertes par les employés restants.

4.1.2 Nouveaux développements

L'utilisation de GENCOL, comme on l'a expliqué plus haut, sert principalement à s'assurer qu'un horaire est possible pour l'employé courant seulement si les tâhes résiduelles peuvent être couvertes par les employés restants.

Ainsi, dans la version PBS 500, on ne résout pas le problème maître pour les premiers employés. Cette relaxation est souvent acceptable car les employés les plus seniors peuvent obtenir les blocs désirés et les tâches résiduelles peuvent être couvertes par les employés restants. Si l'on s'aperçoit qu'un employé plus junior ne peut être contenté, une technique consiste à revenir en arrière (en anglais 'backtrack") pour réduire le nombre des premiers employés qui seront résolus sans GENCOL. Pour tous les autres employés, comme on l'a dit, on relaxe les contraintes de couverture pour les tâches faciles.

Étant donné les limitations de performance des versions *PBS 200* et *PBS 500* ou seuls des problèmes de tailles limitées sont résolus, respectivement 200 et 500 employés, la recherche s'oriente vers le développement d'une nouvelle version, *PBS 5000*, construisant des horaires mensuels pour des problèmes de tailles plus importants (3000 à 5000 personnes) et plus rapidement en une heure ou deux.

Cette approche vise à remplacer le problème maître, qui s'assure de la réalisabilité

globale, par un ensemble de "compteurs" qui pourrait assurer le respect des contraintes globales. En d'autres termes, développer un système de tabulateurs calculant la demande en personnel possédant certaines caractéristiques appelées "attributs", qui sont nécessaires pour faire les tâches résiduelles (rotations) durant un certain intervalle de temps et la disponibilité offerte par le personnel restant pour chaque attribut. Par exemple, il faudra que l'on s'assure tout au long de la résolution du problème, que le nombre d'employés nécessaires parlant l'Espagnol pour assurer les vols (la demande), soit inférieur ou égal, au nombre d'employés disponibles parlant l'Espagnol (l'offre).

Les caractéristiques ("attributs") comprendront :

- le personnel total ;
- le personnel ayant chacune des qualifications (langues parlées, compétences techniques, etc...) ;
- le personnel ayant chacune des combinaisons de qualifications.

et feront l'objet de 3 tabulateurs différents.

La comparaison entre la demande et la disponibilité permettra de déterminer si le personnel disponible peut faire les tâches résiduelles. Si l'offre est supérieur à la demande pour un attribut, cet attribut n'est pas contraignant pour la prochaine personne traitée.

Dans le cas d'égalité pour un attribut, il faudra que la prochaine personne traitée exécute une tâche (rotation) demandant cet attribut si cette personne offre cet attribut. Si l'offre est inférieur à la demande pour un attribut, il faut relâcher le

problème ou remettre en cause certains blocs déjà construits et retourner en arrière ("backtracking"). Cette méthode est en cours de développement et fera l'objet du mémoire de maîtrise de Patrick Jeandroz.

4.1.3 Explications sur la construction du réseau

Dans le logiciel PBS, un réseau comprenant toutes les tâches est construit au départ, il est ensuite personnalisé pour chaque employé. Seules les tâches pouvant être effectuées par le pilote durant le mois sont conservées dans son réseau. En effet certaines tâches peuvent être en conflits d'horaire avec certaines tâches pré-assignées (visites médicales, entraînements etc..), elles sont dites alors non compatibles avec ces dernières et éliminées du réseau de l'employé courant.

Dans chaque réseau, les tâches telles que les rotations, périodes de repos et périodes d'entraînements, sont représentées par des arcs (i,j). La figure 4.2 de la p. 40, donne l'allure générale du réseau d'un employé.
Les souhaits exprimés par les employés (en anglais bids) sont représentés sous la forme de coûts sur les arcs et correspondent à la somme des poids des requêtes pondérées, appliquée à la tâche en question. Ce coût reflète la préférence ou non de l'employé courant pour cette tâche. Plus ce coût est faible, plus la tâche est désirée (nous sommes dans un contexte de minimisation). Ainsi, la résolution du problème consiste à trouver un chemin à coût minimum du nœud SC au nœud SK, c'est-à-dire une succession de tâches (rotations, repos etc...) formant un horaire mensuel et respectant les contraintes locales comme dans l'exemple de la figure 4.1.

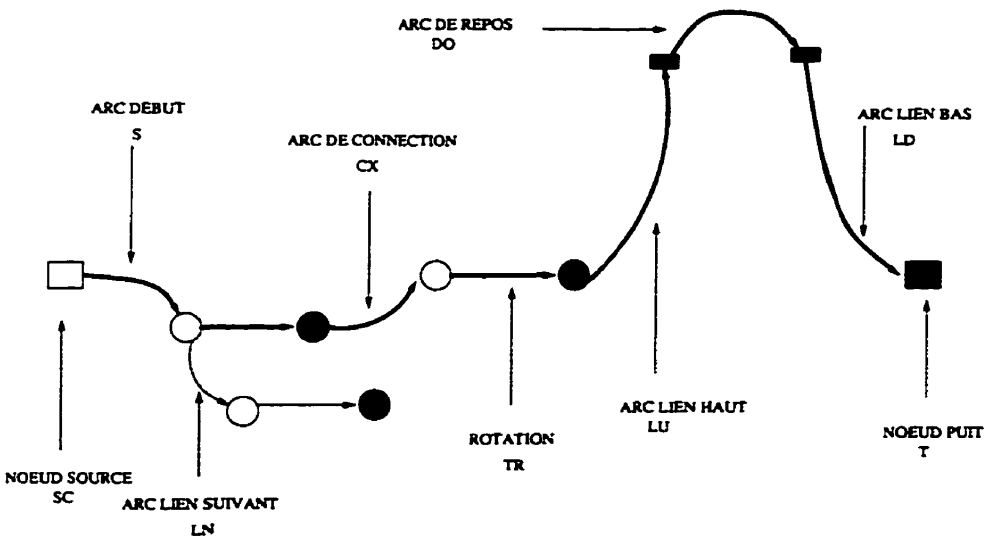


Figure 4.1 – *Exemple d'un horaire dans un sous-problème*

Il y a plusieurs types d'arcs présents dans le réseau de chaque employé : des arcs de rotations, des arcs de connections servant à relier plusieurs rotations entre elles, des arcs de repos, et des arcs de liaisons servant à relier des arcs de repos et de rotations entre eux.

On pourra se reporter à la figure 4.2 pour voir les légendes des différents arcs présents dans le réseau.

1- Ainsi, les arcs de rotations seront nommés TR.

2- Les arcs de liaisons seront nommés LU, LD ou LN et permettront de connecter plusieurs tâches entre elles. Ces arcs existent si les consommations de ressources sont respectées le long du chemin emprunté. Les arcs LD (link down) sont les arcs qui relient les tâches de repos aux rotations, les arcs LU (link up) l'inverse. Les arcs LN (link next) permettent de relier les tâches commençant à la même heure.

3- Les arcs CX/CY permettent d'effectuer des connections entre certaines tâches respectant les consommations de ressources.

4- D'autre arcs nommés DO/FX, ont été crée afin d'autoriser des chemins parmi les tâches fixes (pré-définies) et les tâches de repos.

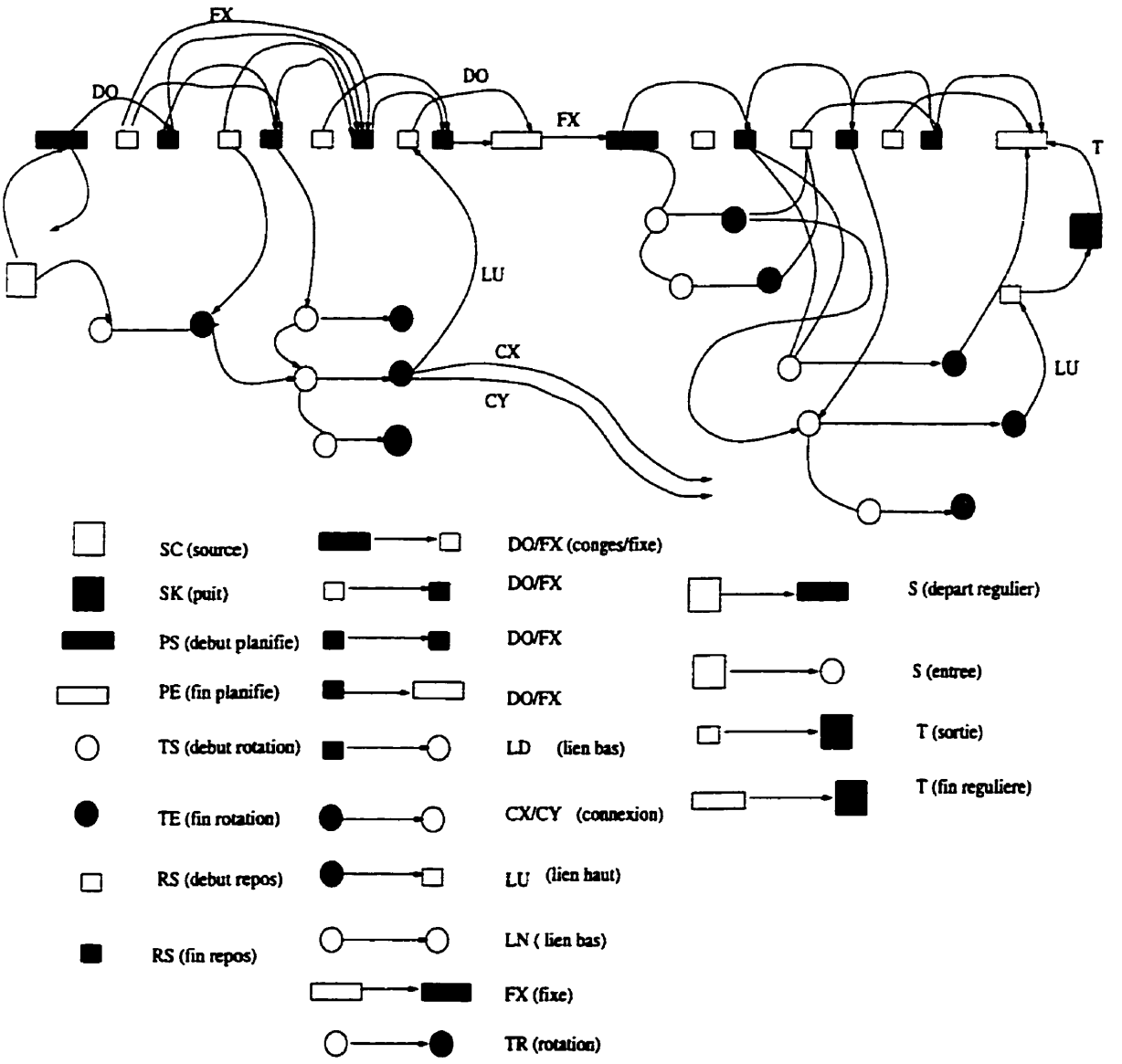


Figure 4.2 – *Réseau d'un sous-problème*

4.2 Techniques de réduction

Ainsi notre travail a consisté à trouver des techniques de réduction du graphe permettant d'améliorer les performances des plus courts chemins sans toutefois détériorer la qualité des solutions. A titre d'exemple pour la résolution de 3000 agents de bords, une ou deux secondes serait acceptable pour chaque employé. Ceci donnerait un temps total d'une ou deux heures de temps de résolution. Ces techniques de réduction sont uniquement utilisables pour le propblème PBS comme nous le verrons plus tard, en raison du concept de "tâches mères" et "tâches filles".

Un réseau est constitué de nœuds et d'arcs, dans notre contexte les tâches sont portées par des arcs. Ainsi le temps de résolution d'un PCC classique dépend du nombre d'arcs. Dans le pire cas la complexité est de $0(n^2)$, n étant le nombre de nœuds présents dans le graphe et m le nombre d'arcs. La complexité peut être améliorée en utilisant des structures de données différentes. La meilleure borne est atteinte pour des longueurs d'arcs non-négatives en $0(m + n\ log\ n)$.
Dans le cadre d'un PCC avec ressources, ce qui est notre cas, le temps de résolution dépend non seulement du nombres d'arcs mais aussi du nombre d'étiquettes qu'il y a par nœuds. A chaque nœud, le nombre d'étiquettes dépend du nombre de chemins jusqu'à ce nœud donc du nombre d'arcs. Ainsi, la complexité de l'algorithme de programmation dynamique dépend du nombre d'étiquettes mais aussi de la structure de données utilisée pour implémenter l'ensemble Q.
Q est l'ensemble des nœuds candidats : c'est-à-dire l'ensemble de nœuds dont les arcs sortants ne garantissent pas $C_i + c_{ij} - C_j \geq 0$, avec C_i, le coût du chemin de la source à i. Ainsi en implémentant Q avec une liste linéaire, pour l'algorithme de

marquage permanent généralisé des étiquettes, appliqué au problème de plus court chemin avec fenêtres de temps, la complexité est de $0(min\{md^2 , D^2\})$ avec m le nombre d'arcs , d la plus grande fenêtre de temps et D le nombre d'étiquettes possibles.

En incluant dans chaque réseau uniquement les tâches (arcs) susceptibles de faire parti du PCC, on diminuera le nombre d'arcs et d'étiquettes et on accéléra le temps de calcul.

4.2.1 Tâches mères et tâches filles

Comme nous avons vu plus haut dans ce rapport, le logiciel PBS crée des horaires mensuels pour du personnel navigant cabine (agents de bord) et technique (pilote) en prenant en considération les préférences de chaque employé.

Pour les problèmes du personnel navigant cabine, presque toutes les rotations doivent être couvertes par plus d'un membre d'équipage. On définit alors une "tâche mère" comme une rotation qui doit être assignée à un équipage en entier. Chaque tâche mère peut être décomposée en "tâches filles", une pour chaque membre d'équipage. Le nombre total de "tâches filles" donne le nombre d'activités à couvrir et peut être utilisé pour estimer le nombre moyen de rotations assignées par horaire. Une "tâche fille" est la copie d'une tâche mère. Dépendemment du nombre d'agents de bords devant couvrir un vol, il y a aura autant de copies (tâches filles) différentes.

