



**Titre:** Recherche tabou et agrégation dynamique de contraintes pour la  
Title: construction d'horaires mensuels d'équipages aériens dans un  
contexte d'équité

**Auteur:** Khaled Boubaker  
Author:

**Date:** 2006

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Boubaker, K. (2006). Recherche tabou et agrégation dynamique de contraintes  
Citation: pour la construction d'horaires mensuels d'équipages aériens dans un contexte  
d'équité [Master's thesis, École Polytechnique de Montréal]. PolyPublie.  
<https://publications.polymtl.ca/7804/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/7804/>  
PolyPublie URL:

**Directeurs de  
recherche:** Guy Desaulniers, & Alain Hertz  
Advisors:

**Programme:** Unspecified  
Program:

**UNIVERSITÉ DE MONTRÉAL**

**RECHERCHE TABOU ET AGRÉGATION DYNAMIQUE DE CONTRAINTES  
POUR LA CONSTRUCTION D'HORAIRES MENSUELS D'ÉQUIPAGES  
AÉRIENS DANS UN CONTEXTE D'ÉQUITÉ**

**KHALED BOUBAKER  
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL**

**MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLOME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(MATHÉMATIQUES APPLIQUÉES)  
DÉCEMBRE 2006**



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*

*ISBN: 978-0-494-25532-2*

*Our file    Notre référence*

*ISBN: 978-0-494-25532-2*

#### NOTICE:

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

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

#### AVIS:

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

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

---

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

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

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

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

  
**Canada**

**UNIVERSITÉ DE MONTRÉAL**

**ÉCOLE POLYTECHNIQUE DE MONTRÉAL**

Ce mémoire intitulé :

**RECHERCHE TABOU ET AGRÉGATION DYNAMIQUE DE CONTRAINTES  
POUR LA CONSTRUCTION D'HORAIRES MENSUELS D'ÉQUIPAGES  
AÉRIENS DANS UN CONTEXTE D'ÉQUITÉ**

présenté par : **BOUBAKER Khaled**

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

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

M. **SOUMIS François**, Ph.D., président

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

M. **HERTZ Alain**, Doct. ès Sc., membre et codirecteur de recherche

M. **GAMACHE Michel**, ing., Ph.D., membre

*À mes parents,*

*Auxquels je dois tout,*

*Pour l'amour qu'ils me portent,*

*Pour toute la peine qu'ils se sont donnée,*

*Pour leur patience et générosité,*

*Je dédie ce travail en témoignage de ma profonde gratitude et ma reconnaissance infinie.*

*À mon frère,*

*À ma sœur,*

*À tous ceux qui me sont chers,*

*Que ce travail soit l'expression de mes vrais sentiments.*

## REMERCIEMENTS

Mes plus vifs remerciements s'adressent à mon directeur de recherche M. Guy Desaulniers qui m'a assisté et m'a guidé durant l'avancement de ce projet ; je lui exprime toute ma gratitude et lui adresse mes plus sincères respects. Je voudrais aussi remercier M. Alain Hertz, mon codirecteur de recherche, qui a enrichi le contenu de ce mémoire avec ses idées originales et ses commentaires constructifs.

J'exprime également ma vive reconnaissance à tous les membres du jury et plus particulièrement à M. François Soumis pour m'avoir fait l'honneur d'accepter de présider le jury de ce mémoire.

Je tiens à remercier également les membres et professionnels du GERAD qui m'ont aidé tout au long de mon travail, ainsi que Karine Dufresne, Norbert Lingaya et Eric Gélinas de la compagnie Kronos Inc.

Je voudrais finalement remercier tous mes amis et plus particulièrement Hichem qui m'a été d'un grand soutien tout au long du projet.

Que tous ceux qui ont contribué de près ou de loin à ce travail trouvent ici l'expression de ma reconnaissance.

# RÉSUMÉ

Le projet discuté dans ce mémoire traite de la construction d'horaires mensuels de membres d'équipage d'une compagnie aérienne pour des problèmes de grande taille. En effet, les données considérées lors du processus de résolution sont de l'ordre de 3000 rotations et quelques 500 employés. On définit l'horaire mensuel d'un employé comme une suite de rotations, de repos, de congés et d'entraînements sur un horizon d'un mois. Lors du processus de création des horaires mensuels réglementaires pour les membres d'équipage de la compagnie, les rotations sont attribuées de façon à couvrir tous les vols de la compagnie avec le personnel disponible. Dans le cadre de ce mémoire, le choix s'est porté sur la méthode du Bidline pour l'élaboration des horaires mensuels. En effet, chaque membre d'équipage est traité de façon impartiale et on cherche à équilibrer les jours de congé ainsi que le nombre d'heures travaillées pour tous les employés tout en respectant les règles de validité d'un horaire. L'objectif de ce projet est développer une méthode qui permettra de résoudre des problèmes de grande taille plus rapidement que les algorithmes existants.

Pour atteindre cet objectif, une nouvelle approche est adoptée pour la résolution du problème. En effet, une récente méthode, en l'occurrence, l'agrégation dynamique de contraintes est choisie pour l'élaboration des horaires personnels. Cette méthode est très prometteuse et donne des résultats nettement meilleurs que la génération de colonnes. Le mécanisme de résolution est en fait une alternance d'agréations et de désagréations de plusieurs tâches de façon à diminuer la taille du problème à résoudre lors de l'itération courante de la

génération de colonnes. Cependant, cette nouvelle approche requiert une solution initiale de bonne qualité pour entamer le processus de résolution. Le choix s'est porté sur l'utilisation d'une métaheuristique Tabou pour le développement de la solution initiale. En effet, puisque le problème de construction des horaires est un problème de partitionnement, il est réductible en un problème de coloration de graphes. Pour ce genre de problèmes, il existe plusieurs algorithmes efficaces dans la littérature.

L'analyse des résultats des trois méthodes, en l'occurrence l'agrégation dynamique, la génération de colonnes et la méthode Tabou, démontre la robustesse de l'approche de résolution adoptée dans ce mémoire. En effet, pour certains problèmes résolus, l'agrégation dynamique offre un gain de 98% pour les temps de calcul par rapport à la méthode traditionnelle de génération de colonnes et une solution de meilleure qualité d'un point de vue de l'uniformisation des horaires produits. D'un autre côté, l'agrégation dynamique produit des horaires de qualité nettement supérieure aux horaires produits par la méthode Tabou en des temps de résolution comparables. Par conséquent, les résultats montrent qu'une métaheuristique jumelée à une méthode exacte, l'agrégation dynamique de contraintes, offre un bon compromis pour les problèmes considérés.



# ABSTRACT

The focus of this master's thesis is the construction of monthly schedules for airline crew members. This problem is known in the industry as airline crew scheduling problem and we are focusing on solving large instances. In fact, the data considered during for our experiments includes about 3000 pairings and some 500 employees. A pairing is defined as a sequence of flights interspersed with connections and layovers and a schedule as a sequence of pairings and days off. In this thesis, the method chosen for the design of the schedules is the Bidline. Indeed, the employees are considered similar and each crew member is treated impartially. We seek to standardize the schedules by balancing the number of days off as well as the number of worked hours for all of the employees while complying with labor and safety rules. The objective of this project is to develop a new method which can solve large instances more quickly than the existing algorithms.

To achieve this goal, a new approach is adopted to solve the airline crew scheduling problem: namely, the recent dynamic constraint aggregation algorithm. This method is very promising and produces better results than column generation. This mechanism aggregates and disaggregates several tasks in order to decrease the size of the problem to be solved during the column generation procedure. However, this new approach requires a good initial solution to start the solution process. We choose to use a Tabu algorithm to compute such an initial solution. Since the airline crew scheduling problem is a set-partitioning problem, it can approximated as a graph coloring problem for which Tabu search has proven to yield good quality results.

Comparing the results obtained with all three methods mentioned above, dynamic constraint aggregation, column generation and Tabu search, shows the effectiveness of this newly adopted approach. For the largest instances, dynamic constraint aggregation coupled with a Tabu algorithm yielded a 98% reduction of the computing times when compared with traditional column generation and better quality solutions with regards to the standardization of the produced schedules. Also, it produces better quality solutions than those generated by a Tabu search algorithm in same computation times. Finally, the results show that a metaheuristic twinned with an exact method offers a good compromise for the problem considered.

# TABLE DES MATIÈRES

<b>DÉDICACE .....</b>	<b>iv</b>
<b>REMERCIEMENTS.....</b>	<b>v</b>
<b>RÉSUMÉ.....</b>	<b>vi</b>
<b>ABSTRACT .....</b>	<b>viii</b>
<b>TABLE DES MATIÈRES.....</b>	<b>x</b>
<b>LSITE DES TABLEAUX .....</b>	<b>xiv</b>
<b>LISTE DES FIGURES .....</b>	<b>xvi</b>
<b>CHAPITRE 1 : INTRODUCTION.....</b>	<b>1</b>
1.1 Terminologie .....	2
1.2 Étapes de la planification des opérations en transport aérien.....	3
1.2.1 Génération des vols .....	3
1.2.2 Assignation de la flotte .....	4
1.2.3 Construction des rotations d'avions .....	4
1.2.4 Construction des rotations d'équipages .....	4
1.2.5 Élaboration des horaires mensuels.....	5

1.3	Définition détaillée du problème de construction des horaires .....	6
1.4	Objectifs du mémoire .....	9
1.5	Structure du mémoire.....	10

## **CHAPITRE 2 : REVUE DE LITTÉRATURE ..... 11**

2.1	Présentation du modèle général .....	11
2.2	Modes de construction d'horaires .....	12
2.2.1	Bidline .....	12
2.2.2	Rostering.....	13
2.2.3	Preferential Bidding.....	13
2.3	Méthodes de résolution.....	14
2.3.1	Heuristiques .....	15
2.3.2	La génération a priori des colonnes .....	17
2.3.3	La génération de colonnes.....	18
2.3.4	L'agrégation dynamique de contraintes .....	20
2.3.5	La programmation par contraintes .....	21

## **CHAPITRE 3 : ALGORITHME TABOU ..... 22**

3.1	Coloration de graphes.....	22
3.1.1	Définition.....	22
3.1.2	Les heuristiques gloutonnes .....	23
3.1.3	Les méthodes de recherche locale .....	24
3.2	L'algorithme Tabou .....	24
3.2.1	Définition.....	24
3.2.2	Techniques d'implémentation .....	25
3.2.3	Détails de l'algorithme.....	26
3.3	Présentation de l'algorithme développé .....	28

3.3.1	Introduction .....	28
3.3.2	L'espace des solutions.....	28
3.3.3	Solution initiale.....	29
3.3.4	La fonction objectif .....	30
3.3.5	Le voisinage d'une solution.....	34
3.3.6	Le contenu de la liste Tabou et sa longueur .....	35
3.3.7	Structure de l'algorithme .....	36
3.4	Limites de l'approche Tabou .....	38

## **CHAPITRE 4 : GÉNÉRATION DE COLONNES ET AGRÉGATION DYNAMIQUE DE CONTRAINTES..... 40**

4.1	Approche classique de génération de colonnes.....	40
4.1.1	Concepts de base .....	40
4.1.2	Applications à la construction d'horaires pour les compagnies aériennes.....	43
4.1.3	Spécialisation au problème d'horaires mensuels des membres d'équipages .....	44
4.1.4	Stratégies de branchement et d'accélération .....	55
4.1.5	Limites de la génération de colonnes.....	56
4.2	L'agrégation dynamique de contraintes .....	57
4.2.1	Introduction .....	57
4.2.2	Présentation de la méthode .....	58
4.2.3	Spécialisation au problème d'horaires mensuels des membres d'équipages .....	62

## **CHAPITRE 5 : EXPÉRIMENTATIONS NUMÉRIQUES ..... 67**

5.1	Résultats de l'algorithme Tabou.....	67
-----	--------------------------------------	----

5.1.1	Approche pour trouver une solution initiale .....	68
5.1.2	Approche en tant que méthode de résolution .....	71
5.2	Résultats de la génération de colonnes .....	73
5.3	Résultats de l'agrégation dynamique de contraintes.....	76
5.4	Comparaison des trois méthodes.....	79
<b>CHAPITRE 6 : CONCLUSION .....</b>		<b>83</b>
<b>RÉFÉRENCES.....</b>		<b>85</b>

## LSITE DES TABLEAUX

Tableau 4.1 : Consommation des ressources .....	51
Tableau 4.2 : Plages de validité des ressources .....	52
Tableau 4.3 : Exemple des plages de validité des ressources.....	55
Tableau 4.4 : Plages de validité des ressources .....	65
Tableau 4.5 : Consommation des ressources .....	66
Tableau 5.1 : Algorithme Tabou pour solution initiale - temps de calcul.....	68
Tableau 5.2 : Algorithme Tabou pour solution initiale – statistiques sur le nombre de crédits.....	69
Tableau 5.3 : Algorithme Tabou pour solution initiale – statistiques sur les jours de congé .....	70
Tableau 5.4 : Algorithme Tabou seul - temps de calcul.....	71
Tableau 5.5 : Algorithme Tabou seul – statistiques sur le nombre de crédits....	72
Tableau 5.6 : Algorithme Tabou seul – statistiques sur le nombre de jours de congé .....	73

Tableau 5.7 : Génération de colonnes – temps de calcul.....	74
Tableau 5.8 : Génération de colonnes – statistiques sur le nombre de crédits .	75
Tableau 5.9 : Génération de colonnes – statistiques sur le nombre de jours de congé .....	75
Tableau 5.10 : Agrégation dynamique de contraintes – temps de calcul .....	77
Tableau 5.11 : Agrégation dynamique de contraintes – statistiques sur le nombre de crédits.....	78
Tableau 5.12 : Agrégation dynamique de contraintes – statistiques sur le nombre de jours de congé.....	79
Tableau 5.13 : Étude comparative des trois approches de résolution .....	80
Tableau 5.14 : Gain de la combinaison de la méthode Tabou associée à l'agrégation dynamique vs génération de colonne .....	81



## LISTE DES FIGURES

Figure 1.1 : Planification des opérations pour une compagnie aérienne .....	6
Figure 4.1 : L'algorithme de génération de colonnes.....	43
Figure 4.2 : Exemple de réseau utilisé pour la génération de colonnes .....	49
Figure 4.3 : Exemple de réseau utilisé pour l'agrégation dynamique de contraintes .....	64

# CHAPITRE 1 : INTRODUCTION

Les dernières décennies ont été marquées par une croissance considérable du trafic aérien commercial. La tâche des planificateurs devant bâtir des horaires pour les avions et le personnel de bord est donc devenue de plus en plus ardue. Que ce soit dans le domaine du transport aérien, du transport routier ou tout simplement pour les employés d'une grande entreprise, la création d'horaires optimaux a toujours fait l'objet d'intenses recherches.

Dans ce mémoire, nous nous intéresserons à la création d'horaires des membres d'équipages des compagnies aériennes commerciales. Ce cas de figure fournit un contexte pour examiner plusieurs des éléments communs à tous les problèmes de construction d'horaires. De plus, il est un problème de planification typique dans le sens où les vols ne sont changés que sur une base au plus mensuelle; pour la compagnie AIR CANADA, cette planification se fait deux fois par an et des changements hebdomadaires sont effectués pour des fins d'ajustement. D'un autre côté, les membres d'équipages reçoivent des salaires sensiblement plus élevés que le personnel équivalent en d'autres modes de transport; l'épargne liée à un programme amélioré d'équipage peut donc être tout à fait significative. Mentionnons aussi que le grand nombre de règles restrictives qui sont exigées par les agences normalisant le transport aérien et les syndicats limite l'ensemble de solutions réalisables, rendant la résolution du problème de plus en plus dure. Pour toutes ces raisons, la construction des horaires d'équipage des compagnies aériennes n'a cessé d'être à l'avant-garde dans le domaine de la recherche industrielle et académique.

## 1.1 Terminologie

Voici une série de termes et définitions communément utilisés dans le domaine de la planification aérienne :

- Une *station* représente un aéroport donné;
- Une *base* est une station à laquelle on associe des membres d'équipage. Une compagnie peut avoir plusieurs bases mais un équipage ne peut être assigné qu'à une seule base qui correspond généralement au lieu de résidence;
- Un *segment de vol* est une portion d'un vol sans escale;
- Une *connexion* est une courte période d'inactivité entre deux segments consécutifs où l'équipage est au sol. Une connexion peut inclure un déplacement par navette;
- Un *service de vol* est une séquence de segments séparés par des connexions qui sont effectuées par le même équipage durant une journée de travail;
- Un *repos* est une période entre deux services de vol durant laquelle l'équipage doit se rendre à un hôtel;
- Une *rotation d'équipage* est une suite alternée de services de vol et de repos entre un départ et un retour à la base d'assignation de l'équipage;
- L'horaire mensuel pour un employé est une suite de rotations, de congés et d'entraînements sur un horizon d'un mois;
- Le personnel technique pour un segment de vol donné est composé du pilote, du copilote et éventuellement de l'ingénieur de bord;
- Le personnel commercial pour un vol donné est composé de l'ensemble des agents de bord ainsi qu'un directeur de bord;
- Une *rotation d'avion* désigne une séquence de segments de vol et d'entretiens qui sont assignés à un appareil spécifique;

- Une *flotte* est l'ensemble de certains types d'appareils appartenant à une compagnie aérienne. Une compagnie peut posséder plusieurs flottes qui contiennent différents types d'appareils. Par exemple, une flotte peut contenir des 747, 757, 727 et des DC8.

## **1.2 Étapes de la planification des opérations en transport aérien**

La planification des opérations en transport aérien a pour but de maximiser les revenus et les profits de la compagnie. Pour estimer les profits, il faut évaluer tous les coûts opérationnels ainsi que les coûts de maintenance. Il s'avère que les salaires des membres d'équipages représentent la deuxième dépense la plus importante après le mazout pour toute compagnie aérienne. Pour évaluer cette dépense, il faut construire des horaires pour les membres d'équipage. Le problème global étant de très grande taille, il ne peut être pris en considération d'un seul coup à cause de la complexité du processus. Dans ce sens, la planification aérienne a été découpée à 5 étapes dont chacune est traitée séparément pour des fins de simplification.

### **1.2.1 Génération des vols**

Cette phase consiste à confectionner les horaires de tous les vols de la compagnie. En d'autres termes, il s'agit d'établir une liste de vols devant être effectués dans une certaine période donnée tout en maximisant les revenus potentiels. L'élaboration de ces vols se fait en fonction des prévisions, des parts de marché, de la concurrence et des intervalles de temps de décollage et d'atterrissage alloués à la compagnie dans différents aéroports. Ce problème incombe en majeure partie au département de marketing de la compagnie aérienne.

### **1.2.2 Assignment de la flotte**

Au cours de cette phase, des types d'avion sont affectés aux vols confectionnés. L'enjeu est de déterminer le type d'avion qui sera le plus adapté pour couvrir chacun des vols tout en maximisant le profit total. Il faut toutefois s'assurer que la construction des rotations d'avions est réalisable et que certaines restrictions sont prises en considération. Parmi ces restrictions, on pourrait citer la limitation du nombre d'appareils disponibles pour la compagnie ou encore l'interdiction de quelques types d'avions sur certains segments de vol.

### **1.2.3 Construction des rotations d'avions**

Durant cette phase, un itinéraire est construit pour chaque appareil de la compagnie aérienne en respectant les assignations confectionnées lors de l'étape précédente et en tenant compte entre autres des vérifications mécaniques nécessaires. Un des objectifs qui peut être poursuivi pour cette étape est de maximiser le nombre de passagers réalisant des connexions sans changement d'avion.

### **1.2.4 Construction des rotations d'équipages**

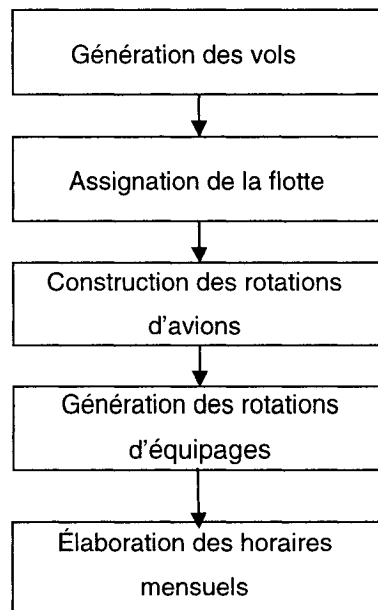
L'élaboration des rotations d'équipages consiste à construire un ensemble de rotations à coût minimum permettant de couvrir tous les vols à effectuer pour un horizon donné. Les rotations doivent respecter les règles des conventions collectives et les lois gouvernementales concernant le transport. Il faut aussi tenir compte des contraintes de répartition des heures créditées entre les différentes bases de la compagnie. Les résultats du problème d'assignation des types permettent de séparer le problème des rotations par flotte d'avions pour le personnel commercial et par type d'avions pour le personnel technique. Les résultats du problème de construction des rotations d'avions définissent

aussi les connexions sans changement d'avion qui peuvent mener à des connexions courtes pour les équipages.

### **1.2.5 Élaboration des horaires mensuels**

Une fois les rotations construites, cette dernière phase consiste à affecter à chaque membre d'équipage une suite de rotations, de journées de repos, de vacances, de périodes d'entraînement et d'examens médicaux. L'objectif de ce problème est de construire des blocs mensuels qui couvrent toutes les rotations tout en respectant les règles des conventions collectives, des normes de sécurité et de la politique de la compagnie. Ce problème est aussi séparable par flotte d'avions pour le personnel commercial et par types d'avions pour le personnel technique. Il faudrait cependant noter qu'il existe principalement trois modes pour l'élaboration des horaires mensuels en l'occurrence le *bidline*, le *rostering* et le *preferential bidding*. Ces modes seront détaillés dans le chapitre 2.

La figure 1.1 résume le processus séquentiel de la planification des opérations décrit précédemment. Les flèches illustrent la séquence des étapes, mais aussi, le fait que les résultats d'une étape influencent la définition des problèmes aux étapes suivantes. Par conséquent, le problème d'horaires mensuels qui nous intéresse plus particulièrement dans ce mémoire se retrouve à la toute fin du processus de planification et est fortement influencé par toutes les étapes précédentes.



**Figure 1.1 : Planification des opérations pour une compagnie aérienne**

### **1.3 Définition détaillée du problème de construction des horaires**

Comme on l'a décrit précédemment, la construction d'horaires mensuels se réduit à trouver le sous-ensemble des horaires qui couvrent, à coût minimal, toutes les rotations générées à l'étape précédente de la planification aérienne. On note que le nombre d'employés et d'activités à assigner mensuellement est généralement très élevé ce qui donne naissance à des problèmes de très grande taille. Pour pallier cela, il est possible de diviser le problème global en problèmes de plus petite taille selon les différentes fonctions des membres d'équipages. Parmi les décompositions possibles, on peut citer le cas du personnel technique et du personnel commercial. En effet, l'affectation des uns à une rotation ne dépend pas des autres et vice-versa et ce surtout à cause des règles de convention collective différentes. On pourrait subdiviser le problème

encore en types d'avions pour le personnel technique et par flotte d'avions pour le personnel commercial. Il est également possible de considérer le problème global en fonction des différentes bases de la compagnie aérienne. Cependant, la taille de certains problèmes, notamment celui du personnel commercial, demeure très grande. On note que les contraintes dont il faudrait tenir compte lors de la construction des horaires se résument principalement à deux familles :

- Les contraintes locales : Les paramètres régissant ces règles changent selon la compagnie aérienne.
  - *Crédits de vol* : Les crédits de vol sont les heures de vols et les crédits pour des activités comme les périodes d'entraînement, réunions etc. La somme de ces crédits doit se situer dans un intervalle prédéfini ;
  - Jours travaillés : Le nombre de jours travaillés consécutifs entre deux périodes de repos ne doit pas dépasser une certaine valeur seuil ;
  - Jours de congé : Chaque employé doit avoir un nombre minimum de jours de congé pendant un mois de travail. Un jour de congé est un jour du calendrier de minuit à minuit ;
  - Indice de fatigue: 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 une valeur seuil au début de n'importe quelle rotation et ne doit jamais excéder une valeur maximale au cours d'un mois.
- Les contraintes globales : Elles concernent la composition de l'ensemble des horaires construits et elles sont regroupées principalement en trois catégories.



- Partition : Le nombre d'employés nécessaire pour une rotation donnée doit être satisfait;
- Choix des horaires : A chaque employé, on doit attribuer un horaire;
- Composition des équipages : Chaque équipage a un certain nombre de requêtes demandées que l'ensemble des membres doit satisfaire. Ces qualifications dépendent généralement du type d'appareil ou de la flotte, de la langue et des visas nécessaires pour les pays visités.

L'objectif du problème de construction d'horaires est la minimisation du coût des horaires pour couvrir l'ensemble des rotations. Éventuellement, l'objectif pourrait inclure la maximisation de la satisfaction des employés et ce en utilisant un certain nombre de règles qui seront détaillées dans le chapitre 2.

Dans le cas de ce mémoire, un certain nombre de contraintes ne seront pas considérés lors du processus de résolution et ce pour des fins académiques. La construction des horaires se fera anonymement et ce sans tenir compte de certains paramètres tels que les activités pré-assignées et les qualifications des employés. De plus, nous nous concentrerons sur les horaires des pilotes (soit un pilote par rotation). Voici une description du problème considéré pendant la suite du présent travail.

➤ Données :

- Horizon de planification (date de début, date de fin);
- Liste de rotations à assigner aux employés, chaque rotation ayant une valeur de crédit;
- Détail des employés.

➤ Contraintes :

- Le nombre de crédits pour chaque employé doit être compris dans un intervalle bien déterminé;
- Chaque horaire doit avoir un nombre minimum de jours de congé;
- Chaque rotation assignée doit être suivie d'un repos post-courrier prédéfini;
- Un employé ne peut travailler plus qu'un certain nombre de jours consécutifs sans un jour de congé.

➤ **Objectif :**

Uniformiser les horaires des employés en minimisant les écarts entre le nombre de jours de congé, ainsi que le nombre de crédits total dans le mois de chacun des et les moyennes de ces valeurs pour tous les employés.

## **1.4 Objectifs du mémoire**

Le défi actuel relatif à la construction d'horaires mensuels pour les compagnies aériennes est de résoudre des problèmes de plus en plus grands tout en couvrant toutes les rotations d'équipages et en satisfaisant toutes les contraintes du problème. Il a été montré que la taille des problèmes augmente exponentiellement avec le nombre de contraintes; on a évalué le nombre d'horaires résultants de 700 rotations à 100 millions. Dans ce sens, le projet de maîtrise traite de la construction d'horaires mensuels de membres d'équipage d'une compagnie aérienne pour des problèmes de très grande taille. Les horaires mensuels sont construits dans un contexte d'équité. L'objectif de ce projet est de développer une méthode qui permettra de résoudre des problèmes de grande taille plus rapidement que les algorithmes existants. Pour atteindre cet objectif, une nouvelle approche est adoptée pour la résolution du problème. En effet, une récente méthode, en l'occurrence, l'agrégation dynamique de contraintes est choisie pour l'élaboration des horaires personnels. Cette méthode est très prometteuse mais elle requiert l'utilisation d'une solution

initiale de bonne qualité pour entamer le processus de résolution. Le choix s'est porté sur une métaheuristique Tabou pour le développement de la solution initiale. En effet, le problème de construction des horaires étant un problème de partitionnement d'ensembles, il est réductible en un problème de coloration de graphes. Or, l'approche Tabou a prouvé son efficacité dans la résolution de ce genre de problèmes.