A l'origine, le logiciel a été conçu pour résoudre des problèmes concernant le personnel navigant technique. Afin d'adapter le logiciel pour résoudre des problèmes d'agents de bord, on a introduit le concept de "tâches filles". La version de PBS actuel pour agents de bord, considère un problème maître ou il faut couvrir toutes

les tâches filles et un sous-problème ou toutes les tâches filles apparaissent.

4.2.2 Stratégies de réduction

Afin de clarifier les idées, nous allons introduire les notions de tâches identiques et de tâches semblables. On appellera **tâches identiques**, voir figure 4.3, des tâches ayant la même date de départ et la même date d'arrivée. On appellera **tâches semblables**, voir figure 4.4, des tâches "presque pareilles", c'est-à-dire des tâches avec des dates de départ et d'arrivée comprises à l'intérieur d'un certain intervalle de temps. Cet intervalle de temps fera l'objet de différents tests au chapitre 5, et sera de 2, 4, 6, 8, 10 et 12 heures.

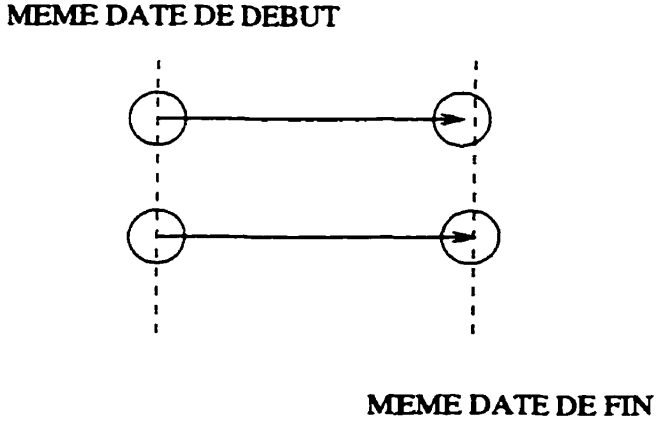


Figure 4.3 – *Tâches identiques*

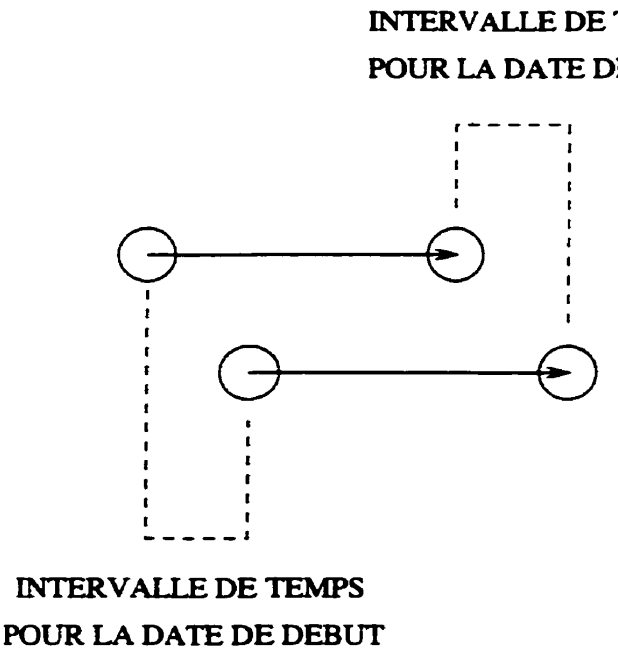


Figure 4.4 – *Tâches semblables*

Remarque 1

La notion de tâches semblables est reliée à l'intervalle de temps car ce sont des tâches qui sont presque pareilles. Lors de la construction de l'horaire d'un employé, les tâches semblables sont des tâches qui commençent et finissent plus au moins au même moment. Ce sont donc des tâches qui sont pratiquement substituables. L'objet de cette réduction est alors d'identifier la tâche qui présente le plus d'intérêt pour l'employé et d'éliminer les autres car le chemin constituant son horaire sera sûrement composé de cette tâche.

Deux stratégies ont été testées afin de diminuer la taille des réseaux et ainsi accélérer le temps de calcul de chaque PCC.

1. Comme il y a autant de tâches filles (copies de tâches mères) dans le réseau de chaque employé, qu'il y a de personnel requis pour couvrir une tâche, la première idée a été d'identifier les tâches identiques (tâches filles) dans chaque réseau et de n'en conserver qu'une copie. Une seule tâche fille représentant sa mère est alors présente dans le réseau de l'employé traité et ce tant que toutes les copies d'une même mère ne sont pas épuisées. La tâche mère (tâche générique) représentant cette fille disparaîtra quand toutes ses filles auront été affectées. Cette réduction est optimale car une tâche identique ne peut être affectée deux fois à l'horaire d'un même employé.
2. La deuxième idée a été d'identifier les tâches semblables c'est à dire les tâches commençant et terminant dans un intervalle de temps variable entre 0 et 12 heures au début et à la fin de la tâche, voir figure 4.5. La tâche ayant le pointage maximal, c'est à dire la tâche ayant le plus de valeur aux yeux de l'employé, sera conservée et les autres éliminées. Cette réduction est heuristique et l'expérimentation nous permettra de déterminer à quel degré d'agrégation il nous sera possible d'aller sans trop dégrader la solution.

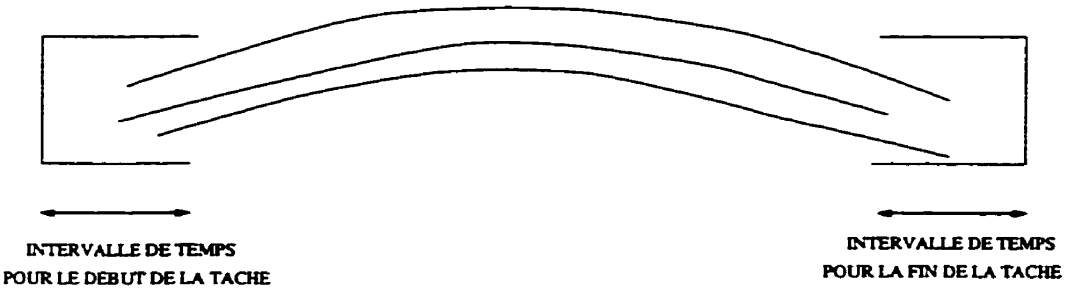


Figure 4.5 – *Réduction de tâches semblables*

3. Une troisième idée pourrait être de séparer l'horizon d'un mois en trois parties. En moyenne, l'horaire d'un employé est composé de trois tâches. Étant donné que les tâches formant son horaire doivent être distinctes, il suffirait de

partitionner les tâches et de ne conserver par période que les plus attrayantes pour l'employé, c'est-à-dire les tâches ayant les pointages les plus importants aux yeux de l'employé courant.

4. Une quatrième idée pourrait être d'identifier les tâches qui se chevauchent et de ne conserver que celles ayant le pointage le plus important pour l'employé courant. En effet un employé ne peut pas couvrir simultanément deux tâches.

La troisième et quatrième idée sont trop heuristiques et n'ont pas été implantées. En effet si l'on découpe l'horizon d'un mois en 3 zones, et que l'on garde un certain pourcentage de bonnes tâches par zones, subsisteront toujours des tâches qui chevauchent deux zones. Le problème se pose alors de savoir dans quelle zone mettre ces tâches? De plus si trois tâches se chevauchent, comme l'illustre la figure suivante, quelle tâche conserver?

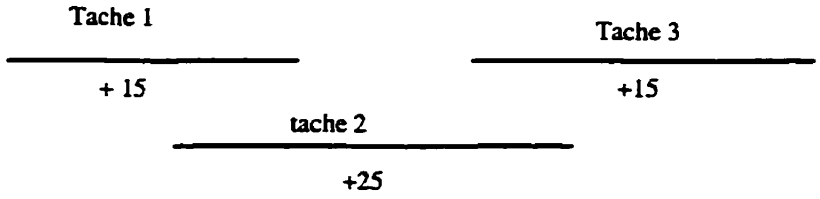


Figure 4.6 – *Exemple de problème*

La première idée serait d'éliminer dans un premier temps la tâche 1 qui chevauche la tâche 2, puis la tâche 3 chevauchant également la tâche 2, pour conserver finalement la tâche 2 avec un coût (+25) (dans un cas de maximisation). Or la solution serait alors non optimale car en conservant la tâche 1 et 3, on aurait pu former un chemin avec un coût supérieur (+30).

Ainsi, nous avons codé en langage C deux fonctions correspondant aux idées 1 et 2.

Remarque 2

La réduction de tâches identiques est une réduction optimale. L'élimination d'arcs identiques dans le réseau d'un même employé ne se fait pas au détriment de la qualité de la solution, car ce ne sont que des copies d'arcs identiques que l'on élimine.

Cependant les réductions de tâches semblables sont heuristiques, c'est-à-dire fournissant une valeur approchée de l'optimalité. En effet, le principe est de regrouper toutes les tâches semblables, c'est-à-dire commençant et terminant dans un intervalle de temps, et de ne conserver que la tâche présentant le plus d'intérêt pour l'employé.

Ainsi, le retrait d'un arc jugé inintéressant pour l'employé peut s'avérer être bon pour la diminution du temps de calcul, car le temps de calcul dépend du nombre d'arcs, mais mauvais pour la qualité de la solution finale comme l'illustre la figure 4.7. Dans le premier ensemble de tâches semblables, on gardera la tâche T2 (+30) car c'est celle qui présente un pointage maximal. Or en conservant T2 et en éliminant T1, les contraintes de précédences nous forcent à conserver T4 (+10). Le pointage des deux tâches est de (+40). Cependant, si l'on avait formé le chemin T1 (+20) et T3 (+40), on aurait eu le score maximal, soit +60.

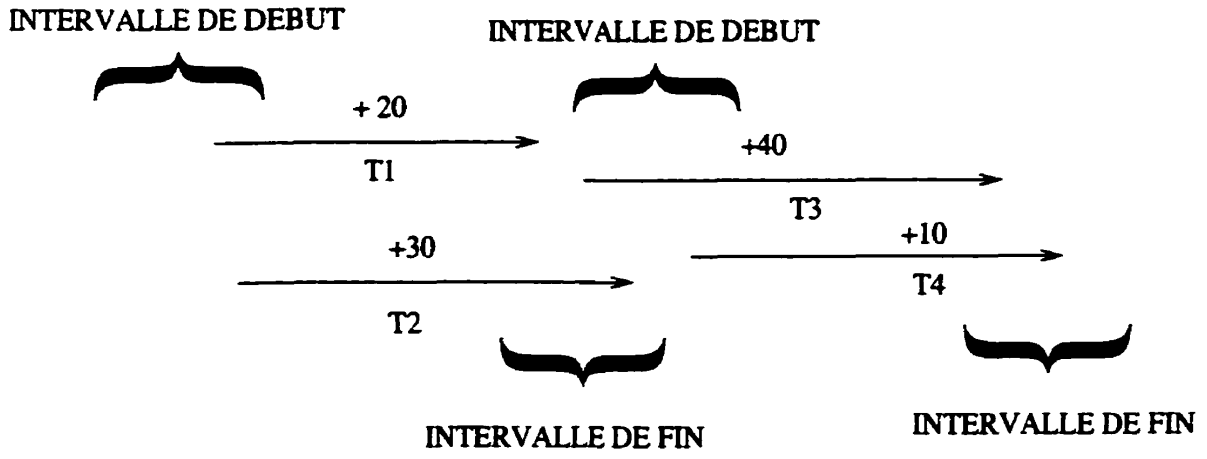


Figure 4.7 – *Exemple de réduction heuristique*

4.2.3 Implémentation

Tâches identiques

L'algorithme utilisé pour identifier les tâches identiques est le suivant :

(1) On initialise le réseau avec toutes les tâches pouvant être effectuées par l'employé courant.

(2) On construit une liste simplement chaînée de tâches considérées dans le réseau.

(3) On trie les tâches suivant :

  (1) Le jour de départ

  (2) Le nom de l'activité

(4) On parcourt la liste triée du début à la fin en cherchant toutes les tâches identiques associées au même vol et on en conserve qu'une seule copie.

  (1) Tant que la tâche courante est identique à la première de la liste, on constitue une sous liste sinon on s'arrête.

(1) Si la sous liste contient une seule tâche , on la conserve.

(2) Sinon, on conserve la première de la sous liste et on élimine les autres.

(2) On continue de progresser dans la liste triée à partir de la dernière tâche de la sous liste précédente.

Tâches semblables

L'algorithme utilisé pour identifier les tâches semblables est le suivant :

(1) On initialise le réseau avec toutes les tâches pouvant être effectuées par l'employé courant.

(2) On construit une liste simplement chaînée de tâches considérées dans le réseau.

(3) On trie les tâches suivant :

(1) La date de départ

(2) L'heure de départ

(4) On crée un seau (en anglais bucket) pour chaque intervalle de fin de tâches possible en divisant le mois en intervalles égaux de 2, 4, 6, 8, 10 ou 12 heures.

(5) On parcourt la liste des tâches ayant le même intervalle d'heure de départ et on place chaque tâche dans une liste, associée au seau approprié. Le numéro du seau se calcule avec une fonction linéaire de l'heure de fin de cette tâche.

(6) Pour chaque seau non vide, on ne conserve que la tâche ayant le pointage le plus intéressant pour l'employé courant.

(7) On passe à la sous liste des tâches ayant l'heure de départ suivant. On réinitialise et on va à (4).

4.2.4 Les retombés

L'application de ces idées peut entraîner plusieurs retombés à différents niveaux.

Dans la version *PBS 200*, les tâches filles sont utilisées dans le problème maître et les sous problèmes. Les contraintes de recouvrement du problème maître ont un second membre égale à 1, $\sum_{r \in R} a_{pr} X_r = 1, \ \forall p \in P$. Ainsi lorsqu'on affecte une tâche à un senior, cette tâche n'est plus disponible pour les employés suivant et peut être enlevée du problème maître.

Cette modélisation des contraintes a pour conséquence de grossir le problème, d'une part au niveau du problème maître, car le nombre de contraintes est égal au nombre de tâches filles et d'autre part au niveau des sous problèmes car les tâches sont portées par les arcs. Le nombres d'arcs dans les réseaux est égal au nombre de tâches.