Les contributions du mémoire sont donc de :

1. Développer une méthode Tabou pour le problème de construction d'horaires mensuels dans le domaine aérien;
2. Appliquer la méthode d'agrégation dynamique de contraintes pour ce problème.

## **1.5 Structure du mémoire**

Suite à cette introduction, le contenu des prochains chapitres est le suivant. Le chapitre 2 présente une revue de littérature sur la construction d'horaires de membres d'équipages aériens. Il abordera les différentes approches adoptées pour la résolution du problème en question et discute de la pertinence de chacune des approches. Le chapitre 3 présente l'implémentation de l'approche métaheuristique de la solution adoptée, tout particulièrement les différents paramètres de l'algorithme Tabou développé ainsi que les mécanismes de recherche locale utilisés. Le chapitre 4 présente la méthode de génération de colonnes et l'agrégation dynamique de contraintes. Il introduit le mécanisme de résolution et présente les résultats antérieurs à cette nouvelle méthode. Ensuite, on rapporte les résultats numériques en analysant les trois approches adoptées et en comparant la qualité des solutions obtenues dans chacun des cas. Finalement, le dernier chapitre conclut le mémoire en rappelant les contributions du projet et en soulignant les voies de recherche futures.

## CHAPITRE 2 : REVUE DE LITTÉRATURE

### 2.1 Présentation du modèle général

Typiquement, la construction des horaires est formulée comme un problème de partitionnement d'ensemble. Le résoudre revient à trouver un sous-ensemble d'horaires mensuels à coût minimal de telle façon que chaque rotation soit couverte par exactement le nombre d'employés requis.

Les paramètres et les variables nécessaires à la formulation du problème sont les suivants:

- $\Omega_k$  est l'ensemble des horaires admissibles pour l'employé  $k$ ;
- $m$  est le nombre d'employés;
- $n$  est le nombre de rotations à affecter;
- $c_j$  est le coût de l'horaire  $j$ ;
- $a_{ij}$  un paramètre qui est égal à 1 si la rotation  $i$  est comprise dans l'horaire  $j$  et 0 autrement;
- $b_i$  est le nombre d'employés requis pour la rotation  $i$ ;
- $x_j^k$  une variable de décision qui est égale à 1 si l'horaire  $j$  est choisi pour l'employé  $k$  et 0 autrement.

Le problème de construction d'horaires se formule alors comme suit :

$$\text{Minimiser } \sum_{k=1}^m \sum_{j \in \Omega^k} c_j x_j^k \quad (2.1)$$

Sujet aux contraintes :

$$\sum_{k=1}^m \sum_{j \in \Omega^k} a_{ij} x_j^k = b_i, \quad \forall i = 1, 2, \dots, n \quad (2.2)$$

$$\sum_{j \in \Omega_k} x_j^k = 1, \quad \forall k = 1, 2, \dots, m \quad (2.3)$$

$$x_j^k \in \{0, 1\}, \quad \forall j \in \Omega^k \text{ et } \forall k = 1, 2, \dots, m \quad (2.4)$$

L'objectif (2.1) vise à minimiser les coûts totaux alors que les contraintes (2.2) assurent la couverture de toutes les rotations par le nombre d'employés nécessaire à chacune. En dernier lieu, les contraintes (2.3) et (2.4) s'assurent de la validité et de l'unicité de l'horaire de chaque employé.

Il faut noter que cette formulation requiert l'énumération de tous les horaires possibles. Énumérer tous les horaires peut être très difficile et ce à cause des règles qui régissent le transport aérien, mais surtout à cause du très grand nombre d'horaires potentiels. En effet, Housos et al. (1997) ont prouvé que ce problème est NP-complet.

## 2.2 Modes de construction d'horaires

Il y a principalement trois approches qui sont admises pour la construction des horaires au sein des compagnies aériennes en l'occurrence le *bidline*, le *rostering* et le *preferential bidding*.

### 2.2.1 Bidline

Le *bidline* consiste en une construction anonyme des horaires mensuels et ce sans tenir compte des préférences des employés, mais plutôt en minimisant les ressources nécessaires à la couverture totale de toutes les rotations. Lors du processus de choix, les employés misent sur les horaires construits et l'affectation se fait selon un certain nombre de critères tel que l'ancienneté. Cette méthode était adoptée par plusieurs compagnies telles que American Airlines (Russell 1989) ou encore British Airways (Wilson 1981). Ce mode de

construction est considéré comme le plus simple de tous les autres modes de construction d'horaires car il ne tient pas compte des préférences de chacun des employés lors de l'affectation des rotations.

### **2.2.2 Rostering**

Le *rostering* introduit la notion d'horaires personnalisés. Le *rostering* construit des horaires mensuels en tenant compte des activités pré-assignées des employés. Ce mode de construction d'horaires consiste à attribuer des poids aux activités qui doivent être incluses dans l'horaire d'un employé donné. Une tâche 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) ou encore une activité de réserve. Les poids sur les activités peuvent varier selon des critères établis par la compagnie aérienne. Notons que ce mode de construction d'horaires est utilisé principalement par les compagnies européennes telles que Air France (Gontier 1985 et Gamache et al. 1999), Alitalia (Fedrici et Paschina 1992), Lufthansa (Glanert 1984). Ryan (1992) a aussi implanté cette méthode au sein d'Air New Zealand.

### **2.2.3 Preferential Bidding**

En continuant dans la même approche, le *preferential bidding* adopte un choix de préférences plus élaboré que le *rostering*. En effet, les horaires personnalisés sont construits en fonction de l'ensemble des demandes pondérées, appelées "*bids*", reflétant les préférences de chaque membre d'équipage de même qu'un ensemble d'activités pré-assignées devant être incluses dans son horaire.

Il existe principalement deux types de variantes pour ce mode de construction : *preferential bidding* dans un cadre d'ancienneté ou dans un cadre d'équité.

- Dans un cadre d'ancienneté, la construction des horaires se fait en favorisant les préférences exprimées par l'employé le plus senior et ce tout en s'assurant qu'il est possible de couvrir toutes les rotations restantes. La principale lacune de cette méthode est que les membres les plus juniors se voient attribuer des horaires qui ne satisfont pas les requêtes qu'ils ont communiquées et ce du fait de l'attribution de l'horaire sur une base de séniorité stricte. Cette méthode a été implantée par Gamache et al. (1998) chez Air Canada.
- Dans un contexte d'équité, chaque membre d'équipage est traité de façon impartiale, sans tenir compte de son ancienneté. Les horaires sont donc fabriqués de façon à ce que, en moyenne, tous les employés soient satisfaits équitablement. Puisqu'il est impossible de satisfaire tous les employés de façon identique pour un mois donné, l'historique des horaires précédents est conservé pour chaque employé. De cette façon, il est possible, pour le mois en cours, de construire des horaires qui avantagent les employés ayant obtenus une satisfaction inférieure à la moyenne dans les mois précédents et vice-versa. En utilisant cette technique, la satisfaction des membres d'équipage est plus ou moins identique sur une longue période de temps.

## 2.3 Méthodes de résolution

Due à la difficulté du problème, il est impossible d'énumérer exhaustivement toutes les solutions du modèle présenté dans la section 2.1 et ce afin de le résoudre optimalement. C'est ce qui a poussé les chercheurs à développer un certain nombre d'algorithmes pour parvenir à des solutions pour le problème de construction des horaires. On note que dans la littérature, il existe un certain nombre d'articles de synthèse qui définissent le problème de construction d'horaires et présentent les approches adoptées pour la résolution du problème

tels que Desaulniers et al. (1998a), Barnhart et al. (1999a) et Klabjan et al. (2005). On pourrait grouper les méthodes de résolution développées dans la littérature en trois catégories :

- Les heuristiques;
- Les méthodes qui requièrent la génération a priori des colonnes du modèle de partitionnement ;
- Les approches de génération de colonnes;
- La programmation par contraintes.

### **2.3.1 Heuristiques**

Due à la simplicité d'implémentation et au large pouvoir constructif des heuristiques, les chercheurs ont pu incorporer deux étapes de la planification aérienne, en l'occurrence la création des rotations et la construction des horaires mensuels. Dans ce cadre, Gershkoff (1989) a développé une heuristique basée sur une liste Tabou pour la résolution du problème de construction d'horaires. Il a procédé à une décomposition du problème en une série de sous-problèmes et ce pour pouvoir trouver une solution locale ayant un coût minimum. À chaque itération, l'optimiseur considère un sous-ensemble de rotations dans la solution courante et énumère toutes les différentes combinaisons possibles créées à partir des vols composants les rotations en question. Par la suite, l'algorithme procède à la résolution d'un problème en nombres entiers en utilisant une méthode de séparation et d'évaluation progressive. Souvent, ces problèmes sont de petite taille et peuvent être résolus rapidement. À chaque étape, le dernier sous-problème résolu est stocké dans une liste Tabou permanente et ce pour éviter de résoudre des sous-problèmes précédemment visités. Cette heuristique a prouvé son efficience suite à une implantation chez American Airlines. En effet, l'introduction avec succès de l'heuristique mise en place a engendré des gains substantiels chez le



gérant du transport aérien. Il faut cependant noter que l'avancée technologique dans le domaine informatique a permis de résoudre de plus grands sous-problèmes pour trouver des solutions de plus en plus près de la solution globalement optimale surtout pour des problèmes de petite taille comportant environ 200 vols par jour. Cette heuristique présente toutefois un désavantage majeur : l'algorithme ne peut éviter des optimums locaux et ce en permettant à l'heuristique d'entreprendre des mouvements dans le voisinage, même s'ils sont vers des solutions de moindre qualité.

Baker et al. (1979), Ball et al. (1985), Levine (1996), Christou et al. (1999) et Ahuja et al. (2002) ont adopté une approche distincte de Gershkoff (1989). Ball et al. (1985) ont développé une solution se basant sur un modèle de graphe. L'optimiseur construit une solution puis essaye d'améliorer le coût par des échanges dans la suite des rotations. Les problèmes considérés variaient de 328 à 1058 vols et les temps de résolution ne dépassaient pas les 5 minutes. Quant à Baker et al. (1979), ils débutent la recherche par une première solution réalisable construite d'une façon gloutonne. Pour trouver les optimums locaux, l'algorithme considéré utilise des changements 2-opt pour se déplacer dans le voisinage des solutions trouvées. Cette méthode a été aussi utilisée par Ahuja et al. (2002). Cependant, ils ont utilisé un voisinage beaucoup plus grand car ils ont considéré un nombre de vols plus important. Cet algorithme a donné des gains substantiels chez American Airlines estimés à 25 millions de dollars par an.

L'algorithme génétique hybride utilisé par Levine (1996) semble être la première métaheuristique utilisée pour la résolution du problème de construction d'horaires dans le transport aérien. Il faudrait cependant attirer l'attention sur le fait qu'il suppose que tous les horaires réalisables sont énumérés avant l'utilisation de l'algorithme génétique. Levine semble ignorer le potentiel constructif puissant d'une métaheuristique surtout qu'il a utilisé une descente

pour améliorer la qualité des descendants après le croisement des parents. Cette lacune a été palliée dans la recherche de Christou (1999) et l'implantation de son algorithme a permis à Delta Airlines d'épargner des montants considérables en dépenses pour le personnel. Cette recherche prouve qu'un algorithme génétique combiné avec une recherche locale, fournit une excellente approche pour générer des solutions de très bonne qualité.

Finalement, une des applications les plus réussies dans l'utilisation des métaheuristiques est l'algorithme Tabou à voisinage variable développé par Combs (2002) dans le cadre de sa thèse de doctorat. L'application de l'algorithme est faite au sein de la Air Force Institute Of Technology et ce pour son programme d'avions cargo. Par conséquent, étant donné le domaine d'application choisi, il existe quelques différences dans les contraintes du modèle de la section 2.1 mais le contexte général reste le même.

Bien que les heuristiques aient été couronnées de succès dans la pratique, elles garantissent seulement la convergence aux optimums locaux. Pour surmonter cette limitation, les chercheurs ont créé d'autres méthodes d'optimisation pour la résolution du problème.

### **2.3.2 La génération a priori des colonnes**

Cette méthode est basée sur la connaissance a priori de toutes les colonnes du problème avant d'entamer le processus de résolution. En d'autres termes, tous les horaires réalisables à partir de l'ensemble des rotations fournies doivent être énumérés avant de commencer à générer les horaires mensuels. Cependant, cette approche a été principalement utilisée pour le problème de la création des rotations ; notamment dans les travaux de Housos et al. (1997) où ils adoptent une approche similaire à celle utilisée par Gershkoff (1989). En effet, le modèle initial expliqué dans la section 2.1 a été décomposé

en un grand nombre de petits modèles de partitionnement d'ensemble qui ont été résolus en utilisant une méthode d'énumération implicite. Mais aussi, plusieurs chercheurs dont Graves et al. (1993) ont su tirer avantage des avancées technologiques pour pouvoir résoudre des problèmes de partitionnement de grande taille (1700 vols par jour). L'approche de résolution adoptée possède deux facettes : un générateur et un optimiseur. Le générateur crée l'ensemble de toutes les colonnes compatibles au sous-problème choisi et l'optimiseur effectue le choix à partir de l'ensemble créé pour atteindre un meilleur coût total. Plusieurs autres approches ont été développées afin de résoudre le problème de construction d'horaires notamment pour le cas du rostering. Parmi celles-ci, Ryan (1992) propose une approche qui consiste à générer a priori des horaires grâce à des règles heuristiques et ensuite à résoudre un problème de recouvrement sur l'ensemble des horaires générés.

### **2.3.3 La génération de colonnes**

Cette approche permet de résoudre les problèmes de la construction des rotations d'équipages et de l'élaboration d'horaires mensuels relatifs au processus de planification aérienne correspondant à la figure 1.1. En effet, Desaulniers et al. (1997a) ainsi que Barnhart et al. (1998b) ont démontré l'efficacité de cette approche pour le problème de construction des rotations respectivement pour Air France et une compagnie Nord-américaine de long courrier.