En éliminant les copies des tâches filles dans les réseaux de chaque employé, on réduit la taille des sous problèmes. Les réseaux contiennent en moyenne 5 fois moins d'arcs et cela en facilite leurs résolutions. Le temps de résolution se trouve alors réduit par un facteur 5 et plus en fonction du nombre d'étiquettes conservées par nœuds.

Au niveau du problème maître, si de nouveaux branchements pouvaient être développés pour respecter des contraintes du type, $\sum_{r \in R} a_{pr} X_r \geq 5, \ \forall p \in P$, par exemple dans le cas ou un vol devrait être couvert par 5 employés, il serait alors possible de travailler directement avec les tâches mères au niveau du problème maître.

Cela permettrait là encore de réduire le temps de calcul par environ un facteur de 25, car il y aurait alors 5 fois moins de contraintes dans la base et le temps de calcul est plutôt quadratiques en fonction de la taille de la base. Ainsi la réduction pourrait être directement exploitable au niveau du problème maître, si les contraintes de recouvrement pouvaient porter sur les tâches mères. Actuellement de telles branchements ont déjà été développés pour certaines applications, comme dans le domaine du transport ferrovière, voir Ziarati, K. (1997).

Cette réduction des tâches au niveau des sous problèmes est comme on l'a dit précédemment directement utilisable. En effet, dans la nouvelle version de *PBS 5000* actuellement en développement, GENCOL n'est pas résolu donc pas de problème maître. On résout uniquement des sous-problèmes séquentiellement du plus senior au plus junior sans se préoccuper des contraintes globales. Chaque sous-problème nous donne alors l'horaire optimal pour l'employé traité. Ainsi, la réduction des réseaux est directement applicable pour diminuer le temps de calcul.

Pour *PBS 200* et *PBS 500*, le problème est résolu avec GENCOL. Le branchement est facilité en travaillant avec des tâches filles car cela revient à imposer un arc avec un flot le traversant égal à 1 ou 0. Or cette façon de faire n'est pas valable pour les tâches mères car si à un nœud de branchement 2 tâches sont fixées comme étant effectuées successivement, elles le seront pour tous les employés tant que sa mère sera présente dans le problème, en d'autres termes tant que toutes les copies d'une même mère ne seront pas épuisées. Une autre idée de branchement pourrait être d'affecter une tâche à un employé, en enlevant toutes les autres tâches à effectuer en même temps dans le réseau de l'employé ou à lui interdire cette tâche en l'enlevant du réseau de l'employé.

Lors de l'expérimentation, les tests ont été réalisés sur des problèmes de petites tailles, de l'ordre de 600 employés pour les plus gros. Il serait intéressant d'extrapoler l'ampleur de la réduction pour de plus gros problèmes. On peut espérer que la réduction de tâches semblables pour 5000 employés peut être beaucoup plus importante que pour les petits problèmes. En effet, les rotations ont des durées assez courtes. De plus, les rotations ont des heures de départ et d'arrivée souvent semblables. Par exemple, les vols intercontinentaux partent en fin de soirée de l'Amérique et arrivent en Europe le matin. Ainsi, quand on augmente le nombre d'employés et donc le nombre de rotations à couvrir, on augmente beaucoup plus le nombre de tâches par seau (en anglais bucket) que le nombre de seaux non vides. Cette étude sera exposée dans les sections suivantes après avoir présenté les différents résultats obtenus pour des problèmes plus petits.

4.3 Étude probabiliste

Au chapitre suivant, nous présenterons les résultats obtenus en appliquant les idées de réduction présentées précédemment, à des problèmes de 100, 300 et 600 employés.

On peut déjà se demander quelle pourrait être la taille d'un réseau pour un problème de grande taille, après avoir appliqué les idées de réduction. Nous présenterons dans cette section une méthodologie pour faire cette étude.

L'idée qui ressort des techniques de réduction, est d'identifier les tâches identiques ou semblables, de les regrouper et de n'en laisser qu'une copie.
On pourrait modéliser ce regroupement en utilisant un ensemble de cellules ou de classes dans lesquelles seraient placées des données (les rotations à effectuer durant notre horizon de temps). Le nombre de classes dépendra du type de regroupement que l'on choisira.

A partir des observations sur les jeux de données que nous possédons, on voudrait alors pouvoir estimer en se basant sur des études probabilistes, le nombre de cellules (ou classes) qui resteraient vides, après avoir mis toutes les données dans ces cellules et ce pour des problèmes de grande taille. On pourra alors estimer quel est le nombre de classes où l'on retrouve une tâche et plus. Ce nombre de classes non vides, déterminera alors le nombre de tâches restantes au début de la résolution du problème et nous renseignera sur la taille des problèmes à résoudre après réduction.

Pour ce faire, l'idée est d'estimer à partir des rotations que nous avons pour les deux mois, la probabilité P_i qu'une tâche soit présente dans une classe i.
Or nous n'avons que peu de données pour estimer les P_i, les données de 2 mois pour un petit problème de 300 employés et ceci introduit des erreurs d'estimation. En effet si on estime qu'une classe a une probabilité nulle d'occupation avec nos observations sur un petit jeu de données, cette classe restera vide en extrapolant à de plus gros problèmes, ce qui n'est pas forcément vrai.

L'idée est alors de regrouper plusieurs classes, de les agréger pour avoir une meilleur estimation des P_i. On pourrait par exemple, faire la moyenne $\bar{P}$ de la probabilité P_i de plusieurs classes, l'erreur sur $\bar{P}$ sera alors moins grande que sur les P_i pris individuellement. Ces agrégations de données feront l'objet des deux approches qui seront présentées dans le calcul analytique de la taille du réseau au chapitre 6. Par soucis de clarté, nous allons présenter dès maintenant les modèles probabilistes qui seront utilisés.

Soit la classe i, on peut considérer que le nombre de tâches (rotations) présent dans une classe particulière i, est une variable aléatoire de distribution bînomiale. En effet, la loi bînomiale est utilisée pour trouver la probabilité qu'un événement se réalise k fois au cours de n essais de la même expérience lorsque :

- l'expérience ne peut aboutir qu'à 2 résultats mutuels exclusifs, (dans notre cas présence ou non d'une rotation).
- la probabilité de réalisation de l'événement reste constante lors de chaque essai,
- les n essais sont indépendants.

On va dans un premier temps déterminer la probabilité P_i qu'une classe soit occupée. A partir de cette probabilité, on utilisera la loi de distribution bînomiale afin de déterminer la probabilité que cette même classe reste vide au cours de n essais. Dans notre cas, n correspondra au nombre total de rotations à placer dans les classes.

Si une classe est vide lors de nos observations, elle aura une probabilité d'occupation nulle, soit $P_i = 0$. Cette classe restera vide après n essais et aura donc une bînomiale nulle. Si la probabilité d'occupation d'une classe $P_i \neq 0$, il se peut qu'après n essais la bînomiale soit nulle ou pas.

Ainsi, l'espérance du nombre de classe non vides, qui correspondra au nombre de tâches présentes dans le réseau après réduction, sera alors égale à la différence entre le nombre totale de classes et l'espérance du nombre de classes vides après n essais.
L'espérance du nombre de classes vides sera obtenue comme la somme des probabilités qu'une classe reste vide après n essais et du nombre de classes vides (c'est-à-dire avec une probabilité d'occupation d'une classe P_i nulle).

Soit P_i : la probabilité qu'une rotation choisit au hasard, soit dans la classe i
(cette probabilité sera déterminé à partir de la moyenne de nos observations sur deux mois)

Soit n : le nombre de rotations du jeu de données considéré

Soit X_i : le nombre de rotations présentes dans la classe i

Définissons la bînomiale $\mathcal{B}$(n, k) comme étant la probabilité $P(X_i = k)$ qu'un événement se réalise k fois lors de n essais, tel que :
$P(X_i = k) = \mathcal{B}(n, k) = \binom{n}{k} p^k (1-p)^{n-k}$ avec p la probabilité de l'événement et n le nombre d'essais.

Pour évaluer P_i, nous verrons au chapitre 6, qu'étant donné que la fréquence des vols se ressemble chaque semaine, on peut évaluer P_i la probabilité théorique d'occupation d'une classe à partir de la moyenne sur 2 mois des fréquences d'occupations observées $\hat{P}_i$. En définissant un mois comme 4.3 semaines de 7 jours ($\simeq$ 30 jours divisés par 7), on aura alors, $P_i = \frac{1}{4.3} \hat{P}_i$.

. A partir des probabilités d'occupation de chaque classe, on pourra déterminer la probabilité qu'une classe reste vide après n essais, 1 essai correspondant à placer une tâche dans une classe et ce pour les n tâches mères.

Pour évaluer la probabilité que chaque classe reste vide après n essais, il suffit de calculer la bînomiale $\mathcal{B}_i$ pour chacune des classes tel que l'événement (la présence de rotation) se réalise 0 fois lors de n essais. :
soit $P(X_i = 0) = \mathcal{B}_i(n, k) = \mathcal{B}(n, 0) = \binom{n}{0} P_i^0 (1-P_i)^{n-0} = (1 - P_i)^n$ avec P_i la probabilité qu'une case soit occupée.

Cette méthodologie sera utilisée au chapitre 6 avec deux méthodes d'estimation des P_i.

Chapitre 5

Expérimentation

Introduction

Les idées de réduction précédentes ont été implantées et testées afin de voir leurs impacts sur la diminution de la taille des réseaux pour chaque employé. Ces tests ont été entrepris sur plusieurs problèmes tests d'Air Transat de tailles différentes. Nous continuerons d'appeler dans la suite de ce rapport tâches identiques, des tâches ayant une même date de départ et une même date de fin. Nous appelerons également tâches semblables, des tâches commençant et terminant dans un intervalle de temps pouvant être de 2, 4, 6, 8, 10 ou 12 heures.

A la section 5.1, nous présenterons les résultats des tests pour le jeu d'une centaine d'employés.
A la section 5.2, nous présenterons les résultats des tests pour un jeu de 300 employés.
A la section 5.3, nous présenterons les résultats des tests pour un jeu de 600 employés.

5.1 Problème test 1

Le premier test a été effectué sur un jeu de données contenant une centaine d'employés. L'objectif visé est d'observer l'ampleur de la réduction et la dégradation

éventuelle de la qualité de la solution.

Réduction de tâches identiques

Tableau 5.1 – Tableau exprimant des % suite à une réduction de tâches identiques

% moyen des tâches enlevées	61 %
% moyen des arcs enlevés	10.3 %

Parmi les rotations présentes dans le réseau de chaque employé, la réduction a éliminé en moyenne 61 % de ces rotations. Sur l'ensemble des arcs du réseau, la réduction a éliminé quant à elle en moyenne 10.3 % des rotations.

Afin de comprendre plus précisement les raisons d'un si faible pourcentage des arcs enlevés, nous avons établi la proportion moyenne du nombre d'arcs par type d'arcs avant réduction des tâches identiques :

1. les arcs de départ (start) : 0,1 %
2. les arcs de fin (end) : 4,3 %
3. les arcs de formation (training) : 0,0 %
4. les arcs de rotations (trip) : 16,4 %
5. les arcs de repos (day off) : 6,3 %
6. les arcs de liens vers le bas (link down) : 2,7 %
7. les arcs de liens vers le haut (link up) : 29,9 %
8. les arcs de liens suivant (link next) : 27,5 %
9. les arcs de connections entre les rotations (connexion) : 12,6 %

10. les arcs d'activités fixes (fixed activity) : 0,2 %
11. les arcs pour les activités de réserve (reserve activity) : 0,0 %

Après réduction des tâches identiques, la proportion moyenne des arcs présents dans le réseau est la suivante :

1. les arcs de départ (start) : 0,1 %
2. les arcs de fin (end) : 4,8 %
3. les arcs de formation (training) : 0,0 %
4. les arcs de rotations (trip) : 6,6 %
5. les arcs de repos (day off) : 7,1 %
6. les arcs de liens vers le bas (link down) : 3 %
7. les arcs de liens vers le haut (link up) : 33,4 %
8. les arcs de liens suivant (link next) : 30,7 %
9. les arcs de connections entre les rotations (connexion) : 14,1 %
10. les arcs d'activités fixes (fixed activity) : 0,2 %
11. les arcs pour les activités de réserve (reserve activity) : 0,0 %

Le pourcentage totale d'élimination d'arcs est assez faible relativement à la réduction du nombre de tâches car la réduction porte seulement sur les arcs de rotations, représentant en moyenne que 16.4 % de l'ensemble totale des arcs et n'affecte pas les autres arcs du réseau (arcs de connections, de repos, etc). La raison pour laquelle les arcs de liens et de connections n'ont pas été éliminés en même proportion que les arcs de rotations, est parce que les idées de réduction du réseau ont

été implémentées après la construction du réseau. Ainsi le réseau est initialement construit avec les copies de chaque mère et avec les liens qui les relient. La réduction élimine dans un second temps les copies de tâches identiques mais en laissant les arcs de liens et de connections alors en grand nombre dans le réseau. Ceci explique la faible diminution totales des arcs lors de la résolution du problème de construction des horaires.

Nous avons alors effectués des nouveaux tests avec des fichiers d'entrées ne comportant que des tâches mères. Lors de la construction du réseau les liens reliant les tâches identiques ne seront alors pas crées et nous pourrons alors avoir une meilleure idée de la portée d'une telle réduction. Les résultats obtenus avec le nouveau fichier dépourvu de tâches filles seront présentés à la fin de cette section.

Les fonctions de réduction n'ont pas été implémentées en amont de la construction du réseau pour les raisons suivantes. Les fonctions de réduction ont été codées de manière à faire appel en entrée à une liste d'arcs provenant de la construction préliminaire du réseau. D'autre part la réduction des tâches semblables avant la construction du réseau serait assez difficile à implémenter. En effet, si des regroupements et des réductions de tâches semblables sont faits à priori c'est-à-dire avant la construction du réseau, ces tâches n'apparaitront plus dans les réseaux des employés et devront alors être réintroduites dynamiquement, ce qui n'est pas trivial.