L'objectif du problème traité dans ce mémoire est de trouver un ensemble d'horaires valides à coût minimum permettant de couvrir toutes les rotations exactement une fois. Pour des fins de fonctionnalité, la génération de colonne introduit une formulation en réseau du problème. Le réseau est formé par toutes les rotations devant être couvertes, un lien est créé entre deux rotations si et seulement si les contraintes temporelles et organisationnelles du problème le

permettent. De ce fait, les horaires construits correspondent à des chemins dans un réseau. Ces horaires sont dits admissibles ou valides lorsqu'ils respectent la réglementation appropriée.

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 (2.1)-(2.4) restreint à un sous-ensemble d'horaires, appelé problème maître restreint. La différence entre ces approches et les méthodes heuristiques proposées dans Gershkoff (1989) et Graves et al. (1993) pour le problème de construction des 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 du 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 solution à partir d'un sous-ensemble d'horaires générés a priori.

Plusieurs chercheurs, notamment au GERAD, ont prouvé l'efficacité de cette approche de génération de colonnes pour le problème de construction d'horaires. Ainsi, Gamache et al. (1998) a appliqué cette méthode au problème de *preferential bidding* avec séniorité stricte chez Air Canada ce qui a donné des gains substantiels. Ou encore, Gamache et al. (1999) proposent une approche par génération de colonnes dans laquelle des problèmes de plus court chemin avec contraintes de ressource sont solutionnés afin de générer les horaires mensuels selon la méthode de *rostering*. Notons que cette méthode est restée à la fine pointe de la recherche dans ce domaine et des améliorations à

cette approche ne cessent de paraître notamment dans les travaux de Achour (2005).

D'un autre côté, la génération de colonnes présente un certain nombre d'inconvénients qui se résument surtout à son incapacité à résoudre des problèmes de grande taille. Due à l'augmentation exponentielle des temps de calculs, la dégénérescence que présente la méthode de génération de colonnes rend le problème difficile à résoudre optimalement au-delà d'un nombre avoisinant de 300 employés et 1500 rotations. C'est ce qui a poussé les chercheurs au GERAD à développer une nouvelle méthode de résolution, en l'occurrence l'agrégation dynamique de contraintes.

#### **2.3.4 L'agrégation dynamique de contraintes**

Cette méthode a été introduite par les travaux d'El Hallaoui et al. (2005 et 2006) où ils présentent un certain nombre de résultats très prometteurs pour cette nouvelle approche. En effet, cette technique permet de réduire significativement la taille du problème maître aussi bien que la dégénérescence et le temps de résolution, particulièrement pour les problèmes de grande taille. Le mécanisme de résolution est en fait une alternance d'agrégations et de désagrégations de plusieurs tâches de façon à diminuer la taille du problème à résoudre lors de l'itération courante de la génération de colonnes. Cette méthode est très prometteuse et donne des résultats nettement meilleurs que la génération de colonnes. Cependant, l'agrégation dynamique de contraintes nécessite une solution initiale d'assez bonne qualité en vue d'obtenir des résultats meilleurs que les autres méthodes basées sur l'utilisation de la génération des colonnes. Pour ce mémoire, le choix s'est porté sur une métaheuristique Tabou pour le développement de cette solution initiale et le détail de cette méthode est présenté au chapitre suivant.

### **2.3.5 La programmation par contraintes**

Dans les travaux de Fahle et al (2002), les auteurs présentent une nouvelle approche pour la résolution du problème. Cette approche combine génération de colonnes et programmation par contraintes qui est utilisée pour résoudre le sous-problème comme un problème de satisfaction de contraintes. En effet, en plus des contraintes habituelles du sous-problème, une contrainte additionnelle englobant un algorithme de plus court chemin est introduite afin de produire des colonnes ayant un coût réduit négatif. Cette contrainte réduit l'espace de recherche du sous-problème de manière significative. Les réductions dans le domaine de résolution résultantes sont propagées aux autres contraintes réduisant davantage l'espace de recherche.

## CHAPITRE 3 : ALGORITHME TABOU

Dans cette section, nous allons présenter la première étape du processus de résolution choisi, en l'occurrence, l'élaboration d'une solution initiale. En effet, le choix de l'utilisation d'une métaheuristique pour cette étape a été dicté par plusieurs paramètres. Tenant compte de la nature du problème et du temps de mise en marche de l'algorithme de l'agrégation dynamique de contraintes, il fallait obtenir une solution initiale de bonne qualité. Puisque le problème de construction des horaires est un problème de partitionnement, une approche commune de résolution est de le formuler comme un problème de coloriage de graphe.

### 3.1 Coloration de graphes

Pour se ramener à un problème de coloriage de graphe, nous devons d'abord construire un graphe auxiliaire,  $G = (V, E)$ , tel que chaque rotation est modélisée par un nœud et que deux nœuds sont reliés par une arête si les deux rotations correspondantes ne peuvent être effectuées par le même employé. D'un autre côté, chaque couleur correspond à un employé donné. L'affectation qui satisfait à l'exigence prédéfinie du nombre d'employés sous les contraintes de faisabilité de cette dernière peut être obtenue en coloriant les sommets du graphe  $G$ .

#### 3.1.1 Définition

Étant donné un graphe  $G = (V, E)$ , où  $V$  est l'ensemble des sommets et  $E$  l'ensemble des arêtes, le coloriage des sommets revient à affecter une couleur à chacun d'entre eux, tel que deux sommets adjacents n'aient pas la même

couleur. Un ensemble  $S$  est dit stable s'il ne comprend aucune paire de sommets adjacents. La taille maximale d'un stable dans  $G$  est notée  $\alpha(G)$ . Étant donné un entier  $k$ , un  $k$ -coloriage du graphe  $G$  revient à le colorer en  $k$  couleurs. Si le graphe peut être coloré par  $k$  couleurs, il est dit  $k$ -colorable ; sinon, on essaie généralement de maximiser le nombre de sommets colorés noté  $\alpha_k(G)$ . Le nombre chromatique de  $G$  est le nombre minimum de couleurs pour colorer tous les sommets et il est noté  $\chi(G)$ . La  $k$ -coloration est un problème de décision qui a été démontré NP-complet, de plus, le nombre chromatique d'un graphe est difficile à déterminer. Cependant, on se contente généralement de trouver une borne supérieure  $k$  sur  $\chi(G)$  et à appliquer ainsi des heuristiques pour obtenir de telles solutions acceptables c'est-à-dire des  $k$ -colorations de  $G$ . Nous présenterons ci-dessous deux types d'heuristiques rencontrées dans la littérature.

### 3.1.2 Les heuristiques gloutonnes

Pour cette méthode de résolution, les sommets sont colorés un par un. Il faut noter que le processus ne considère aucune révision d'une couleur déjà allouée à un sommet. Le choix du prochain sommet à colorier dépend du critère glouton. Ce critère peut être statique ou dynamique. Dans le cas statique, un ordre des sommets est établi à l'avance, tandis que pour le cas dynamique, le critère glouton est calculé à chaque itération pour faire le choix du prochain candidat. Voici une présentation des heuristiques gloutonnes les plus connues :

- L'heuristique séquentielle (statique) : Dans ce cas, les sommets sont classés dans un ordre bien déterminé à l'avance, suivant un critère donné, par exemple, le degré du sommet. À chaque itération, le sommet considéré est coloré avec la première couleur légale parmi la liste de toutes les couleurs disponibles;



- L'heuristique de saturation: C'est une des méthodes les plus utilisées dans le cas des heuristiques gloutonnes dynamiques. Comme son nom l'indique, elle est basée sur la saturation des sommets. La saturation d'un sommet donné correspond au nombre de couleurs utilisées par ses voisins immédiats. Le sommet ayant une saturation maximale est choisi à chaque itération.

### **3.1.3 Les méthodes de recherche locale**

Le principe de cette classe de métaheuristiques est très simple. Elle est basée principalement sur la notion de voisinage. Ce type d'algorithme est appliqué aux problèmes d'optimisation avec un ensemble  $S$  de configurations et une fonction de coût  $f$ . On définit aussi une fonction de voisinage et un mécanisme d'exploration de ce dernier. Comme configuration initiale, nous pouvons prendre la solution d'une heuristique gloutonne. À chaque itération, une configuration voisine est choisie en fonction de son coût. Ce processus se poursuit tant qu'on n'a pas atteint un critère d'arrêt fixé à l'avance (nombre maximum d'itérations,...). Comme exemples de méthodes de recherche locale, citons la recherche Tabou et le recuit simulé.

## **3.2 L'algorithme Tabou**

### **3.2.1 Définition**

La recherche Tabou a été introduite par Glover en 1986. Cette méthode est une métaheuristique itérative qualifiée de recherche locale au sens large. L'idée de la recherche Tabou consiste, à partir d'une solution donnée, à explorer le voisinage et à choisir la solution dans ce voisinage qui minimise la fonction objectif. Il est essentiel de noter que cette opération peut conduire à

augmenter la valeur de la fonction : c'est le cas lorsque toutes les solutions du voisinage ont une valeur plus élevée. C'est à partir de ce mécanisme que l'on sort d'un minimum local.

Le risque cependant est qu'à l'étape suivante, on retombe dans le minimum local auquel on vient d'échapper. C'est pourquoi il faut que l'heuristique ait de la mémoire : le mécanisme consiste à interdire (d'où le nom de Tabou) de revenir sur les dernières solutions explorées.

Les solutions déjà explorées sont conservées dans une pile (appelée liste Tabou) d'une taille donnée, qui est un paramètre ajustable de l'heuristique. Cette liste doit conserver des solutions complètes, ce qui dans certains types de problèmes, peut nécessiter l'archivage d'une grande quantité d'informations. Cette difficulté peut être contournée en ne gardant en mémoire que des « portions » de la meilleure solution associées à la valeur de la fonction à minimiser. Il est cependant utile de noter que de nombreuses variantes existent, principalement au niveau de la définition du voisinage et de la manière de gérer la mémoire.

D'un autre côté, la convergence de l'algorithme, ou encore la capacité de l'algorithme à trouver le minimum global si le temps imparti tend vers l'infini est prouvée avec un certain nombre d'hypothèses devant être satisfaites. Mais, la démonstration suppose des conditions strictes rarement présentes en pratique.

### **3.2.2 Techniques d'implémentation**

Comme tout algorithme de résolution, les chercheurs ont développé un certain nombre d'astuces afin d'augmenter l'efficacité de l'algorithme Tabou. Il est cependant utile de souligner que cette liste n'est pas exhaustive et il existe plusieurs autres techniques associées à cette heuristique de recherche Tabou :

- Dans certains cas, le statut Tabou peut être trop restrictif, en d'autres termes, il empêche d'atteindre une solution intéressante. Ou encore, l'algorithme peut cycliser dans le voisinage d'une solution lors de la procédure de recherche, ce qu'il faudrait éviter. Pour pallier ces problèmes, un critère d'aspiration doit être introduit dans l'algorithme. L'exemple le plus simple consiste à enlever le statut Tabou associé à une solution déjà explorée si celle-ci permet d'aboutir à une solution meilleure que toutes les solutions trouvées auparavant;
- Il est également possible d'intensifier la recherche sur des zones qui paraissent "prometteuses", ce qui permet de déceler l'existence d'autres minima locaux et éventuellement la solution optimale plus rapidement;
- Lorsque le processus de recherche parcourt une même région de l'espace des solutions sur une longue période, un processus de diversification peut être entamé en stoppant la recherche et en l'amorçant sur une autre solution générée aléatoirement;
- Afin d'éviter d'avoir une liste Tabou trop grande, il est possible de gérer une liste circulaire de taille fixe dépendamment du type de problème traité, où l'ajout d'un élément entraîne l'effacement du plus ancien élément de la liste;
- Si l'évaluation des voisins requiert un temps de calcul important, l'appel à des listes d'échantillons aléatoires peut diminuer substantiellement les coûts d'évaluation.

### **3.2.3 Détails de l'algorithme**

Afin de mieux visualiser cette technique, voici un pseudo-code de l'algorithme Tabou dans sa version la plus simple qui fait appel à la notation suivante :

- $S$  l'ensemble des toutes les solutions possibles ;
- $i$  le numéro de l'itération courante ;
- $s$  la meilleure solution trouvée à l'itération courante ;
- $s^*$  la meilleure solution trouvée depuis le début du processus de résolution ;
- $f(s)$  la valeur de la fonction objectif pour la solution  $s$  ;
- $N(s,i)$  l'ensemble de toutes les solutions voisines de la solution  $s$  lors de l'itération  $i$ .

#### Étape 1

- Choisir une solution initiale  $s$  dans  $S$  ;
- Poser  $s^* = s$  et  $i = 0$

#### Étape 2

- Poser  $i = i + 1$  et générer un sous-ensemble de solutions parmi  $N(s,i)$  tel que le critère Tabou est respecté ou le critère d'aspiration est applicable.

#### Étape 3

- Soit  $s'$  la meilleure solution parmi l'ensemble généré à l'étape 2;
- Poser  $s = s'$ .

#### Étape 4

- Si  $f(s) \leq f(s^*)$ ;
  - Poser  $s^* = s$ .

#### Étape 5

- Mettre à jour la liste  $T$  et le critère d'aspiration.

#### Étape 6

- Si une condition d'arrêt est atteinte stop;
- Sinon, retourner à l'étape 2.

### **3.3 Présentation de l'algorithme développé**

#### **3.3.1 Introduction**

Le choix de l'algorithme pour le problème d'horaires, s'est porté sur l'utilisation d'une méthode Tabou car cette approche a prouvé son efficacité pour la résolution du problème de coloration de graphes appliqués à la construction d'horaires mensuels notamment dans les travaux de Ouellet (2004) et Gamache et al. (2005). En effet, lors de ces travaux, une nouvelle méthodologie a été développée en vue d'optimiser les appels récursifs à la méthode de génération de colonnes. Cela se fait en déterminant si l'ensemble des rotations non assignés à ce stade du processus de résolution aboutit à une solution faisable. Dans le cas contraire, un processus de retour en arrière est amorcé pour y pallier. Le principal avantage de cette méthode est de minimiser les itérations de génération de colonnes subséquentes à une affectation d'une rotation qui donne lieu à un problème irréalisable.

#### **3.3.2 L'espace des solutions**

L'espace des solutions est composé de graphes entièrement colorés. Cependant lors du processus de résolution, on permet l'existence de conflits

dans la coloration d'une solution donnée et cela se traduit par l'attribution d'une même couleur à deux sommets voisins. Mais, la solution finale, si elle existe n'admet pas d'arêtes conflictuelles. La modélisation choisie est la suivante :

- Les sommets représentent les rotations;
- Les couleurs représentent les employés. Donc chaque couleur est associée à un horaire donné;

La suite de rotations et de jours de repos constitue un horaire mensuel pour un employé donné. En d'autres termes, il suffit de parcourir les sommets (rotations) et ceux qui ont la même couleur seront affectés à la même personne durant le mois en question.

### **3.3.3 Solution initiale**

Le choix de la génération d'une solution initiale s'est porté au début sur la coloration séquentielle du graphe selon un schéma d'élimination parfait. En effet, le graphe  $G$  construit est un graphe d'intervalle car toutes les rotations peuvent se placer sur un axe temporel (du début à la fin de la période de planification). Or cette classe de graphes est incluse par définition aux graphes triangulés possédant un schéma d'élimination parfait qui peut être utilisé pour une coloration du graphe. Cependant, l'utilisation de cette heuristique s'est avérée très coûteuse d'un point de vue du temps de calcul. Donc le choix s'est porté sur une coloration aléatoire du graphe. D'un autre côté, en adaptant l'algorithme Tabou aux instances traitées, la coloration est ajustée rapidement dans les premières itérations. Finalement, le choix de la solution gloutonne aléatoire a donné des résultats très satisfaisants et ce avec des temps de calcul minimes.

### 3.3.4 La fonction objectif

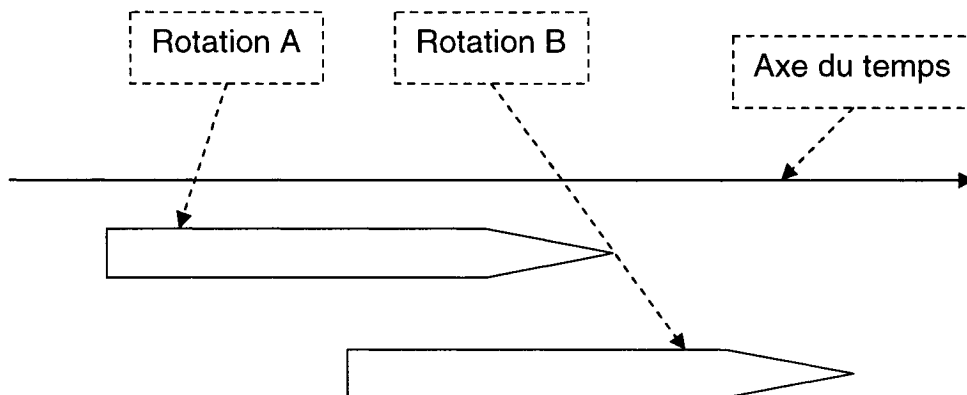
L'objectif primaire de l'algorithme Tabou est de trouver une assignation des rotations qui satisfait aux contraintes du problème décrites précédemment. Le processus de résolution considère par la suite l'uniformisation du nombre de crédits travaillés et du nombre de jours de congé pour chaque employé. Le processus de résolution est donc divisé en deux étapes :

- Recherche d'une solution réalisable;
- Uniformisation des horaires des employés.

Lors du processus de la première étape de résolution, la fonction objectif se compose principalement de 5 parties :

#### 1. Les chevauchements simples ( $F_{\text{Chevau\_simples}}$ ) :

Soit les rotations A et B suivantes :



Puisque le début de la rotation B est compris entre l'heure de début et l'heure de fin de la rotation A, il y a chevauchement simple entre les deux rotations. À cause de contraintes temporelles simples, les rotations A et B ne peuvent appartenir au même horaire et c'est pour cela qu'un tel cas est

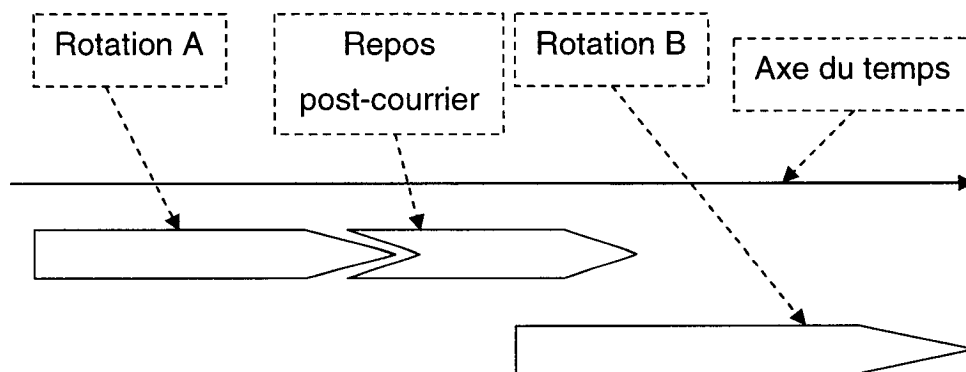
comptabilisé par une violation qui vient s'ajouter à la fonction objectif partielle  $F_{\text{Chevau\_simples}}$ .

## 2. Le nombre de crédits total dans le mois ( $F_{\text{Credits}}$ ) :

Le nombre total de crédits de chaque horaire doit être compris entre les deux valeurs  $\text{Min}^{\text{credits}}$  et  $\text{Max}^{\text{credits}}$ . La somme pour tous les employés de la valeur de tout écart par rapport à ces nombres limites est ajoutée à la fonction objectif  $F_{\text{Credits}}$ .

## 3. Les chevauchements post-courriers ( $F_{\text{Chevau\_post}}$ ) :

Soit les rotations A et B suivantes :



Puisque le début de la rotation B est compris entre l'heure de fin et l'heure de fin avec un repos post-courrier de la rotation A, il y a chevauchement post-courrier entre les deux rotations. À cause de contraintes temporelles simples, les rotations A et B ne peuvent appartenir au même horaire et c'est pour cela qu'un tel cas est comptabilisé par une violation qui vient s'ajouter à la fonction objectif partielle  $F_{\text{Chevau\_post}}$ .



#### 4. Le total des jours de congé ( $F_{\text{Congé}}$ ) :

Le nombre de jours de congé minimal est  $\text{Min}^{\text{congé}}$ . La somme pour tous les employés de la valeur de tout écart au dessous de ce nombre est ajoutée à la fonction objectif  $F_{\text{Congé}}$ .

#### 5. Les jours de travail consécutifs ( $F_{\text{Travail\_consécutifs}}$ ) :

Le nombre maximal de jours travaillés consécutifs est  $\text{Max}^{\text{travaillés}}$ . La somme pour tous les employés de la valeur de tout écart au dessus de ce nombre est ajoutée à la fonction objectif  $F_{\text{Travail\_consécutifs}}$ .

La fonction objectif  $F$  totale est une combinaison linéaire de toutes les fonctions objectifs partielles. La mise à jour des coefficients  $p$  correspondant à chaque contrainte pénalise les contraintes qui demeurent violées lors de l'avancement du processus de résolution.

$$F = \rho_{\text{Chevau\_simples}} \times F_{\text{Chevau\_simples}} + \rho_{\text{Crédits}} \times F_{\text{Crédits}} + \rho_{\text{Chevau\_post}} \times F_{\text{Chevau\_post}} \\ + \rho_{\text{Congé}} \times F_{\text{Congé}} + \rho_{\text{Travail\_consécutifs}} \times F_{\text{Travail\_consécutifs}}$$

Lors de la recherche dans le voisinage, on ne procède pas à la réévaluation de la fonction objectif pour chaque solution; on calcule plutôt la différence entre la valeur de l'objectif correspondant à la solution de l'itération précédente et la valeur de l'objectif des voisins considérés lors de l'itération courante. Afin d'augmenter la vitesse d'exécution de l'algorithme, on a stocké les valeurs des fonctions objectifs partielles et ce pour tous les voisins possibles dans des tableaux. La mise à jour de ces tableaux est faite de façon incrémentale pour minimiser les temps de calculs lors de chaque itération.

Voici une illustration de l'équation régissant l'évaluation de la valeur objectif lors du processus de recherche dans le voisinage. Notons que  $\Delta$  est la différence de

la valeur de la fonction objectif partielle entre les deux états avant le mouvement et après le mouvement considéré :

$$\begin{aligned}
 \text{Eval\_solution}_1 = & \rho_{\text{Chevau\_simples}} \times \Delta (\text{nombre de violation des chevauchements} \\
 & \text{simples résultant du changement de l'affectation de la} \\
 & \text{rotation}) \\
 & + \rho_{\text{Crédits}} \times \Delta (\text{nombre de violation des crédits résultant du} \\
 & \text{changement de l'affectation de la rotation}) \\
 & + \rho_{\text{Chevau\_post}} \times \Delta (\text{nombre de violation des chevauchements post-} \\
 & \text{courriers résultant du changement de l'affectation de la} \\
 & \text{rotation}) \\
 & + \rho_{\text{Congé}} \times \Delta (\text{nombre de violation des jours de congé totaux} \\
 & \text{résultant du changement de l'affectation de la rotation}) \\
 & + \rho_{\text{Travail\_consécutifs}} \times \Delta (\text{nombre de violation des jours de travail} \\
 & \text{consécutifs résultant du changement de l'affectation de la} \\
 & \text{rotation})
 \end{aligned}$$

Il faudrait aussi noter que l'évaluation d'un voisin se fait par une simple lecture dans les tableaux des fonctions objectives partielles correspondantes et ce afin de savoir si le voisin en question améliore le coût total de la fonction objectif.

Quant à la deuxième étape du processus de résolution, en l'occurrence l'uniformisation de la solution obtenue précédemment, on a établi la moyenne des jours de congé totaux de tous les employés ainsi que le nombre de crédits moyens dans chaque horaire et on essaye de minimiser l'écart entre ces deux entités et l'horaire de chaque employé. Ces deux termes additionnels viennent s'ajouter aux 5 fonctions termes précédents qui s'assurent de la validité de

l'horaire considéré. Cependant, on peut avoir une solution non réalisable lors du processus de résolution, mais les poids accordés à chacun des termes de l'objectif s'assurent de guider l'algorithme vers une solution réalisable lors des itérations suivantes.

Voici une illustration de l'équation de la fonction objectif correspondante :

$$\text{Eval\_solution}_2 = \text{Eval\_solution}_1$$

$$+ \rho_{\text{Moyenne\_tot\_congé}} \times \Delta (\text{Moyenne} - \text{jours de congé résultant du changement de l'affectation de la rotation})^2$$

$$+ \rho_{\text{Moyenne\_Crédits}} \times \Delta (\text{Moyenne} - \text{nombre de crédits résultant du changement de l'affectation de la rotation})^2$$

### 3.3.5 Le voisinage d'une solution

Le voisinage d'une solution est composé par tous les horaires résultant du changement de l'attribution d'une rotation d'un employé à un autre, sont en d'autres termes le changement de la couleur d'un sommet dans le graphe construit. Il est cependant utile de souligner que lors de la première étape de résolution, on essaye de cibler les voisins qui règlent les violations des contraintes des fonctions objectifs partielles. D'un autre côté, il s'est avéré que considérer seulement un échantillon du voisinage total de chaque solution s'avère une alternative très payante d'un point de vue du temps de calcul. Donc, on a considéré des voisinages dont la taille augmente dépendamment du nombre de violations dans la fonction objectif. En effet, au début du processus de résolution, il n'est pas nécessaire de considérer tout le voisinage d'une solution pour trouver une solution avec un meilleur coût. Mais, au fur et à mesure que le processus de résolution est proche de la fin, l'algorithme requiert plus de voisins pour pouvoir améliorer la valeur de la fonction objectif. D'où,

l'utilisation de voisinages variables en fonction du nombre de violations dans la fonction objectif. Voici une illustration de la taille du voisinage utilisé.

Si la valeur de chaque fonction objectif partielle est supérieur à un seuil  $Y$ , la taille du voisinage est égale à  $X \times \text{Nombre de solutions à évaluer}$  ;

➤ Pour  $Y = 100$ ;  $X = \frac{1}{10}$  ;

➤ Pour  $Y = 50$ ;  $X = \frac{1}{5}$  ;

➤ Pour  $Y = 30$ ;  $X = \frac{1}{3}$  ;

➤ Pour  $Y = 10$ ;  $X = \frac{1}{2}$  ;

Lorsque l'algorithme arrive à couvrir toutes les rotations et commence le processus d'uniformisation, on procède à une réévaluation de la taille du voisinage ( $X = 1$ ) pour intensifier la recherche locale autour des solutions trouvées et évaluer tous les voisins possibles. Cette taille de voisinage est maintenue tout au long de la suite du processus de résolution, et ce même si l'algorithme parcourt des solutions irréalisables résultant en une valeur de la fonction objectif supérieure à 10.