Remarque 1

Par soucis de présentation, les courbes présentées dans ce chapitre, seront des courbes continues bien que les employés soient discrets. Il est ainsi évident qu'il n'existe pas d'employé "0", la courbe devrait commencer pour le premier employé.



Figure 5.1 – *Nombres de tâches restantes avant et après réduction des tâches identiques*

Comme on peut le voir sur la figure 5.1, représentant le nombre de tâches restantes avant et après réduction en fonction de l'employé traité, l'élimination des tâches identiques a été importante. L'irrégularité de la courbe est causée par les tâches pré-assignées que l'employé doit effectuer et qui doivent nécessairement faire parties de son réseau. Ces tâches peuvent dans certains cas être en conflit avec certaines des tâches à affecter durant l'horizon de temps. La réduction entraîne une diminution du temps de calcul pour chaque sous-problème comme l'illustre la figure 5.2.



Figure 5.2 – *Temps de calcul des PCC avant et après réduction des tâches identiques*

D'autre part, la qualité de résolution du PCC avant et après réduction reste identique. En effet lors de la réduction du réseau, seules les tâches identiques ont été éliminées.

Des fonctions calculant le temps passé dans la résolution des sous-problèmes, nous ont permis d'évaluer le gain en temps obtenu par l'élimination des tâches identiques. Cela a permis de diminuer en moyenne le temps de calcul de **29 %**. Ceci s'explique par une diminution moyenne proportionnelle d'appels à la fonction de prolongation à chaque nœud de **30.1 %** et aux nombres moyens d'étiquettes restantes avant dominance de **23 %**, poucentages que nous avons pu déterminer en calculant le nombre d'étiquettes générées avant et après réduction des tâches identiques. En effet, le temps passé dans le plus court chemin est proportionnel au nombre d'arcs et au nombre d'étiquettes par nœud. Ainsi, en réduisant le nombre d'arcs, on diminue le nombre d'étiquettes qui vont être prolongées par la fonction de prolongation et ainsi on réduit le temps de calcul.

Après prolongation des étiquettes, on s'assure que l'ensemble des étiquettes soient réalisables par rapport aux fenêtres sur les ressources à chaque nœud. Par exemple, dans le cas de la ressource temps, on ne peut partir avant d'arriver. On appelle cette opération le filtrage. Certaines étiquettes sont alors éliminées si elles ne respectent pas les intervalles de ressource. L'effort de calcul suivant vient de la dominance c'est-à-dire le nombre d'états présents à considérer dans la programmation dynamique. Cet effort a diminué en moyenne de **23 %**. Dans certains problèmes afin de diminuer le temps de calcul, il devient intéressant de ne pas dominer sur toutes les ressources. La solution est alors réalisable mais heuristique.

Afin de comprendre l'impact sur le temps de calcul sauvé après élimination de 10 % des arcs, nous avons calculé le nombre moyen d'étiquettes présentes à chaque nœud et qui sont prolongées avant et après réduction des tâches identiques . En effet ce nombre peut s'évaluer facilement comme suit.
Le nombre d'étiquettes par nœud avant réduction est égale au nombre d'étiquettes générées après dominance divisé par le nombre d'arcs avant réduction. De même, on peut calculer le nombre d'étiquettes par nœud après réduction. Il vient après calculs que le nombre d'étiquettes par nœud avant réduction est égale à 8.3 avec un écart type de 4.2 et le nombre d'étiquettes par nœud après réduction est égale à 7 avec un écart type de 3.5. La diminution d'étiquettes à chaque nœud est responsable de la diminution du temps de calcul.

Remarque 2

Afin d'expliquer le concept de génération d'étiquettes et de dominance prenons l'exemple de la figure 5.3. Trois chemins vont du nœuds s au nœud t en passant par le nœud i. Le coût d'emprunt de l' arc (s,i) et sa consommation en temps sont connus. Compte tenu de la valeur de ces variables de ressources en s et de la consommation (coût et temps) de ces deux ressources sur l'arc (s,i), on peut connaître la valeur des variables de ressources en i .

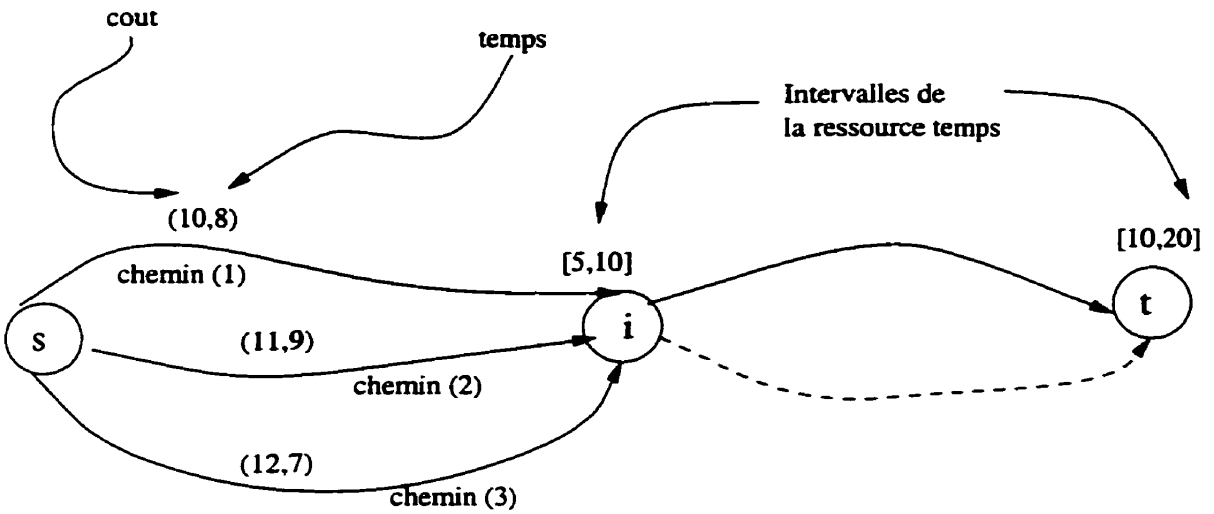


Figure 5.3 – *Exemple sur le principe de dominance*

Le chemin (2) est éliminé car il coûte plus cher et il arrive plus tard que le chemin (1). On dit que le chemin (2) est dominé par le chemin (1). Les chemins (1) et (3) sont tous deux réalisables, car la valeur de la ressource temps se trouve dans l'intervalle au noeud i. On a le choix de garder la valeur des étiquettes de ces ressources en i pour ces deux chemins ou "dominer" sur une ressource, le temps ou le coût de l'arc afin de n'en conserver qu'une. Bien sur, dans des réseaux plus importants, il sera parfois nécessaire de dominer sur plusieurs ressources afin de ne

pas conserver trop d'étiquettes tout au long du calcul du plus court chemin du nœud source au nœud puits. Cependant la solution obtenue est heuristique car le principe de dominance diminue le réseau global, quand une étiquette est enlevée, ce sont tous les chemins passant par ce nœud qui le sont aussi. Ainsi un compromis est à trouver entre l'échantillonnage et la qualité de la solution.

Remarque 3

Après avoir modifié les fichiers d'entrées en ne laissant que les tâches mères, nous avons effectué de nouveaux tests.

Les moyennes ont été calculées pour les 25 premiers employés, le problème étant irréalisable pour les employés suivants. La moyenne du pourcentage des tâches enlevées pour 25 employés est plus importante que sur l'ensemble des employés car la réduction est plus importante pour les premiers employés, en effet un plus grand nombre de tâches sont encore présentes pour les plus seniors car celles-ci n'ont pas encore été attribuées.

Tableau 5.2 – Tableau exprimant des % suite à une réduction de tâches identiques

% moyen de l'ensemble des arcs enlevés	64,1 %
% moyen des arcs de rotations enlevés	82,5 %
% moyen des arcs de liens vers le haut enlevés	68,3 %
% moyen des arcs de liens suivants	74,7 %
% moyen des arcs de connections	67,6 %

En terme de temps de calcul, les gains sont supérieurs et présagent des améliorations

notables pour les plus gros réseaux.

Tableau 5.3 – Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques

% moyen de la diminution du temps de calcul	73,8 %

Réduction de tâches semblables

Des tests équivalents ont été réalisés avec une réduction sur des tâches semblables, c'est-à-dire commençant et terminant dans un intervalle de temps donné. Les résultats obtenus avec des intervalles de 2, 4, 6, 8, 10 et 12 heures pour le même problème test de 108 employés sont présentés dans le tableau suivant.

Tableau 5.4 – Tableau exprimant le % moyen de tâches semblables enlevées

2 hrs	**4 hrs**	**6 hrs**	**8 hrs**	**10 hrs**	**12 hrs**
72.1 %	72.1 %	72.2 %	73.8 %	73.8 %	73.8 %

D'après le tableau 5.4, on peut observer que le pourcentage de réduction reste constant lors de l'élimination des tâches semblables à partir d'un intervalle de 8 heures et plus.

Un résumé des pourcentage d'arcs enlevées suite à la réduction des tâches est donné dans le tableau 5.5 :

Tableau 5.5 – Tableau exprimant le % moyen d'arcs enlevés

2 hrs	4 hrs	6 hrs	8 hrs	10 hrs	12 hrs
11.9 %	11.9 %	12 %	12.2 %	12.2 %	12.2 %

Comme nous l'avons expliqué plus haut, les fonctions de réduction ont été implémentées après la construction du réseau. Ainsi, l'élimination des tâches semblables pourrait avoir un impact plus important en terme de réduction d'arcs et donc de temps de calcul, si elle avait été implémentée avant la construction du réseau pour chaque employé. Mais cela nécessite une réécriture complète du code et sort du cadre de cette maîtrise.

La figure 5.4 représente les courbes des tâches restantes en fonction des employés traités, obtenues après élimination des tâches semblables.

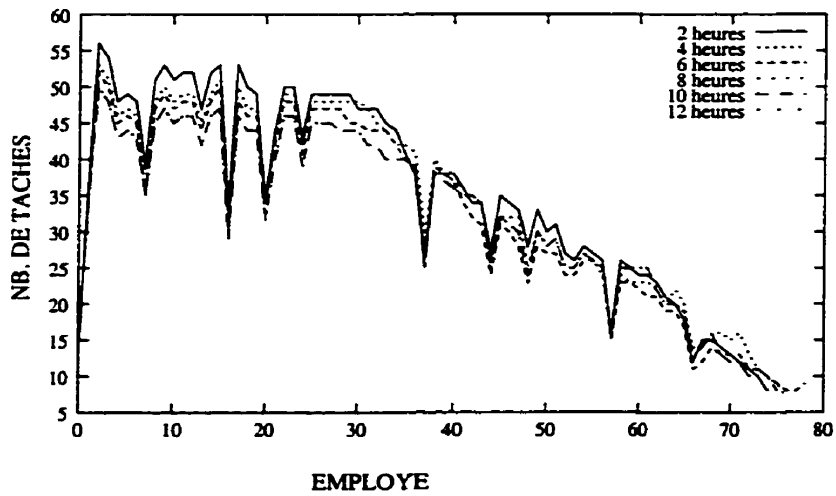


Figure 5.4 – *Nombres de tâches restantes après réduction de tâches semblables*

L'élimination de tâches semblables dans un intervalle diminue sensiblement le temps de calcul des sous problèmes comme l'indique le tableau 5.6.

Tableau 5.6 – Tableau du % moyen de la diminution du temps de calcul après réduction de tâches semblables

2 hrs	4 hrs	6 hrs	8 hrs	10 hrs	12 hrs
40 %	40 %	43.7 %	44.9 %	44.9 %	44.9 %

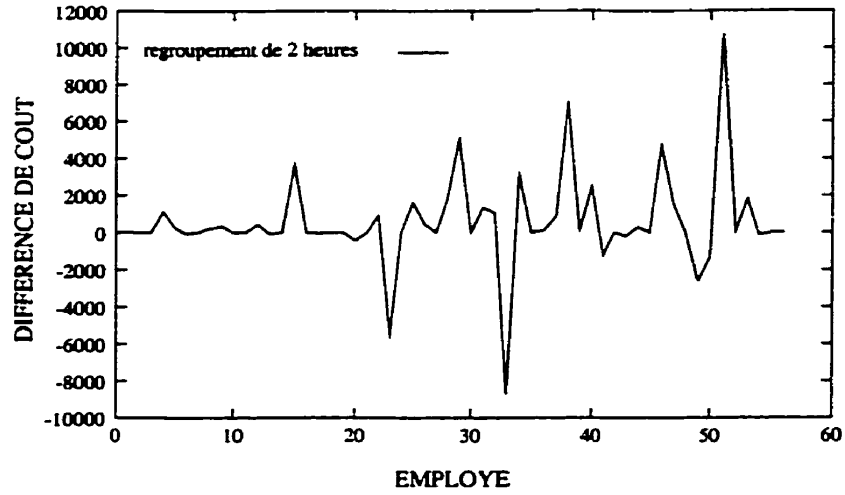


Figure 5.5 – *Différence de qualité de la solution après un regroupement de 2 heures*

La figure 5.5 nous renseigne sur la dégradation de la solution après un regroupement de tâches dans un intervalle de 2 heures. La dégradation est d'autant plus importante que l'intervalle de temps sur lequel s'effectue le regroupement est plus large. On peut voir aussi sur cette figure que la modification de la solution augmente au fur et à mesure de la construction des horaires, ceci peut s'expliquer par les copies des tâches qui diminuent et qui ne sont plus disponibles pour les plus juniors. Les choix différents effectués pour les plus seniors changent les tâches résiduelles pour les plus juniors. Ces tâches peuvent être des tâches encore non affectées ou des tâches qui ont été supprimées des réseaux des plus seniors et qui dans certains cas améliorent ou dégradent la solution des plus juniors. Ainsi les modifications pour les juniors dépendent du cumuls des approximations faits pour les plus seniors. On

peut remarquer d'autre part qu'il y a quelques détériorations pour certains seniors car la réduction heuristique des tâches semblables a dans certains cas éliminée des tâches qui auraient pu être bonnes pour l'employé traité. La réduction de tâches semblables pénalise les moins seniors.

5.2 Problème test 2

Les mêmes genres de tests ont été effectués sur un problème test d'Air Transat de taille plus importante comportant un peu plus de 300 employés. L'objectif visé est là encore d'observer l'ampleur de la réduction et la dégradation éventuelle de la qualité de la solution.