### 3.3.6 Le contenu de la liste Tabou et sa longueur

La liste Tabou contient des paires (Sommet, couleur) interdites ainsi que le nombre d'itérations après lesquels l'interdiction est levée. En effet, la nouvelle rotation doit être interdite à l'ancien employé pour un certain nombre d'itérations mais aussi la même rotation doit demeurer dans l'horaire du nouvel employé pour un certain nombre d'itérations et ce pour prospector au maximum le voisinage de cette solution. Comme on l'a décrit précédemment, cela est

considéré comme un critère d'intensification de la recherche Tabou. D'un autre côté, le critère d'aspiration adopté permet le choix d'une solution même si la paire (Sommet, couleur) est Tabou dans le cas où un tel choix améliore la valeur de la meilleure fonction objectif trouvée. Quant à la longueur de la liste en question, elle est proportionnelle à la racine carrée du nombre de voisins évalués (noté  $NbV$ ) lors de l'itération courante. Les valeurs des coefficients  $\alpha_1$  et  $\alpha_2$  correspondant respectivement à la nouvelle et l'ancienne couleur sont ajustées automatiquement au cours du processus de résolution.

### 3.3.7 Structure de l'algorithme

Voici le pseudo-code de l'algorithme Tabou développé dans le cadre de ce mémoire. Pour plus d'informations sur l'algorithme ou encore l'étape 4.

Pour des fins de simplification, la procédure RECHERCHE qui évalue le coût du mouvement choisi et renvoie à la fin de l'itération dans le cas où la meilleure solution courante est améliorée.

Tant que le nombre des itérations est inférieur à  $Max\_ITER$  Faire

1. Parcourir la liste des horaires (Couleurs) n'ayant pas  $Min^{credits}$

- nouvelle\_couleur = couleur n'ayant pas  $Min^{credits}$ ;
- sommet\_courant = RAND (nombre\_des\_sommets) ;
- Parcourir toute la liste de tous les sommets tout en s'assurant que la couleur du sommet n'est pas nouvelle\_couleur ;
  - Si le sommet n'est pas Tabou pour la nouvelle et l'ancienne couleur ou que le critère d'aspiration est satisfait ;

- ancienne\_couleur = couleur du sommet\_courant;

➤ RECHERCHE

2. Parcourir la liste des couleurs ayant plus que  $\text{Max}^{\text{credits}}$

- Parcourir toute la liste de toutes les tâches formant l'horaire de la personne courante
  - ancienne\_couleur = couleur du sommet\_courant (tâche courante);
  - nouvelle\_couleur = RAND (nombre\_des\_couleurs );
  - Parcourir toute la liste des couleurs tout en s'assurant que la nouvelle couleur est différente de l'ancienne couleur ;
    - ◆ Si la couleur n'est pas Tabou pour le sommet choisi ou que le critère d'aspiration est satisfait ;

➤ RECHERCHE;

sommet\_courant = RAND(nombre\_des\_sommets );

3. Parcourir toute la liste des sommets

- Si le sommet\_courant est problématique pour les contraintes restantes
  - ancienne\_couleur = couleur du sommet\_courant;
  - nouvelle\_couleur = RAND(nombre\_des\_couleurs );
  - Parcourir la liste des couleurs tout en s'assurant que la nouvelle couleur est différente de l'ancienne couleur ;
    - ◆ Si la couleur n'est pas Tabou pour le sommet choisi ou que le critère d'aspiration est satisfait ;

➤ RECHERCHE;

4. Mise à jour des structures de données avec la nouvelle couleur qui est adoptée au sommet\_courant;
5.  $\text{Tabou}[\text{sommet\_courant}][\text{ancienne\_couleur}] = \alpha_1 \times \sqrt{NbV}$  ;
6.  $\text{Tabou}[\text{sommet\_courant}][\text{nouvelle\_couleur}] = \alpha_2 \times \sqrt{NbV}$  ;

Mise à jour du nombre des itérations et les coûts correspondants à l'itération courante;

Voici le détail de la procédure RECHERCHE utilisée précédemment :

- Évaluer la fonction objectif  $F$  de ce mouvement et stocker le meilleur mouvement s'il correspond à une meilleure valeur de la fonction objectif que celle de l'itération courante;
- Si le mouvement courant améliore la meilleure solution trouvée
  - Aller à l'étape 4;

### 3.4 Limites de l'approche Tabou

Les métaheuristiques en général et l'algorithme Tabou en particulier ont de nombreux avantages mais aussi un certain nombre de limitations. En effet, ces méthodes sont connues par leur généralité et l'application possible à une large classe de problèmes. Outre le fait de leur efficacité prouvée pour de nombreux problèmes, ces méthodes offrent des possibilités de compromis entre qualité des solutions et temps de calcul. La principale limitation pratique de ces méthodes est l'absence de garantie d'optimalité de la solution trouvée. D'un autre côté, ces méthodes nécessitent de multiples ajustements des paramètres de la recherche locale afin d'adapter la méthode au problème traité. Résultat de

toutes ces limitations, il est difficile de prévoir la performance de ce genre de méthodes.

En définitive, les métaheuristiques n'offrent aucune garantie en vue de trouver la solution optimale ; mais aussi elles ne prouvent pas l'optimalité de la solution trouvée. Dans ces conditions, les méthodes exactes telles que la génération de colonnes et les métaheuristiques sont plutôt complémentaires que véritablement concurrentes. Ces deux approches peuvent d'ailleurs également coopérer de différentes manières comme ce fut le cas pour les travaux de Ouellet (2004) et Gamache et al. (2005). En s'alignant sur cette approche, la méthode adoptée dans ce mémoire utilise la complémentarité de ces deux méthodes de résolution. En effet, l'utilisation d'une méthode Tabou pour l'élaboration de la solution initiale est suivie par l'appel à l'agrégation dynamique de contraintes pour l'obtention éventuelle de la solution optimale du problème traité.



## **CHAPITRE 4 : GÉNÉRATION DE COLONNES ET AGRÉGATION DYNAMIQUE DE CONTRAINTES**

Ce chapitre présente la méthode de génération de colonnes ainsi que l'approche par agrégation dynamique de contraintes. Pour chacune d'elles, nous allons présenter d'abord l'approche dans un contexte générique avant de décrire l'application au problème de construction d'horaires et les différents paramètres utilisés.

### **4.1 Approche classique de génération de colonnes**

#### **4.1.1 Concepts de base**

La génération de colonnes est une méthode pour résoudre efficacement les programmes linéaires de grande taille. Elle repose sur la décomposition de Dantzig et Wolfe (1960) qui consiste à décomposer l'ensemble des contraintes en deux sous-ensembles. Cette approche a suscité un grand intérêt de recherche qui a donné lieu à un grand nombre d'articles notamment les articles de synthèse de Desaulniers et al. (1998b) et Barnhart et al. (1998a) et un récent livre de Desaulniers et al. (2005) qui présente une dizaine d'articles sur l'application de la génération de colonnes dans différents contextes.

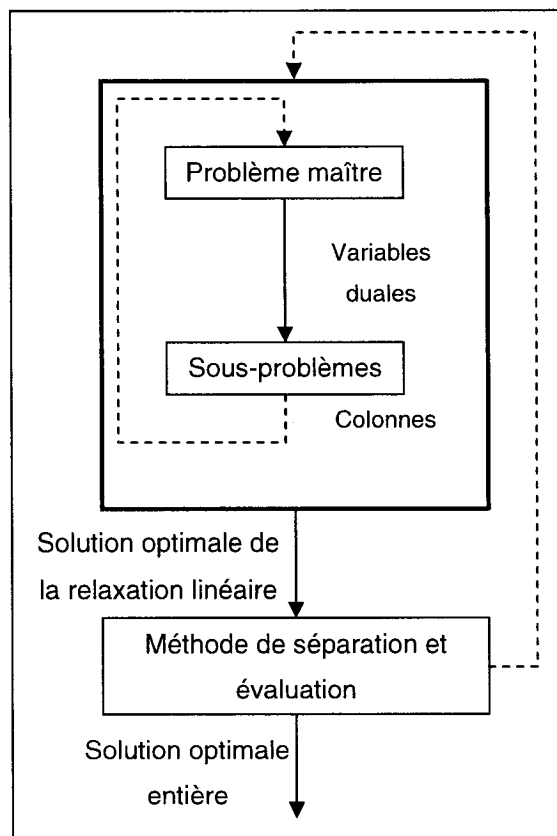
L'idée centrale de cette méthode repose sur le fait que les programmes linéaires de grande taille ont trop de variables, ou colonnes, pour qu'on puisse les représenter toutes de manière explicite. À l'optimum, la plupart des variables

sont hors base et nulles, c'est-à-dire que seul un sous-ensemble restreint de variables doit être pris en compte pour résoudre le problème. Une méthode utilisant la génération de colonnes initialise le programme linéaire avec un sous-ensemble de colonnes de petite taille. Le mécanisme de la génération de colonnes consiste alors à générer, de façon itérative, les variables qui sont susceptibles d'améliorer la solution courante, c'est-à-dire celles qui ont des coûts réduits négatifs. D'où la décomposition du problème initial en deux : un problème maître et un sous-problème qui est souvent séparé en plusieurs sous-problèmes de plus petite taille. Le problème maître est identique au problème initial 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 bien des contextes, l'approche de résolution proposée par la génération de colonnes utilise une représentation en réseau pour modéliser les sous-problèmes. Ces réseaux sont une série de nœuds et d'arcs. La résolution des sous-problèmes consiste à trouver des chemins dans ces réseaux en résolvant un plus court chemin avec contraintes de ressources et correspondent aux colonnes de la matrice dans la formulation du problème initial. Une fois que ces chemins sont validés, ils serviront comme données d'entrée pour le problème maître. Le processus de résolution de la génération de colonnes procède de façon itérative résolvant tantôt le problème maître tantôt le sous-problème et ce comme l'illustre la figure 4.1. Le problème maître est résolu à chaque itération tout en tenant compte des contraintes globales. Ces contraintes sont conservées tout au long du processus en plus des contraintes locales qui sont quant à elles prises en compte par le sous-problème et sont généralement modélisées sous forme de ressources. Une ressource est une commodité consommée à chaque fois qu'un arc du réseau est emprunté. La consommation

de ressources se fait sur les arcs et la validation des restrictions sur les ressources sur les noeuds. Dès lors, un chemin est dit valide si la consommation de chaque ressource cumulée le long du chemin respecte les restrictions à chaque noeud. 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 en tenant compte de la validité des chemins générés. 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. Le problème maître est alors de nouveau résolu en utilisant les colonnes produites par les sous-problèmes. Le processus itératif se termine lorsque la valeur optimale du sous-problème est non négative. En effet, cela démontre qu'aucun chemin non considéré ne peut contribuer à l'amélioration de la solution optimale courante du problème maître, prouvant ainsi l'optimalité de cette dernière.

Afin d'obtenir une solution entière lorsque cela est exigé, l'approche de génération de colonnes est habituellement imbriquée dans un algorithme de séparation et évaluation progressive. Dans ce cas, l'approche de génération de colonnes est utilisée pour calculer une borne inférieure à chaque nœud de l'arbre de branchement.



**Figure 4.1 : L'algorithme de génération de colonnes**

#### **4.1.2 Applications à la construction d'horaires pour les compagnies aériennes**

L'application de la méthode de génération de colonnes a permis de faire une percée dans le domaine de la fabrication d'horaires d'équipages aériens. En effet, c'est la première approche capable de résoudre de façon optimale des problèmes de grande taille. Pour ce qui est du transport aérien, cette approche peut servir à la génération d'horaires de travail aussi bien pour le personnel technique que pour le personnel commercial, mais elle peut aussi servir à la construction de rotations d'équipages et éventuellement de rotations d'avions. Notons que chacune de ces applications nécessite certaines adaptations quant à la modélisation du problème, chaque problème ayant ses propres données

d'entrée et ses propres contraintes requérant différents ajustements avant l'exécution du logiciel d'optimisation.

#### **4.1.3 Spécialisation au problème d'horaires mensuels des membres d'équipages**

Comme présenté précédemment la génération de colonnes est basée sur la décomposition du modèle initial en problème maître et sous-problèmes. Voici une illustration du modèle utilisé dans le cadre de ce mémoire. Il faudrait cependant souligner que ce modèle est basé sur celui présenté à la section 2.1 auquel on a apporté quelques modifications :

- Ajout d'une nouvelle partie représentant l'uniformisation des horaires des employés;
- Adoption du Bidline pour la construction des horaires : suppression des indices pour différencier les employés ainsi que les préaffectations car ils sont considérés tous identiques dans notre cas.

Soit :

- $\Omega$  l'ensemble des horaires admissibles;
- $m$  le nombre d'employés;
- $n$  le nombre de rotations à affecter;
- $a_{ij}$  un paramètre qui est égal à 1 si la rotation  $i$  est comprise dans l'horaire  $j$  et 0 autrement;
- $x_j$  une variable de décision qui est égale à 1 si l'horaire  $j$  est choisi et 0 autrement;
- $e_j$  nombre de crédits travaillés dans l'horaire  $j$  ;
- $f_j$  le nombre de jours de congé dans l'horaire  $j$ ;
- $Moy^{\text{credits}}$  le nombre moyen de crédits travaillés par horaire ;

- $Moy^{congé}$  le nombre moyen de jours de congé par horaire ;
- $\lambda$  un poids utilisé pour équilibrer les deux parties de l'objectif.

Le problème de construction d'horaires se formule alors comme suit :

$$\text{Minimiser} \quad \sum_{j \in \Omega} \frac{(e_j - Moy^{credits})^2}{m} x_j + \lambda \sum_{j \in \Omega} \frac{(f_j - Moy^{congé})^2}{m} x_j \quad (4.1)$$

Sujet aux contraintes :

$$\sum_{j \in \Omega} a_{ij} x_j = 1, \quad \forall i = 1, 2, \dots, n \quad (4.2)$$

$$\sum_{j \in \Omega} x_j = m, \quad (4.3)$$

$$x_j \geq 0, \text{ entier} . \quad (4.4)$$

L'objectif (4.1) correspond à minimiser les variances sur les crédits travaillés par horaire et sur les jours de congé par horaire. Il faudrait cependant noter que la moyenne des crédits travaillés  $Moy^{credits}$  est connue au préalable et elle est égale à  $\frac{\text{Total des crédits travaillés}}{m}$ . Par contre, du fait que les horaires ne sont

pas connus au préalable, la moyenne du nombre de jours de congé

$Moy^{congé} = \frac{\sum_j f_j x_j}{m}$  n'est pas connue et introduit de la non linéarité dans

l'objectif. De ce fait, on a eu recours à une modélisation heuristique du problème. La moyenne qui a été utilisée est celle fournie par l'algorithme Tabou fournie au préalable. `

Bien que cette substitution rend la formulation (4.1) - (4.4) linéaire, celle-ci ne peut être résolue facilement par génération de colonnes. En effet, les termes de l'objectif devraient aussi apparaître dans l'objectif du sous-problème comme des termes non linéaires qui sont fonctions de deux ressources, nombre de crédits travaillés et nombre de jours de congé. Pour ces deux ressources, il faudrait utiliser des fonctions de prolongation (voir Irnich et Desaulniers, 2005) qui ne sont pas non décroissantes ce qui rend totalement inefficace les algorithmes de programmation dynamique habituellement utilisés pour résoudre le sous-problème.

Pour éviter cette difficulté, nous proposons d'utiliser plusieurs sous-problèmes  $SP^z$  avec  $z = 1, 2, \dots, Z$  ayant des niveaux de flexibilité différents quant aux écarts permis pour les crédits travaillés et les jours de congé par rapport aux moyennes visées. Ces niveaux de flexibilité seront imposés en resserrant plus ou moins les plages de validité des ressources pour les crédits travaillés et les jours de congé. L'utilisation d'un horaire provenant d'un sous-problème avec peu de flexibilité sera faiblement pénalisée alors que l'utilisation d'un horaire provenant d'un sous-problème avec beaucoup de flexibilité sera fortement pénalisée. Il faudrait souligner le fait qu'un horaire qui satisfait aux plages de validité les plus strictes peut être généré par tous les sous-problèmes. Par contre, le sous-problème le plus restrictif sera favorisé pour le générer à cause qu'il est moins pénalisé.

Les sous-problèmes sont donc des problèmes de plus court chemin avec contraintes de ressources. Ils sont basés sur des réseaux espace-temps. Un tel réseau  $G^z$  avec  $z = 1, 2, \dots, Z$  est illustré à la figure 4.2.

Il comporte les nœuds suivants :

- Source : nœud initial du réseau ;

- Minuits : il y a 31 nœuds de ce type, ils représentent la succession des minuits sur une période de un mois. Ces nœuds sont la base de l'axe espace-temps du modèle ;
- Début\_Tâche : ces nœuds représentent le début de chaque rotation et ils sont placés entre les nœuds minuits sur l'axe espace-temps et ce en fonction de la journée pendant laquelle la tâche débute ;
- Fin\_Tâche : ces nœuds représentent la fin effective de chaque rotation ;
- Fin\_Post\_courrier : ces nœuds représentent le repos post-courrier associé à chaque rotation et ils sont placés entre les nœuds minuits sur l'axe espace-temps ;
- Fin\_Avec\_Congé : ces nœuds représentent la fin de la rotation suivi par un jour de congé et ce sans comptabiliser un repos post-courrier et ils sont placés entre les nœuds minuits sur l'axe espace-temps ;
- Puits : nœud final du réseau.

Comme illustré dans la figure 4.2, il y a 7 groupes d'arcs dans ce modèle:

- Arcs d'attente : ils représentent l'attente d'un employé en vue d'une affectation durant la journée courante;
- Minuit-Minuit : ces arcs représentent les jours de congé. Ils relient les minuits consécutifs durant le mois;
- Début\_Tâche-Fin\_Tâche : Ces arcs sont valables pour toutes les tâches et représentent la rotation correspondante à la tâche en question ;
- Fin\_Tâche -Fin\_Post\_courrier : Ces arcs sont valables pour toutes les tâches et représentent le repos post-courrier obligatoire associé à chaque rotation avant de pouvoir entamer une nouvelle affectation ;



- Fin\_Tâche -Fin\_Avec\_Congé : Ces arcs sont valables pour toutes les tâches et représentent le cas d'un jour de congé attribué après la fin d'une rotation;
- Source-Premier\_Minuit et Dernier\_Minuit-Puits : il y a seulement un arc de chaque genre par réseau. Ils représentent le débit et la fin d'un horaire.

Tous les arcs ont un coût nul sauf les arcs de tâche qui ont un coût  $(-\pi_i)$  où  $\pi_i$  est la valeur de la variable duale de la contrainte (4.2) associée à la rotation correspondante et l'arc « Source - Premier\_Minuit » qui a un coût égal à  $P^Z - \sigma$  où  $P^Z$  est la pénalité pour l'uniformisation associée au sous-problème  $SP^Z$  et  $\sigma$  est la valeur de la variable duale de la contrainte (4.3).

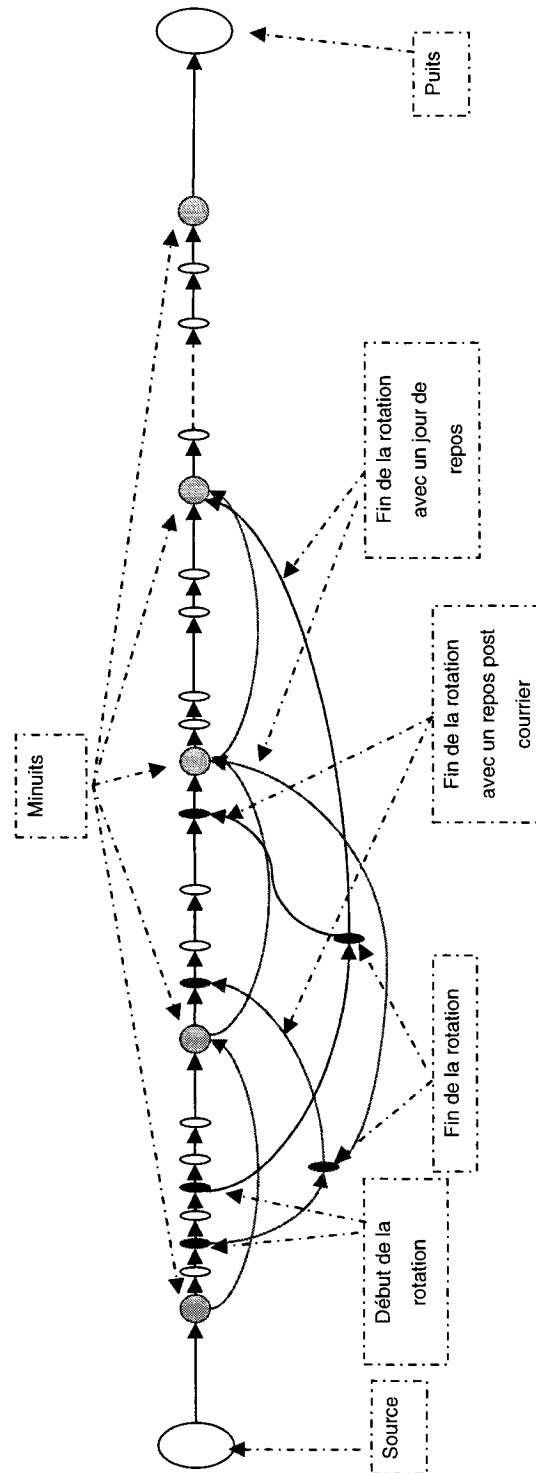


Figure 4.2 : Exemple de réseau utilisé pour la génération de colonnes

Dans les sous-problèmes, 5 ressources sont utilisés pour valider les contraintes sur :

1. Nombre maximal de crédits pour chaque employé;
2. Nombre minimal de crédits pour chaque employé;
3. Nombre minimal de jours de congé dans le mois pour chaque employé;
4. Nombre maximal de jours de congé dans le mois pour chaque employé;
5. Nombre maximal de jours travaillés consécutifs.

D'un autre côté, l'ajout d'une sixième ressource a été nécessaire pour la validation des jours de congé. En effet, cette ressource est utilisée afin d'interdire l'utilisation des arcs d'attente pour modéliser un jour de congé au lieu de choisir l'arc Minuit-Minuit. Il faudrait cependant noter que la ressource 4 a été ajoutée pour uniformiser la contrainte des jours de congé totaux et ce pour satisfaire aux exigences du problème. La consommation des ressources sur les arcs est présentée dans le tableau 4.1.

Les plages de validité des ressources permettent de modéliser les contraintes suivantes qui dépendent du sous-problème  $SP^z$  avec  $z = 1, 2, \dots, Z$  :

- $[\text{Min}^{\text{credits}}_z, \text{Max}^{\text{credits}}_z]$  : l'intervalle représentant le nombre minimal et maximal de crédits dans le mois;
- $[\text{Min}^{\text{congé}}_z, \text{Max}^{\text{congé}}_z]$  : l'intervalle représentant les jours de congé totaux pour chaque employé ;
- $\text{Max}^{\text{travaillés}}$  : le nombre maximal de jours travaillés consécutifs.

Les plages de ressources sont données dans le tableau 4.2.

**Tableau 4.1 : Consommation des ressources**

<div> <div>Ressource consom- -mée</div> <div>Arcs du réseau</div> </div>	Nombre maximal de crédits	Nombre minimal de crédits	Nombre maximal de jours de congé	Nombre minimal de jours de congé	Nombre de jours travaillés consécutifs	Ajustement des jours de congé
Arcs d'attente	0	0	0	0	+1 si départ d'un Minuit et 0 sinon	+1 si départ d'un Minuit et 0 sinon
Minuit - Minuit	0	0	1	- 1	Remise à 0	0
Début_Tâche - Fin_Tâche	Crédits de la tâche i	- (Crédits de la tâche i)	0	0	+ 1 pour chaque minuit rencontré par la tâche	-1
Fin_Tâche - Fin_Post_courrier	0	0	0	0	+1 si passage par Minuit et 0 sinon	0
Fin_Tâche - Fin_Avec_Congé	0	0	1	- 1	Remise à 0	0
Source - Minuit	0	0	0	0	0	0
Minuit - Puits	0	0	0	0	0	0

Tableau 4.2 : Plages de validité des ressources

Plage de la ressource	Nombre maximal de crédits	Nombre minimal de crédits	Nombre maximal de jours de congé	Nombre minimal de jours de congé	Nombre de jours travaillés consécutifs	Ajustement des jours de congé
Noeuds du réseau						
Source	$[0, \text{Max}^{\text{crédits}}_z]$	$[\text{Min}^{\text{crédits}}_z, \text{Min}^{\text{crédits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[\text{Min}^{\text{congé}}_z, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}]$	$[0, 0]$
Minuit	$[0, \text{Max}^{\text{crédits}}_z]$	$[0, \text{Min}^{\text{crédits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}]$	$[0, 0]$
Début_Tâche						
Fin_Post_courrier	$[0, \text{Max}^{\text{crédits}}_z]$	$[0, \text{Min}^{\text{crédits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}]$	$[0, 1]$
Fin_Avec_Congé						
Fin_Tâche	$[0, \text{Max}^{\text{crédits}}_z]$	$[0, \text{Min}^{\text{crédits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}]$	$[0, 0]$
Puits	$[0, \text{Max}^{\text{crédits}}_z]$	$[0, 0]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, 0]$	$[0, \text{Max}^{\text{travaillés}}]$	$[0, 0]$

Les valeurs des paramètres  $Min^{credits}_z$ ,  $Max^{credits}_z$ ,  $Min^{congé}_z$  et  $Max^{congé}_z$  sont calculés comme suit :

➤ Cas du nombre de crédits dans le mois (Ressources 1 et 2) :

$$Min^{credits}_z = Min^{credits} + (z-1) \times \left[ \frac{\left( \frac{Max^{credits} - Min^{credits}}{2} \right)}{Z} \right], \forall z = 1, 2, \dots, Z$$

$$Max^{credits}_z = Max^{credits} - (z-1) \times \left[ \frac{\left( \frac{Max^{credits} - Min^{credits}}{2} \right)}{Z} \right], \forall z = 1, 2, \dots, Z$$

Où  $[Min^{credits}, Max^{credits}]$  est l'intervalle initial sur le nombre de crédits alloués par horaire fourni par la compagnie aérienne.

➤ Cas du nombre maximal de jours de congé (Ressources 3 et 4) :

On a  $Min^{congé}$  le nombre minimum de jours de congé dans un horaire selon la compagnie aérienne et  $Max^{congé}$  un nombre maximal fixé à 20 dans notre cas.

On a choisi d'utiliser une valeur différente pour l'écart correspondant à  $Min^{congé}$  et  $Max^{congé}$  pour cette ressource du fait que la distribution des valeurs du nombre total de jours de congé n'est pas uniforme sur tout l'intervalle considéré.

$$\begin{aligned}
Min^{Congé}_z &= Min^{Congé} + \left\lfloor (z-1) \times \left( \frac{Max^{Congé} - Min^{Congé}}{Z} \right) \right\rfloor, \quad \forall z = 1, 2, \dots, Z \\
Max^{Congé}_z &= \left\lceil Min^{Congé} + 4 \times (Max^{Congé} - Min^{Congé}) - \left\lfloor (z-1) \times \left( \frac{2 \times (Max^{Congé} - Min^{Congé})}{Z} \right) \right\rfloor \right\rceil \\
&\quad \forall z = 1, 2, \dots, Z
\end{aligned}$$

On note que la pénalité ajoutée à l'arc entre la source et le premier nœud du réseau est d'autant plus grande que la différence entre les deux bornes est grande. On a utilisé la structure de coût suivante (rappelons que  $Moy^{congé}$  est la valeur fournie par l'algorithme Tabou pour la moyenne des jours de congé totaux pour tous les employés) :

$$P_z = \left[ Max^{Credits}_z - \frac{Min^{Credits} + Max^{Credits}}{2} \right]^2 + \lambda [Max^{Congé}_z - Moy^{congé}]^2, \forall z = 1, 2, \dots, Z.$$

Voici un exemple numérique qui illustre explicitement les intervalles utilisés. Notons que ces valeurs sont fournies par la compagnie aérienne et ont été considérés lors des tests numériques présentés au chapitre 5 :

- Le nombre de crédits pour chaque employé doit être compris entre 65 heures et 85 heures pour toute la durée du mois. Cela revient à 3900 et 5100 minutes pour chaque horaire;
- Un horaire doit avoir un minimum de 10 jours de congé;
- La moyenne des jours de congé pour tous les employés est 13 jours par mois;
- Un employé ne peut travailler plus que 6 jours consécutifs sans un jour de congé;
- Le nombre de réseaux choisis pour l'uniformisation est 5.