Réduction de tâches identiques

Tableau 5.7 – Tableau exprimant des % suite à une réduction de tâches identiques

% moyen des tâches enlevées	57.2 %
% moyen des arcs enlevés	9 %

L'élimination des tâches identiques a porté en moyenne sur 57.2 % des tâches, ce qui a mené à la réduction moyenne sur l'ensemble des sous problèmes de 9 % des arcs .

La figure 5.6 représente le nombre de tâches restantes avant et après réduction en fonction de l'employé traité.

L'élimination des tâches identiques a réduit sensiblement le réseau et a entraîné une diminution du temps de calcul pour chaque sous-problème comme l'illustre la figure 5.7. En n'enlevant que les tâches identiques, chaque sous problème est résolu en une à deux secondes.

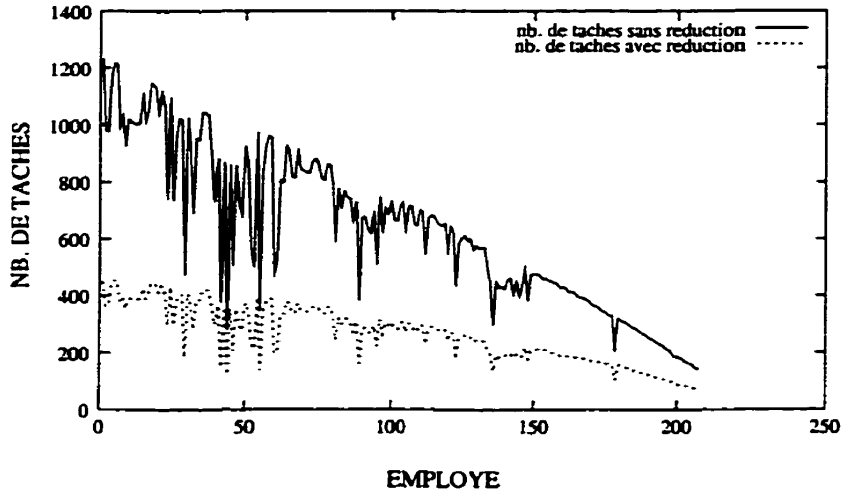


Figure 5.6 – *Nombres de tâches restantes avant et après réduction des tâches identiques*

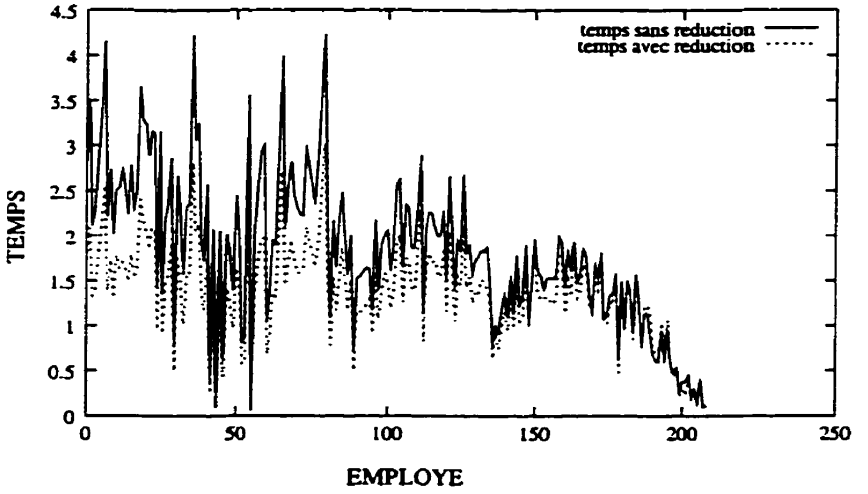


Figure 5.7 – *Temps de calcul des PCC avant et après réduction des tâches identiques*

La qualité de la solution quant à elle, n'a pas été dégradée pour les mêmes raisons expliquées précédemment.

Tableau 5.8 – Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques

Moyenne de la diminution du temps de calcul	22.4 %

Afin d'expliquer le pourcentage de la diminution du temps de calcul, des statistiques sur le nombre d'appels à la fonction de prolongation et le nombre d'étiquettes générées ont été calculées.

Tableau 5.9 – Tableau de % de la diminution de prolongations et d'étiquettes

% moyen de la diminution d'appel à la fct de prolongation	29.3 %
% moyen de la diminution d'étiquettes avant dominance	28.1 %
% moyen de la diminution d'étiquettes après dominance	15.3 %

Remarque

Comme pour le problème de 108 employés, nous avons effectués de nouveaux tests après avoir modifié les fichiers d'entrée en ne laissant que les tâches mères.

Les moyennes ont été calculées pour les 75 premiers employés, le problème étant irréalisable pour les employés suivants. La moyenne du pourcentage de tâches enlevées pour 75 employés est plus importante que sur l'ensemble des employés car la réduction est plus importante pour les premiers employés, en effet un plus grand nombre de tâches sont encore présentes pour les plus seniors car celles-ci n'ont pas encore été attribuées.

Tableau 5.10 – Tableau exprimant des % suite à une réduction de tâches identiques

% moyen de l'ensemble des arcs enlevés	63,4 %
% moyen des arcs de rotations enlevés	78,4 %
% moyen des arcs de liens vers le haut enlevés	64,1 %
% moyen des arcs de liens suivants	70,3 %
% moyen des arcs de connections	63,7 %

En terme de temps de calcul, les gains sont supérieurs et présagent des améliorations notables pour les plus gros réseaux.

Tableau 5.11 – Tableau exprimant le % de la diminution de temps suite à une réduction de tâches identiques

% moyen de la diminution du temps de calcul	71,2 %

Réduction de tâches semblables

Les pourcentages de la moyenne respective des tâches enlevées après réduction des tâches semblables commençant et terminant dans un intervalle de 2, 4, 6, 8, 10, et 12 heures pour un jeu de données de 313 employés sont :

Tableau 5.12 – Tableau exprimant le % moyen de tâches semblables enlevées

2 hrs	**4 hrs**	**6 hrs**	**8 hrs**	**10 hrs**	**12 hrs**
78.9 %	81.9 %	84.2 %	84.6 %	85.7 %	87.2 %

Tableau 5.13 – Tableau exprimant le % moyen d'arcs enlevés

2 hrs	4 hrs	6 hrs	8 hrs	10 hrs	12 hrs
12 %	12.7 %	12.9 %	13 %	13.1 %	13.7 %

La figure 5.8 représente les courbes des tâches restantes obtenues après réduction des tâches semblables, obtenues pour des tâches commençant et terminant dans un intervalle de 2, 4, 6, 8, 10 et 12 heures. On peut observer que le logiciel PBS n'est pas en mesure de trouver un horaire faisable pour chaque employé car l'algorithme est glouton. Il n'y a pas de contraintes globales qui s'assurent que les employés résiduels peuvent faire les tâches résiduelles.

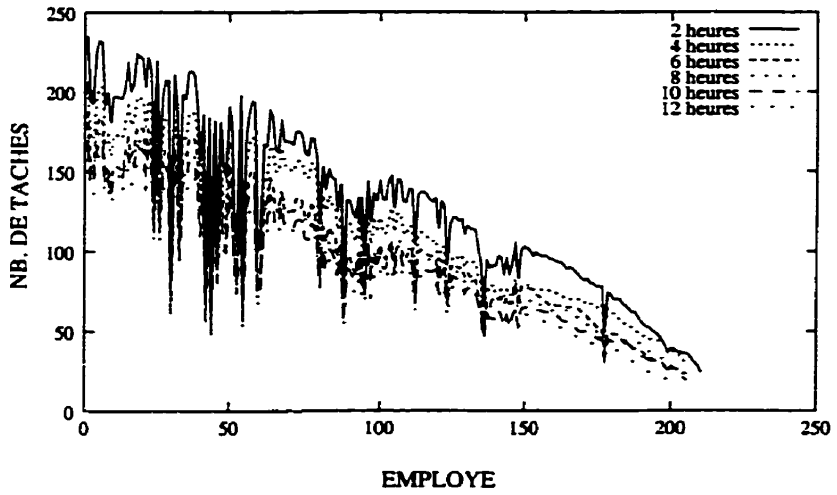


Figure 5.8 – *Courbes du nombres de tâches restantes après réduction de tâches semblables*

Là encore le temps de résolution des sous problèmes diminue en fonction du regroupement utilisé. Cependant il pourrait encore diminuer si l'implémentation des fonctions de réduction avaient été faites avant la construction du réseau.

Tableau 5.14 – Tableau du % moyen de la diminution du temps de calcul après réduction de tâches semblables

2 hrs	4 hrs	6 hrs	8 hrs	10 hrs	12 hrs
36.5 %	37.1 %	46.5 %	46.3 %	45.1 %	50.1 %

La figure 5.9 nous renseigne sur la dégradation de la solution après élimination des tâches semblables commençant et terminant dans un intervalle de 2 heures. Après élimination des tâches semblables commençant et terminant dans un intervalle de 12 heures, la dégradation est encore plus évidente et commence dès la construction d'horaires pour les premiers employés comme le montre la figure 5.10.

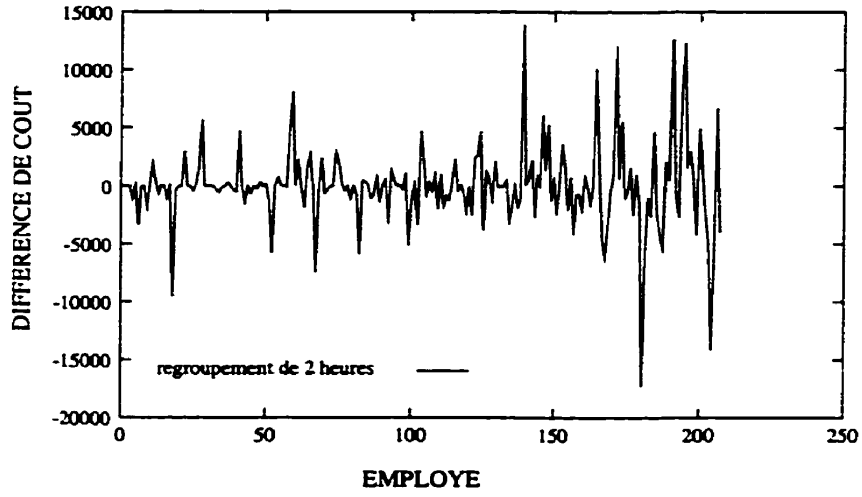


Figure 5.9 – *Courbes donnant la différence de qualité (coût) de la solution avec un regroupement de 2 heures*



Figure 5.10 – *Courbes donnant la différence de qualité (coût) de la solution avec un regroupement de 12 heures*

5.3 Problème test 3

Afin d'obtenir des jeux de données de taille plus importante, nous avons réalisé une fusion des données pour deux problèmes tests de 313 employés pour les mois de juin et juillet.

Cette opération fut délicate pour plusieurs raisons. Un premier problème a été de modifier les fichiers des noms des employés car les mêmes employés figuraient dans ces fichiers pour les deux mois. Un second problème a été la fusion des données de deux fichiers pour des mois de longueurs différentes. En effet les rotations et par conséquent les services de vols ont du être modifiés pour respecter le même nombre de jours. Le mois de juillet a été choisi car c'est un mois de 31 jours alors que juin est un mois de 30 jours. Les modifications ont porté sur les tâches chevauchant le mois de juin et juillet.

Des tests semblables ont été effectués sur ce nouveau jeu de données. Le tableau suivant présente ces nouveaux résultats ainsi qu'un résumé des tests précédents.

Tableau 5.15 – Tableau résumé des résultats

	108 employés	**313 employés**	**626 employés**
identique %	61 %	57.2%	61.8 %
2 hrs %	72.1 %	78.9 %	82.4 %
4 hrs %	72.1 %	81.9 %	85,2 %
6 hrs %	72.2 %	84.2 %	86,9 %
8 hrs %	73.8 %	84.6 %	87,1 %
10 hrs %	73.8 %	85.7 %	88,3 %
12 hrs %	73.8 %	87.2 %	89,7 %

On peut remarquer que la réduction du nombre de tâches semblables augmente avec la taille du problème et on espère qu'elle sera significative pour les problèmes de grande taille.

En terme de temps de calcul, les résultats sur ce problème de 626 employés sont fort encourageants et pourraient l'être encore plus avec une réduction avant la construction du réseau.

Tableau 5.16 – Tableau résumé du % moyen de la diminution du temps de calcul

identique	**2 hrs**	**4 hrs**	**6 hrs**	**8 hrs**	**10 hrs**	**12 hrs**
23,6 %	41,3 %	42,1 %	51,4 %	51,7 %	52,4 %	54,3 %

Chapitre 6

Étude théorique : Calcul analytique de la taille du réseau

L'élimination des tâches identiques dans un premier temps, puis des tâches semblables dans un second temps a permis de diminuer globalement le temps de calcul sans trop modifier la qualité des horaires pour l'ensemble des employés.

Des tests ont été réalisés sur trois jeux de données de tailles différentes, un jeu de 108 employés, 313 employés et 626 employés, tout trois de la compagnie Air Transat.

Ces différents tests nous ont permis de nous rendre compte de l'impact de ces réductions. Cependant les problèmes d'agents de bord auxquels font face les grosses compagnies aériennes ont des tailles beaucoup plus importantes. Ainsi il serait intéressant de pouvoir évaluer analytiquement l'ampleur de la réduction pour de plus gros problèmes afin d'extrapoler les résultats. A titre d'exemple, si l'on passe d'un problème de 200 à 2000 employés, il y aura dix fois plus de sous problèmes et donc dix fois plus d'appels à Gencol et les sous problèmes seront dix fois plus gros, d'où l'intérêt d'une telle étude.

En moyenne il y a quatre copies de tâches par chaque tâche mère et en moyenne l'horaire d'un employé comporte trois tâches pour le problème de 108 employés et 6 tâches pour le problème de 313 employés. En éliminant les tâches identiques

et en ne travaillant qu'avec les tâches mères, le nombre de tâches mères restantes croît alors suivant une fonction linéaire du nombre d'employés à traiter avec un coefficient d'environ $(\frac{3}{4})$ pour le problème de 108 employés, car le nombre de tâches mères est égal à $(\frac{3}{4})$ * le nombre d'employés, et d'environ $(\frac{3}{2})$ pour le problème de 313 employés, pour les mêmes raisons.

Expérimentalement, dans un réseau de 108 employés, il y a 284 tâches filles, et après élimination des tâches identiques, il ne reste que 83 tâches mères au début du problème. Pour un réseau de 313 employés, il y a 1232 tâches filles et après élimination des tâches identiques, il ne reste que 456 tâches mères. Pour 626 employés, il y a 2672 tâches filles et après élimination des tâches identiques, il ne reste au début du problème que 993 tâches mères.

En supposant pour un problème de 5000 employés, qu'il y ait quatre copies par tâche mère et six tâches pour l'horaire de chaque employé, on estime à environ $(\frac{3}{2})$ * le nombre d'employés soit $(\frac{3}{2})$ * 5000 = 7500, le nombre de tâches mères. Ainsi, par projection le nombre de tâches filles présentes au début du problème sera d'environ 20 000 tâches.

Pour les petits réseaux, l'agrégation des tâches semblables diminue faiblement le réseau, donc peu d'enjeux. Cependant pour les plus gros réseaux cette réduction peut diminuer d'une manière substantielle le nombre de tâches car les heures des tâches sont plutôt homogènes. On peut se demander, quelle serait alors le nombre de tâches mères restantes après réduction sur un problème d'environ 5000 employés. Cette extrapolation nous renseignera sur le temps de résolution pour les grands problèmes et jusqu'à quelle taille on peut espérer les résoudre.

6.1 Première approche

A partir des hypothèses formulées plus haut sur le nombre de tâches filles et de tâches mères présentes au début d'un problème de 5000 employés, nous allons essayer d'évaluer la taille du réseau après élimination des tâches semblables commençant et terminant dans un intervalle de 6 heures.

A partir des probabilités d'occupation de chaque case, on pourra déterminer la probabilité qu'une case soit vide lors de 7500 essais, 1 essai correspondant à placer une tâche dans un case et ce pour les 7500 tâches mères estimées présentes au début du problème.

Pour utiliser l'étude probabiliste présentée à la fin du chapitre 4, il faut définir les classes et la méthode d'estimation des probabilités dans chaque classe. Les classes sont définies en discrétisant en cases de six heures un horizon d'un mois comme à la figure 6.1.
Dans cette figure, chaque case ou classe est à deux indices. L'indice i correspond à une date et une heure dans un intervalle de commencement d'une tâche et l'indice j à une date et une heure dans un intervalle de fin d'une tâche.
Après élimination des tâches filles de chaque réseau, on aimerait connaître comment se distribuent ses tâches mères en fonction d'un regroupement de 6 heures. On connaîtra alors le nombre de cases non vide, autrement dit le nombre de tâches présentes après élimination des tâches semblables c'est-à-dire commençant et terminant dans un intervalle de 6 heures.

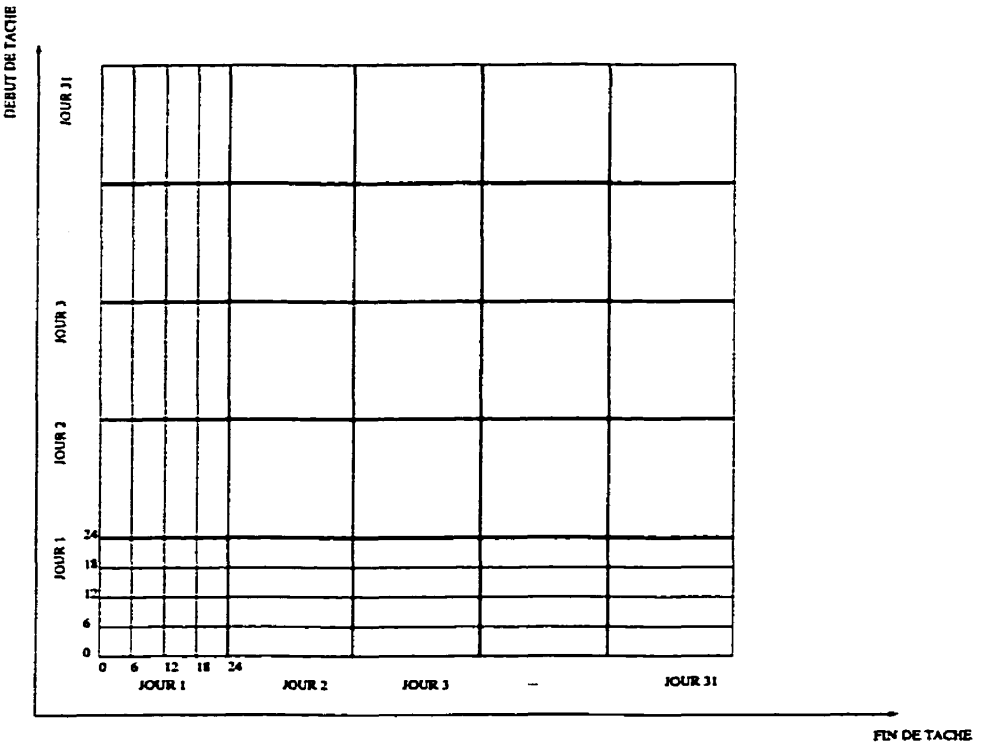


Figure 6.1 – Grille

Afin de déterminer l'espérance du nombre de cases vides, essayons d'évaluer la probabilité qu'une case soit vide. Pour estimer la probabilité de présence d'une rotation dans chaque case, nous allons utiliser les données de deux mois.

Afin de vérifier l'uniformité au cours de la période des deux mois considérés, on a tracé à la figure 6.2, les distributions du nombre de cases ayant des fréquences égales, c'est-à-dire avec le même nombre de rotations présentes dans une case pour les mois de juin et juillet. On peut s'aperçevoir que, le nombre de cases ayant une certaine fréquence de tâche par case, est sensiblement identique d'un mois à l'autre.

Après observation de la dispersion des tâches dans une grille pour le problème test comportant 313 employés pour chacun des mois de juin et juillet, nous avons

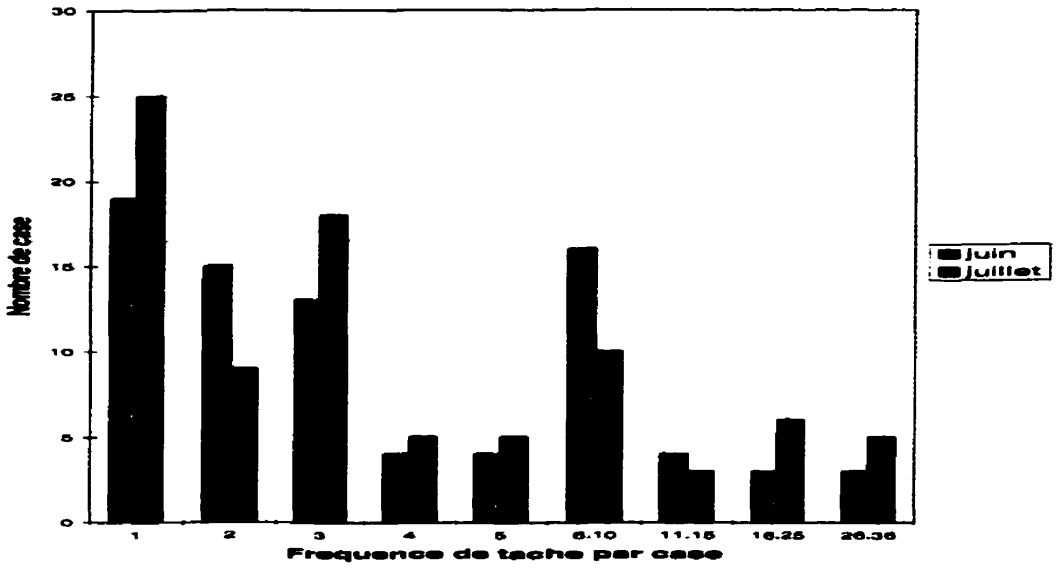


Figure 6.2 – *Distribution du nombre de cases ayant un même nombre de rotations*

obtenues des moyennes de probabilité d'occupation par semaine.

Pour chaque semaine avec une discrétisation au six heures (quatre cases de six heures par jours pour les sept jours de la semaine), il y a 28 * 28 cases c'est-à-dire 784 cases. Après observation du nombre de cases occupées par semaine pour un problème d'environ 300 employés et en superposant ces semaines pour les mois de juin et juillet, on constate que 132 cases sont occupées par semaine.

A partir de ces observations et en utilisant nos hypothèse, nous allons estimer l'espérance du nombre de cases non vides parmi ces 132 cases pour un problème de 5000 employés contenant 7500 tâches mères au début du problème.

Pour évaluer la probabilité que chaque case reste vide après 7500 essais, il suffit de calculer la probabilité qu'un événement ne se réalise pas lors de n essais, soit comme on l'a vu au chapitre 4, la bînomiale $\mathcal{B}_{i,j}(\ n,\ 0\ )$ pour chacune des cases (i,j) : soit $P(X = 0) = \mathcal{B}_{i,j}(\ n,\ 0\ ) = (\ {}^{n}_{0}\ )\ P^{0}_{i,j}\ (1\text{-}P_{i,j})^{n-0} = (\ 1\ -\ P_{i,j})^{n}$.
Avec n égal à 7500 il vient : $\mathcal{B}_{i,j}(\ 7500,\ 0\ ) = (\ 1\ -\ P_{i,j})^{7500}$ avec $P_{i,j}$ la probabilité qu'une case soit occupée.

L'espérance du nombre de cases vides pour 5000 employés, est égale à la somme des bînomiales de ces 32 cases non vides plus le nombre de cases avec des $P_{i,j}$ nuls c'est-à-dire les cases étant restées vides. Elles sont au nombre de 652 car 652 cases n'ont pas été remplies et ont donc des probabilités de rester vide, égale à un.

Après calcul, la somme des bînomiales pour les 132 cases occupées est égale à $4,7 \simeq 5$. Par conséquent, l'espérance du nombre de cases non vides pour un problème de 5000 employés est égale à, $784 - (652 + 5)$ soit 127 tâches restantes par semaine. Ainsi pour un mois, l'espérance du nombre de cases non vides est de $127*4,3 = 546,1$ soit **546** tâches restantes car il y a 4,3 semaines pour un horizon d'un mois.

Cette réduction représente une élimination de 97,2 % des tâches filles présentes au début du problème composé, comme nous l'avons supposé de 20 000 tâches filles. Ce pourcentage semble réaliste et cohérent avec les pourcentages de réduction obtenus pour des problèmes plus petits. En effet, pour 108 employés, l'élimination des tâches semblables dans un intervalle de six heures a diminué de 81,7 % le nombre de tâches au début du problème, pour 313 employés de 84,9 % et pour 623 employés de 86,9 %. A raison d'une augmentation du pourcentage de 1,04 % à chaque fois que l'on triple le nombre d'employés présents au début du problème, un pourcentage de

97,2 % de tâches enlevées est de l'ordre de grandeur attendue.

Remarque

Afin de valider les estimations que l'on a faîtes pour évaluer le nombre de cases occupées pour un jeu de donnée de 5000 employés, utilisons la même méthode pour un problème de 313 employés.

Il reste expérimentalement pour le jeu de données du mois de juin contenant 313 employés et pour un horizon d'un mois, **186** tâches mères au début du problème après regroupement des tâches semblables au six heures. Pour le mois de juillet avec le problème de 313 employés, il reste **197** tâches mères.

Comparativement en estimant la probabilité $P_{i,j}$ d'occupation d'une case avec les données du mois de juin et juillet et en utilisant l'estimation basée sur la loi bînomiale pour le problème de 313 employés, nous estimons que l'espérance du nombre de cases non vides parmi les **82** cases occupées, pour le mois de juin est de 37 et parmi les **86** cases occupées pour le mois de juillet de 39,7. Il vient alors que l'espérance du nombre de cases non vides pour le mois de juin est de 784 - (37 + 702) soit 45 cases pour une semaine (car il y 702 cases sur les 784 cases qui ont eu des probabilités $P_{i,j}$ nuls). Pour un mois, l'espérance du nombre de cases non vides est de 45 * 4,3 = **193** tâches restantes. De même pour le mois de juillet, l'espérance du nombre de cases non vides est de 784 - (39,7 + 698) soit 46,3 cases par semaine et donc 46,3 * 4,3 $\simeq$ **199** tâches restantes par mois.

Nos hypothèses semblent ainsi avoir du sens pour le problème de 313 employés

et on suppose qu'il en sera de même pour des problèmes de plus grande taille. On a estimé plus haut l'espérance du nombre de cases occupées pour un problème de 5000 employés à **546**. D'autre part, on a établi sur une moyenne de deux mois, que le nombre de cases occupées par semaine est de 132, soit en utilisant nos hypothèse que les semaines se ressemblent, 132 * 4,3 $\simeq$ 568 cases devraient être occupées pour un mois. Nos estimations sont donc bonnes car on "sature" le nombre de cases non vides espérées, autrement dit une case occupée pour un problème de 313 employés a de grande chance d'être occupé pour un problème de 5000 employés.

Cependant, en faisant ceci, on sous-estime l'espérance du nombre de cases non vides. En effet, si l'espérance de cases non vides parmi les cases occupées est assez fidèle , il en est autrement pour les 652 autres cases par semaine, vide et supposé comme tel pour 5000 employés. L'erreur n'est peut être pas importante mais notre méthode est sûrement biaisée.

C'est pourquoi une autre approche a été étudiée afin d'évaluer le nombre initial de tâches restantes que pourrait contenir un problème de 5000 employés après réduction des tâches semblables.

6.2 Seconde approche

Précédemment, nous avons essayé de déterminer la distribution du nombre de tâches suivant l'heure de départ et d'arrivée. La probabilité des cases ayant des probabilités $P_{i,j}$ faibles est mal estimée quand l'estimation est faite en utilisant un petit échantillon (deux jeux de données de 300 employés). En effet, plusieurs $P_{i,j}$

petits sont remplacés par des zéros car aucune tâche n'a été obtenue dans cette case avec des petits échantillons.

Afin de réduire l'erreur, notre seconde approche vise à agréger les données pour obtenir de plus grands échantillons et ainsi une meilleure estimation des faibles probabilités. Cette seconde approche consiste à établir les distributions marginales des tâches suivant les heures de départ par semaine et des tâches en fonction de leur durée. On obtiendra alors les lois marginales pour ces deux distributions, soit P_i la distribution de probabilité pour les durées des rotations et Q_j pour les heures de départ. La même discrétisation en cases de six heures est utilisée pour le calcul des $P_{i,j}$ dans chaque case.

On suppose que ces deux variables aléatoires sont indépendantes, c'est-à-dire les heures de départ ne dépendent pas de la durée d'une tâche. Intuitivement, cela semble réaliste. On pourrait facilement par cette hypothèse d'indépendance obtenir une distribution conjointe, soit $P_i * Q_j$, la probabilité qu'une case soit occupée. On pourra alors évaluer l'espérance du nombre de cases occupées. Cette distribution de probabilité sera alors plus répartie que celle dans la première approche et va nous permettre d'établir une borne supérieur pour le nombre de tâches restantes.

Sur la figure 6.3 représentant le % de tâches suivant la durée d'une tâche, on peut voir que les durées sont sensiblement les mêmes pour le mois de juin et juillet et qu'il y a un fort pourcentage de tâches ayant une durée d'un jour.

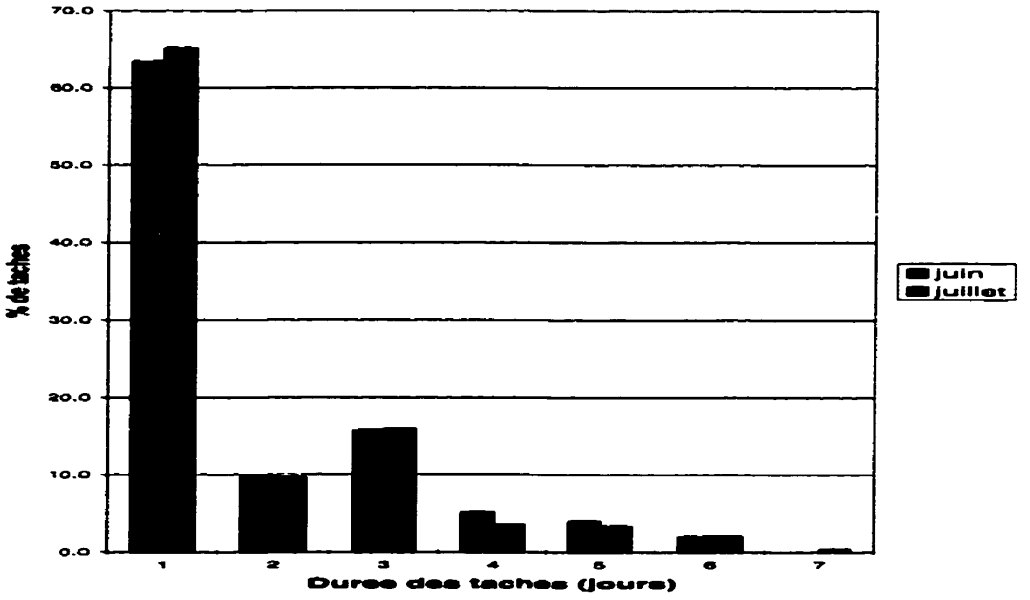


Figure 6.3 – *Distribution du % de tâches en fonction de leurs durées*

On observe de plus sur cette figure qu'il n'y a pas de tâche ayant une durée supérieure à sept jous, en d'autres termes les cases avec une durée ≥ 7 ont une probabilité $P_i = 0$. Il vient directement que la distribution conjointe $P_i * Q_j = 0$ pour toutes ces cases.

Sur la figure 6.4, représentant le % de tâches suivant l'heure de départ, on ne peut dégager une tendance entre le mois de juin et juillet car la discrétisation est trop fine.

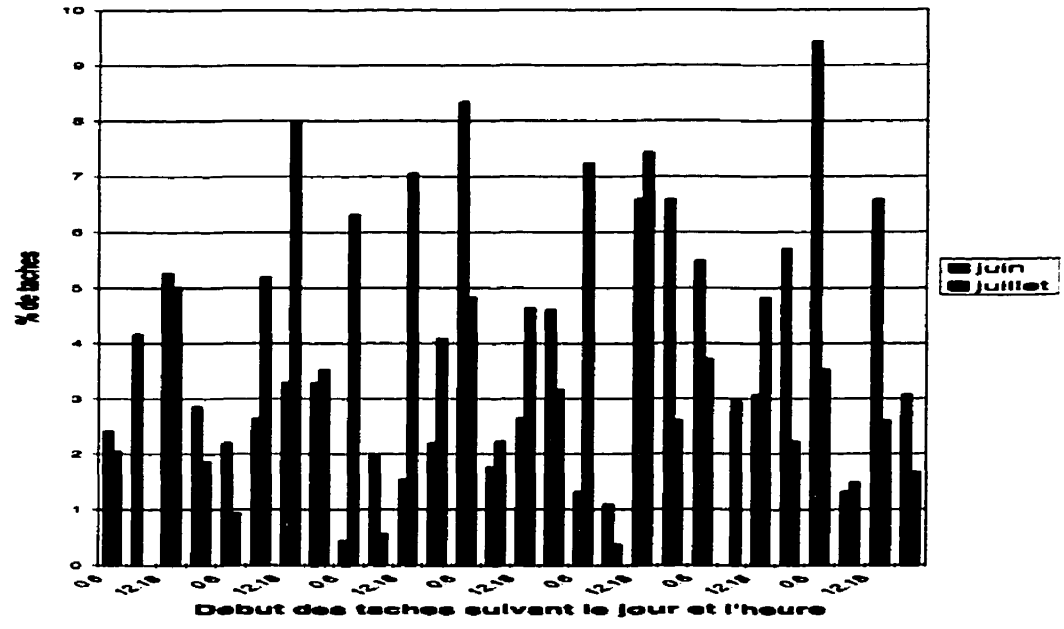


Figure 6.4 – *Distribution du % de tâches par tranche d'heure et jour de départ*

On a alors agrégé les données afin de déterminer le pourcentage de tâches commençant chaque jour et le pourcentage de tâches commençant à chaque tranche d'heure. Sur la figure 6.5, on peut observer qu'il y a à peu près le même pourcentage de tâches à chaque jour pour le mois de juin et juillet. On s'aperçoit que trois jours sont presque identiques, le jour 4, 5 et 6. Deux jours sont assez proches, le jour 1 et 2. Et il y a un déplacement assez important des vols le jour 3 et 7 de juin à juillet.

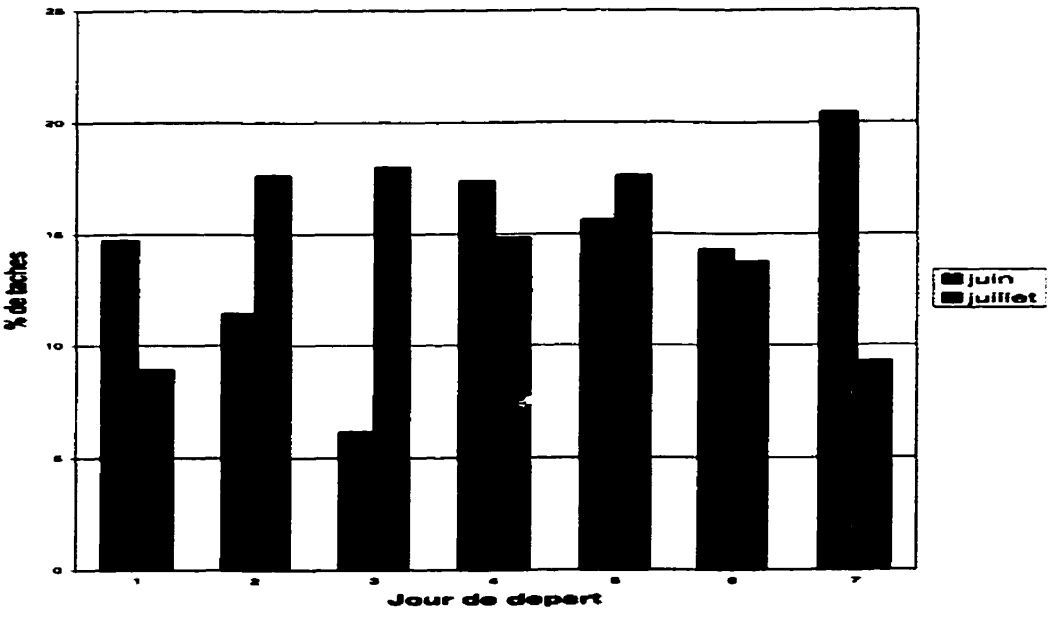


Figure 6.5 – *Distribution du % de tâches en fonction du jour de départ*

Sur la figure 6.6 page suivante, on peut observer également que le pourcentage de tâches commençant dans les mêmes tranches d'heures sont relativement équivalents pour le mois de juin et juillet.

À la lumière de ces figures, on peut en déduire que l'on n'est pas trop loin de l'indépendance, même si il y a une certaine dépendance. En effet, l'hypothèse de notre approche repose sur l'indépendance des deux distributions marginales, celle des tâches suivant les heures de départ par semaine et celle des tâches en fonction de leurs durées. L'hypothèse d'indépendance signifierait que les heures de départ des tâches ne dépendent pas des durées de ces tâches. Cependant, on voit sur les figures 6.5 et 6.6 que les pourcentage de tâches varient légèrement suivant le jour et l'heure de départ, ce qui prouve que l'indépendance n'est pas totale.

L'intérêt d'utiliser l'hypothèse d'indépendance, et que l'on obtiendra des estimations des $P_{i,j}$ plus réparties que la réalité. Ceci amènera ainsi à sur-estimer faiblement l'espérance du nombre de cases non vides et ainsi d'obtenir une borne supérieur du nombre de tâches, qui seront présentes après réduction.

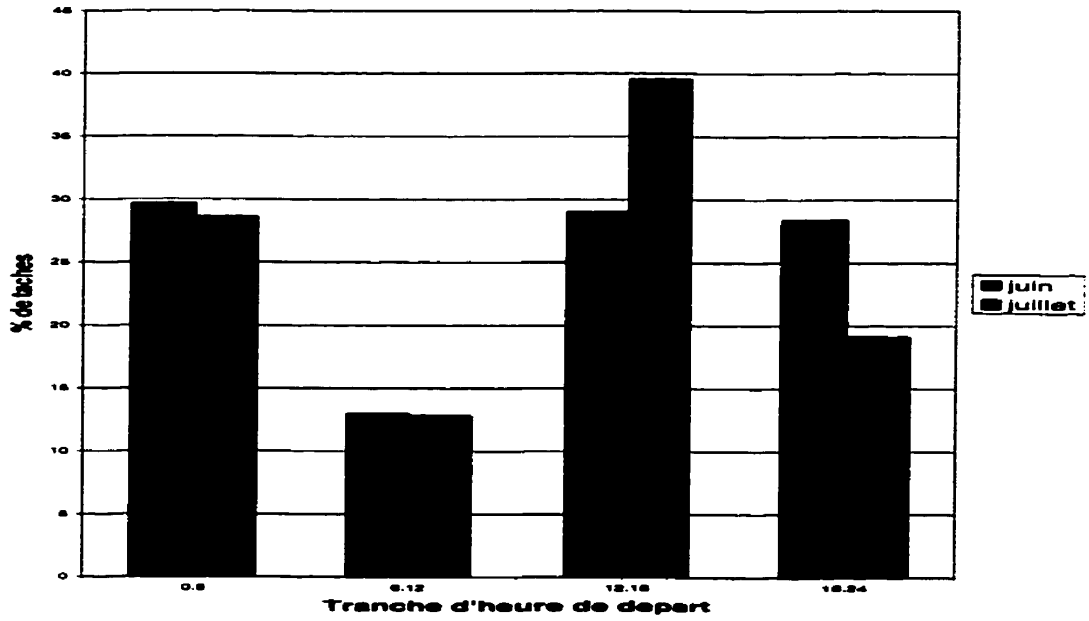


Figure 6.6 – *Distribution du % de tâches par tranche d'heure de départ*

A partir du produit cartésien des deux distributions de probabilités, on a obtenu la probabilité $P_i * Q_j$ qu'une case (i, j) soit occupée. De même que dans la première approche, on a établit l'espérance du nombre de cases vides en calculant la bînomiale pour chaque case.

Le nombre de cases présentes est égal au nombre d'heures de départ possible multiplié par le nombre de durées de tâches différentes. Ainsi, le nombre d'heures de départ possible est égal à 7 * 4, car il y a 7 jours de départ possible et pour chaque

jour, 4 dates de départ possible (0-6, 6-12, 12-18, 18-24 heures). Le nombre de cases est donc de (7 * 28) cases par semaine soit 196 cases. Il y a donc moins de cases totales que dans la première approche car un certain nombre d'entre elles ont des probabilités $P_i = 0$ à cause de la durée des tâches qui est inférieure à 7 jours.

Il vient après calcul que l'espérance du nombre de cases vides (somme des bînomiales) est égale à 36 pour le mois de juin, 28.6 pour le mois de juillet et 33.2 pour la moyenne des deux mois.
Toutes les cases ont été occupées, ainsi l'espérance du nombre de cases non vides est égal au nombre total de cases moins l'espérance du nombre de cases vides.

On a vu qu'il y a 196 cases. L'espérance du nombre de cases non vides pour le mois de juin est donc de 160 par semaine (196 - 36 = 160), soit 4,3 * 160 = 688 cases pour un mois.
Pour le mois de juillet de la même manière, on obtient 196 - 28,6 = 167,4 cases non vides soit 720 cases non vides pour un mois et en moyenne sur les deux mois, 196 - 33,2 = 169,8 cases non vides soit **700** cases non vides pour le mois.

On a obtenu par la deuxième approche une borne supérieur. L'espérance du nombre de tâches restantes après réduction des tâches semblables au six heures, semblent être ainsi entre notre première borne inférieur de la première approche soit **546** et notre borne supérieur soit **700**.

Chapitre 7

Conclusion

Dans ce mémoire de maîtrise nous nous sommes intéressé à la réduction de réseaux pour le problème de construction d'horaires d'équipage en transport aérien.

Les tests effectués et les résultats présentés ici, ont permis de réduire sensiblement les réseaux et ainsi de rencontrer les objectifs visés. En effet, l'élimination des tâches semblables a permis de réduire les réseaux quelque soit leur taille, d'environ 60 % des tâches. L'élimination des tâches semblables a quant à elle, réduit les réseaux pour des petits problèmes de l'ordre de 100 employés de 72,2 % et pour les plus gros problèmes d'environ 85 % des tâches. Cette réduction a permis ainsi de diminuer sensiblement le temps de résolution globale du problème.

On a pu observer d'autre part que la réduction des tâches semblables augmentent en fonction de la taille du problème. On a ainsi pu estimer analytiquement dans la section précédente, l'ampleur d'une telle réduction pour des problèmes de l'ordre de 5000 employés. En supposant que la fréquence des vols se ressemblent à chaque semaine, on a estimé le nombre de tâches présentes au début d'un problème contenant initialement 20 000 tâches filles, à un chiffre entre 546 et 700 tâches. Ceci est encourageant pour la résolution des gros problèmes qui auraient nécessité sans cela des temps de calcul beaucoup plus importants.

De plus, la réduction pourrait être encore plus considérable parmi l'ensemble des arcs du réseau avec une implémentation des fonctions de réduction en amont de la construction du réseau. Ceci fera l'objet d'une nouvelle version du logiciel dans lequel certaines des tâches enlevées devront être réintroduites dynamiquement. L'implémentation d'une telle réduction n'est pas du tout trivial et impliquerait de recoder tout le logiciel, ce qui est impensable dans le cadre de ce mémoire.

Actuellement, le logiciel ne fournit pas de solutions optimales. La dominance porte sur un petit nombre de ressources et ceci afin d'éviter de surcharger la mémoire. Désormais avec la réduction de la taille des réseaux, il est désormais envisageable de dominer sur un plus grand nombre de ressources et ainsi obtenir de meilleurs solutions.

D'autre part, une procédure d'échanges reste à être implémentée pour modifier une solution lorsque des erreurs peuvent être décelées à la main. Par exemple si l'on s'aperçoit qu'une rotation refusée à un senior a été accordée à un junior, on envisage de "réparer" l'horaire du senior, même si l'horaire du junior s'en trouve détruit. Cette heuristique devra être implantée pour permettre des échanges de rotations entre certains employés afin d'améliorer la qualité de la solution chez certains seniors.

Bibliographie

[1] ANBIL, R., GELMAN, E., et PATTY, B. et al. (1991) Recent Advances in Crew Pairing Optimization at American Airlines. *Interfaces*, no.21, p.62-74.

[2] ANERA, Y.P, AGGARWAL, V., et NAIR, K.P.K. (1983) Shortest Chain Subject to Side Constraints. *Network*, p. 295-302, no.21, p.62-74.

[3] BALL, M. et ROBERTS, A. (1985) A graph Partitionning Approch to Airline Crew Scheduling. *Transportation Science*, no.19(2), p.107-126.

[4] BELLMAN, R.E. (1958) On a Routing Problem. *Q.Appl. Math.*, no.16 p.87-90.

[5] BYRNE, J. (1988) A Prefential Bidding System for Technical Aircrew. *AGIFORS*, Symposium Proceeding 28, p.87-99.

[6] CPLEX REFERNCE MANUAL (1992) Using the CPLEX Callable Library and CPLEX Mixed Integer Library. *CPLEX Optimization, Inc., Incline Village*, NV 89451-9436, U.S.A.

[7] DANTZIG, G., et WOLFE, P. (1960) The decomposition algorithme for linear programming. *Operations Research*, no.8, p.101-111.

[8] DESAULNIERS, G., DESROSIERS, J., et DUMAS, Y. et al.(1997) Daily Aircraft Routing and Scheduling. *Management Science 43*, 841-855.

[9] DESAULNIERS, G., DESROSIERS, J., et DUMAS, Y. et al. (1997) Crew Pairing at Air France, *EJOR 97*, 245-259.

[10] DESAULNIERS, G., DESROSIERS, J., et GAMACHE, M. et al. (1998) Crew Scheduling in Air Transportation. *Fleet Management and Logistics*, T.Crainic et G. Laporte (eds.), Kluwer, p.169-185.

[11] DESAULNIERS, G., et VILLNEUVE, D. (1998) The shortest path problem with time windows and linear waiting costs. *Les Cahiers du GERAD*, no.G-97-21.

[12] DESROCHERS, M., DESROSIERS, J., et SOLOMON, M.M. (1992) A New Optimization Alghorithme for the Vehicule Routing Problem with Time Windows. *Operations Research*, no.40, p.342-354.

[13] DESROCHERS, M. (1986) La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes. *Thèse de Ph.D.*,, Université de Montréal, Montréal, Canada.

[14] DESROCHERS, M., et SOUMIS, F. (1988a) A Reoptimization Algorithme for the Shortest Path Problem with Time Windows. *European Journal of*

Operational Research, no.35, p.242-254.

[15] DESROCHERS, M., et SOUMIS, F. (1988b) A generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows. *INFOR 26*, p.191-212.

[16] DESROSIERS, J., DUMAS, Y., et DESROCHERS, M. et al. (1991) A Breakthrough in Airline Crew Scheduling. *Proceedings 26th Annual Meeting of the Canadian Transportation Research Forum*, Québec, p.464-478.

[17] DESROSIERS, J., DUMAS, Y., et SOLOMON, M.M et al., (1995) *Time Constrained Routing and Scheduling.* in BALL, M.O et al. *Handbook in OR & MS : Network Routing.* Elsevier Science B.V, vol.8, M.O, p.35-139.

[18] DESROSIERS, J., LASRY, A., et MCINNIS, D. et al. (1995) The Airline Operations Management System at Air Transat.*Les Cahiers du GERAD*, no.G-95-23.

[19] DESROSIERS, J., PELLETIER, P., et SOUMIS, F. (1983) Plus court chemin avec contraintes d'horaires., *RAIRO Recherche opérationnelle*, vol. 17, n.4, p. 357-377.

[20] GAMACHE, M. (1995) Fabrication d'horaires mensuels pour les membres d'équipages en transport aérien. *Thèse de Ph.D.,* Ecole Polytechnique de

Montréal, Montréal, Canada.

[21] GAMACHE, M., et SOUMIS, F. (1998) A Method for Optimally Solving the Rostering Problem. *Operation Research in the Airline Industry*, G.Yu (ed.), Kluwer, Boston, 124-157.

[22] GAMACHE, M., SOUMIS, F., et MARQUIS, G. (1998) et al. A column Generation Approch for Large Scale Aircrew Rostering Problems. *Les Cahiers du GERAD*, no.G-94-20.

[23] GAMACHE, M., SOUMIS, F., et VILLENEUVE, D. et al. (1997) The Preferential Bidding System at Air Canada. *Les Cahiers du GERAD*, no.G-97-12.

[24] GERSHKOFF, L. (1989) Optimizing Flight Crew Schedules. *Interfaces*, no.19, p.29-43.

[25] GRAVES, G.W., McBRIDE, R.D., et GERSHKOFF, I. et al. (1993) Flight Crew Scheduling. *Management Science*, no.39, p.736-745.

[26] HANDLER G.Y., et ZANG I. (1980) A Dual Algorithm for the Constrained Shortest Path Problem. *Networks*, 10, p.293-310.

[27] JAFFE J.M. (1984) Algorithms for the Finding Paths with Multiple Constraints . *Networks*, 14, p.95-116.

[28] LASRY, A. (1996) Rotations déquipage pour un transporteur aérien en milieu régional. Mémoire de maîtrise : Sciences de la gestion : Université de Montréal.

[29] LAVOIE, S., DESROSIERS, J., et DUMAS, Y. et al. (1995) An Optimizer for Aircraft and Crew Scheduling. *Les Cahiers du GERAD*, no.G-94-23.

[30] MACULAN, N., et SALLES, J.J.C. (1991) A lower bound for the Shortest Hamiltonean Path in Directed Graphs. *OR Spektrum*, Symposium, 13, p.99-102.

[31] MINOUX, M. (1975) Plus court chemin avec contraintes : algorithmes et applications. *Annales des télécommunnications*, tome 30, p.326-329.

[32] MOORE, R., EVANS, J., et NGO, H. (1978) Computerized Tailored Blocking. *AGIFORS*, Symposium Proceeding 18, p.343-361.

[33] NAGIH, A. (1999) L'agégation des contraintes de ressources dans un problème de plus court chemin. *Les Cahiers du GERAD*, no.G-99-02.

[34] PIRES, J.M. (1997) Développement de méthodes parralèles pour des problèmes de grandes tailles. *Thèse de Ph.D.*,, École Polytechnique de

Montréal, Montréal, Canada.

[35] SALKIN, H.M (1975) *Integer Programming.*, Massachussett, Addison-Wesley.

[36] ZIARATI, K., SOUMIS, F., et DESROSIERS, J. (1998) Locomotive Assignement Using Train Delays, **7th International Workshop on Computer-Aided Scheduling of Public Transport** Boston.

[37] ZIARATI, K., SOUMIS, F., et DESROSIERS, J., (1997) et al. Locomotive Assignement with Heterogeneous Consists at CN North America, bf EJOR 97, 281-292.

Annexe A : Description et terminologie

Dans les pages qui suivent nous décrivons la terminologie spécifique au domaine du transport aérien.

Aller-retour : Vol qui part de la base et qui revient à la base dans un même duty.

Bank Time : Temps accumulé par un membre d'équipage.

Base/Domicile : Endroit définit par les compagnies comme étant la/les principales stations de leurs opérations où sont assignés des membres d'équipage et où débutent et se terminent les rotations.

Bid Option : Chacune des options incluses dans l'interface

Bidder Graphical Interface Interface graphique du logiciel d'optimisation PBS, utilisé par les membres d'équipage pour exprimer leurs préférences.

Bloc/ligne : L'horaire d'un mois donné d'un PNC/PNT.

Bloc Réserve : Horaire constitué de jours où le membre d'équipage doit être disponible en tout temps. Ces gens se promènent habituellement avec des télé-avertisseurs.

Block Time : Les heures réelles d'un vol.

Briefing : Réunion d'employés où les membres de l'équipage se préparent à effectuer la journée de travail.

CARs : Les règlements Canadiens de l'aviation.

Check-in : Début d'une période en devoir correspondant à l'embarquement des passagers.

Checkout/Release Time : La fin d'une période en devoir qui correspond à la durée allouée par la compagnie pour le débarquement des passagers.

Connections/temps d'attente au sol : Temps de pause où l'équipage est au sol entre deux segments de vol. Un repos est une longue connexion qui se trouve entre deux services de vol. En général, les repos se passent la nuit afin de permettre au personnel de dormir.

Credit Time : Les heures payées d'un vol.

De-briefing : Réunion d'employés où les membres de l'équipage donnent un compte rendu de la journée et rapportent tout incident spécial.

Destination : Le point d'arrivée d'un tronçon.

En Devoir/On Duty : Le temps entre chaque check-in et checkout.

Escale/Layover : Ville où il y a un repos. Une période entre 2 services de vols où les gens ne sont pas en devoir, ni à leurs bases domicilières.

Jour de Congé : Chez Air Transat et Air Canada, un nombre minimum de jours de congé est habituellement prédéterminé. Des patrons de dispositions des congés sont aussi élaborés à l'avance.

Language Qualified : Un membre d'équipage qui parle une langue étrangère. Chez Air Transat, ils doivent assurer des vols allant à certaine station des gens qui parlent et comprennent la langue de cette station.

Low Block : Un horaire où on accepte que les heures de crédits ou blocs soient plus basses que la valeur minimum.

Open Time : Vols non-attribués aux membres d'équipages.

Pairing/rotation/trip/tâche/tournée : Synonymes pour exprimer une suite alternée de service de vol et de périodes de repos entre un départ et un retour à la base d'assignation du personnel.
Une rotation d'équipage doit être valide, c'est-à-dire qu'elle doit respecter les lois gouvernementales régissant les opérations de l'aviation civile, ainsi que les conventions collectives du personnel navigant technique et cabine. Il peut y avoir plusieurs bases pour une compagnie aérienne, mais un équipage est affecté à une seule base, généralement son lieu de résidence.
Le temps d'absence d'une rotation varie d'une durée allant d'une seule journée (dans un contexte court courrier) à deux ou trois semaines (dans un contexte long courrier). Lorsqu'une rotation dure une seule journée, il n'y a pas de repos de nuit crédité aux membres du personnel navigant puisqu'ils pourront rentrer chez eux après le de-briefing.

Période de repos/Crew rest : Période de repos allouée aux membres d'équipage durant laquelle la compagnie ne peut pas les déranger.

PNC/PNT : Abréviation pour Personnel Navigant Cabine et Personnel Navigant Technique. Le personnel navigant cabine peut être affecté à n'importe quel type d'avion ; il inclut le chef de cabine et les agents de bord. Les membres du personnel navigant technique (pilote, copilote et mécanicien) sont qualifiés pour contrôler un appareil ; ils sont accrédités à un type d'avion.

Positionnement/Mise en place : Période durant laquelle un membre d'équipage effectue un segment de vol en passager sur un vol offert par la compagnie aérienne ou sur un vol offert par une autre compagnie partenaire ou concurrente.

Service de vol (SDV) : Suite de vols, de positionnements et de connections effectués par un équipage durant une journée de travail.

Soumission/Bid : Fichier qui inclut les demandes d'un membre d'équipage pour un mois donné.

SPPRC : Plus court chemin avec contraintes de ressources (en anglais Short path problem with ressource constraints)

SPPTW : Plus court chemin avec fenêtres de temps (en anglais Short path problem with time windows)

Tronçon de vol (TDV) : Portion d'un vol sans escale. Élément de base pour la construction des rotations. Dans le problème de rotations d'équipage, on suppose

que les segments de vol sont connus.