- La valeur de  $\lambda$  a été fixée à 12 après une série de tests pour déterminer le meilleur poids entre les deux composantes du poids  $P_Z$ .

Voici un tableau explicatif des plages de validité des ressources pour les différents réseaux utilisés. Notons que la ressource six n'est pas incluse dans ce tableau car elle ne varie pas en fonction du réseau choisi, elle s'assure seulement de la validité de l'horaire.

**Tableau 4.3 : Exemple des plages de validité des ressources**

Ressource et coût du sous- réseau N° du Sous-réseau	Coût	Nombre maximal de crédits	Nombre minimal de crédits	Nombre maximal de jours de congé	Nombre minimal de jours de congé	Nombre de jours travaillés consécutifs
1	361680	5100	3900	20	10	6
2	203868	4980	4020	19	10	6
3	91080	4860	4140	17	10	6
4	23316	4740	4260	16	11	6
5	576	4620	4380	14	12	6

#### 4.1.4 Stratégies de branchement et d'accélération

Afin d'accélérer le processus de résolution, on a utilisé les deux méthodes suivantes :

- Un branchement fixant les variables d'horaires  $x_j$ . Cette stratégie consiste à fixer à 1 une ou plusieurs variables de chemin ayant une valeur fractionnaire au nœud courant supérieure à un seuil donné. En



pratique, il est impossible d'établir une condition nécessaire et suffisante qui puisse être rapidement vérifiée et qui assure que le problème soit toujours réalisable. Cependant, nous pouvons nous assurer que chaque contrainte de couverture de tâche est couverte au plus une fois dans l'ensemble des colonnes fixées à 1, ce qui constitue une condition nécessaire de réalisabilité ;

- L'utilisation d'une règle de dominance heuristique. À chaque nœud de branchement, on domine en premier lieu seulement sur les ressources correspondant au minimum des crédits travaillés et au maximum de jours travaillés consécutifs. Ensuite, lorsqu'aucune colonne de coût réduit négatif ne peut être générée, on rajoute la ressource correspondant au minimum de jours de congé dans le test de dominance. Ainsi la dominance entre les étiquettes n'est jamais fixé de façon exacte et on ne peut donc pas garantir l'optimalité des solutions des relaxations linéaires.

Il faudrait mettre l'accent sur le fait que l'approche de résolution adoptée ne garantit pas l'optimalité de la solution obtenue à la fin du processus. Ainsi un compromis a été trouvé entre les méthodes d'accélération et la qualité de la solution finale.

#### **4.1.5 Limites de la génération de colonnes**

Comme on l'a étalé lors des chapitres précédents, la génération de colonnes est utilisée avec succès pour traiter de nombreuses applications relatives à la planification aérienne. Cependant, il faudrait souligner que cette méthode a un certain nombre d'inconvénients limitant considérablement la possibilité d'utiliser cette approche pour les problèmes de grande taille. En effet, les temps de calculs qui augmentent rapidement avec le nombre de contraintes rendent les problèmes de grande taille très difficiles à

résoudre. Cette limitation est d'autant plus contraignante avec l'augmentation du nombre moyen d'éléments non nuls par colonne. Cela est due surtout à la nature des problèmes considérés qui causent beaucoup de dégénérescence ce qui se répercute directement sur l'algorithme de résolution en augmentant considérablement le temps de convergence. En effet, il est impossible de résoudre dans des temps raisonnables des problèmes de quelques milliers de contraintes avec plus d'une dizaine d'éléments non nuls par colonne. Toutes ces limitations ont poussé les chercheurs à développer des améliorations à l'algorithme de la génération de colonnes, notamment une méthode appelée l'agrégation dynamique de contraintes.

## **4.2 L'agrégation dynamique de contraintes**

### **4.2.1 Introduction**

La méthode d'agrégation dynamique de contraintes a été développée au GERAD et présentée dans les travaux d'El Hallaoui et al. (2005 et 2006). Cette nouvelle approche résout efficacement les problèmes de partitionnement d'ensemble de grande taille en combattant la dégénérescence survenant régulièrement dans ces problèmes. Il s'agit d'une méthode primale et exacte qui propose un nouveau paradigme en recherche opérationnelle. De plus, cette méthode peut se jumeler à la méthode de génération de colonnes, comme nous le proposons dans ce mémoire.

Les motivations qui ont poussé à développer cette approche résident principalement dans la nature des problèmes considérés. En effet, pour les problèmes de création d'horaires ou d'assignation de véhicules, il s'avère qu'en cas de réoptimisation, les horaires de la solution optimale s'écartent peu des horaires de la solution planifiée. D'un autre côté, il a été prouvé que les

équipes empruntent des horaires sensiblement identiques aux horaires des véhicules. De ce fait, chacun de ces deux cas peut donner lieu à une solution initiale s'écartant peu de la solution optimale. Cela se traduit par le fait que plusieurs tâches qui sont consécutives dans les horaires initiaux vont probablement demeurer groupées dans les horaires finaux appartenant à la solution optimale. L'ensemble de tâches composant chaque horaire initial peut alors être agrégé et les contraintes associées remplacées par une seule contrainte dans le problème maître. Cependant, puisqu'on ne connaît pas a priori la constitution de la solution optimale, on ne peut prédire les séquences de tâches qui vont rester inchangées par rapport à la solution initiale. De ce fait, l'agrégation initiale peut ne pas être définitive et doit être ajustée dynamiquement pour obtenir la solution optimale. L'agrégation initiale est souvent construite à l'aide d'une heuristique, Tabou dans notre cas, et utilisée pour entamer le processus de résolution par l'algorithme d'agrégation dynamique de contraintes.

#### **4.2.2 Présentation de la méthode**

L'approche de résolution de l'algorithme d'agrégation dynamique de contraintes est basée sur l'agrégation d'un certain nombre de contraintes du problème maître restreint de la méthode de génération de colonnes afin de diminuer le nombre total de contraintes de partitionnement d'ensemble. Le problème résultant est appelé le problème maître restreint agrégé PMRA. Dû à sa petite taille, celui-ci se résout plus rapidement à chaque itération. Pour expliquer le mécanisme de résolution de cette nouvelle approche, on va considérer le modèle (4.1) – (4.4).

Chaque agrégation est obtenue à partir d'une partition  $Q$  des rotations (tâches) en séquence de rotations. Pour former une partition, les séquences de rotations

doivent être disjointes et leur union doit contenir toutes les rotations du problème.

Une variable  $x_j$  et son horaire associé sont dits compatibles avec une partition  $Q$  si les rotations couvertes par l'horaire correspondent à une concaténation de séquences de rotations de  $Q$ . Les variables et les horaires qui ne sont pas compatibles sont dits incompatibles. De plus, on leur associe un degré d'incompatibilité qui indique approximativement le nombre de fois que les séquences de rotations devraient être brisées pour les rendre compatibles avec  $Q$ .

Au début du processus de résolution, la partition  $Q$  est définie à partir d'une solution planifiée ou encore d'une solution initiale fournie par une heuristique quelconque. Dans notre cas, l'algorithme Tabou fournit une telle solution et la partition initiale est composée des séquences de rotations correspondant aux horaires initiaux de cette solution. En cours de résolution, la partition  $Q$  est ajustée dynamiquement en modifiant l'ensemble des éléments qui la compose jusqu'à l'obtention d'une solution optimale.

À chaque itération de la génération de colonnes, le PMRA contient une seule contrainte de partitionnement d'ensemble (4.2) pour chaque séquence de rotations de  $Q$  et seulement les variables générées qui sont compatibles avec  $Q$ .

Le processus de résolution de l'agrégation dynamique de contraintes comporte deux types d'itérations : mineures et majeures. Dans une itération mineure, le PMRA est optimisé et des colonnes compatibles sont produites par les sous-problèmes. Tandis que, lors d'une itération majeure, une série d'itérations mineures est exécutée avant d'ajuster la partition  $Q$ . Pour produire de nouvelles colonnes, les sous-problèmes requièrent l'utilisation d'une solution duale

complète, c'est-à-dire, d'une valeur pour chaque variable duale des contraintes (4.2) du problème maître non agrégé. Par conséquent, l'algorithme doit procéder à la désagrégation de la solution duale du PMRA. Cette désagrégation peut se faire en résolvant un problème de plus court chemin et vise à produire des variables duales qui limitent le plus possible le nombre de colonnes incompatibles de coût réduit négatif. Mentionnons que les sous-problèmes peuvent produire des colonnes qui sont compatibles ou incompatibles avec la partition  $Q$ . Les colonnes incompatibles ne sont évidemment pas ajoutées au PMRA, mais elles peuvent d'abord être utilisées pour la mise à jour de la partition  $Q$  lorsque leurs coûts réduits semblent bien avantageux par rapport aux coûts réduits des variables compatibles. Dans ce cas, la partition  $Q$  est désagrégée afin de permettre de considérer ces variables dans le PMRA. Finalement, notons que lorsque le nombre de contraintes du PMRA dépasse un seuil prédéterminé, la partition  $Q$  est ré-agrégée en prenant en compte seulement les horaires associés à des variables de valeurs positives dans la solution courante.

Depuis l'introduction de cette nouvelle méthode, plusieurs améliorations ont été apportées à l'algorithme d'agrégation dynamique afin d'en augmenter l'efficacité et réduire les temps de calcul. La première de ces améliorations fut d'adopter un choix des colonnes susceptibles à ajouter dans le PMRA suivant leur degré d'incompatibilité avec la partition considérée. Au début du processus de résolution, uniquement les colonnes compatibles avec la partition choisie sont considérées. Par la suite, le degré d'incompatibilité augmente jusqu'à considérer toutes les colonnes même celles qui sont incompatibles avec la partition initiale et ce afin de pouvoir atteindre l'optimalité. Outre le fait de maintenir la taille de la partition plus petite, la nouvelle approche réduit le nombre de désagrégations de la partition ce qui accélère considérablement le processus de résolution. Ce nouvel algorithme est appelé agrégation

dynamique de contraintes avec phases multiples (ou *Multi-Phase Dynamic Constraint Algorithm* MPDCA en anglais). En effet, chaque augmentation du degré d'incompatibilités admissibles donne lieu à une nouvelle phase. En pratique, on utilise souvent les phases 0, 1, 2 et  $\infty$  signifiant que, dans ces phases, seulement les colonnes compatibles, avec au plus une incompatibilité, avec au plus deux incompatibilités, et toutes les colonnes, respectivement, peuvent être générées par les sous-problèmes. Pour nos tests, la phase  $\infty$  a été omise rendant l'approche heuristique.

D'un autre côté, El Hallaoui (2006) s'est aussi penché sur la résolution en nombres entiers, ce qui n'avait pas été fait dans les deux versions précédentes de l'agrégation dynamique. Il s'est avéré que, lors du processus de résolution, dû à la faible taille de la base, on observe une réduction du nombre de variables fractionnaires et du nombre de variables pouvant être non nulles. Ceci a pour effet de diminuer le nombre de noeuds de branchement nécessaires pour trouver une solution entière optimale. Mais, cela reste inefficace quant à la diminution de la taille du sous-problème. La plus récente version de l'algorithme de l'agrégation dynamique (El Hallaoui, 2006) propose une réduction de la taille du réseau sous-jacent au sous-problème en éliminant un certain nombre d'arcs et ce suivant leur degré d'incompatibilité avec la partition  $Q$ . Cette réduction est dynamique au sens qu'elle est révisée à chaque itération de la génération de colonnes. De plus, elle n'est pas activée à la dernière itération pour garantir l'optimalité. L'analyse des résultats de cette approche montre qu'elle est capable d'atteindre l'intégrité plus rapidement et en utilisant nettement moins de noeuds de branchement que les versions précédentes. Il faudrait cependant souligner que le nombre de décisions avantageuses qui sont prises est étroitement lié à la qualité de la partition initiale et l'efficacité des stratégies d'agrégation-désagrégation utilisées. Des tests préliminaires ont toutefois démontré que la réduction dynamique des réseaux n'est pas avantageuse pour

le problème traité. Par conséquent, cette technique n'a pas été employée pour obtenir les résultats présentés dans la section 5.3.

L'agrégation dynamique de contraintes est une nouvelle approche qui permet de réduire la taille du problème maître, du sous-problème ainsi que la dégénérescence. Elle fonctionne avec des problèmes maîtres et des sous-problèmes de taille plus petite pour améliorer la solution primale. Cette méthode permet d'obtenir une solution optimale primale à la fin de la résolution de la relaxation linéaire du problème considéré. C'est un avantage important par rapport aux autres méthodes de réduction du nombre de contraintes telles que la relaxation lagrangienne qui convergent généralement vers des solutions non réalisables.

#### **4.2.3 Spécialisation au problème d'horaires mensuels des membres d'équipages**

Sachant que le processus de résolution par agrégation dynamique de contraintes nécessite une solution initiale de bonne qualité, dans le cadre de ce mémoire, cette solution a été fournie par l'algorithme Tabou décrit précédemment. Notons que chaque horaire confectionné lors de la solution initiale représente une séquence de rotations et l'ensemble de ces séquences forme la partition  $Q$  initiale. Afin d'utiliser la version MPDCA, les réseaux utilisés par la méthode de génération de colonnes doivent être modifiés afin de bien comptabiliser le nombre d'incompatibilités d'une colonne (voir El Hallaoui, 2006).

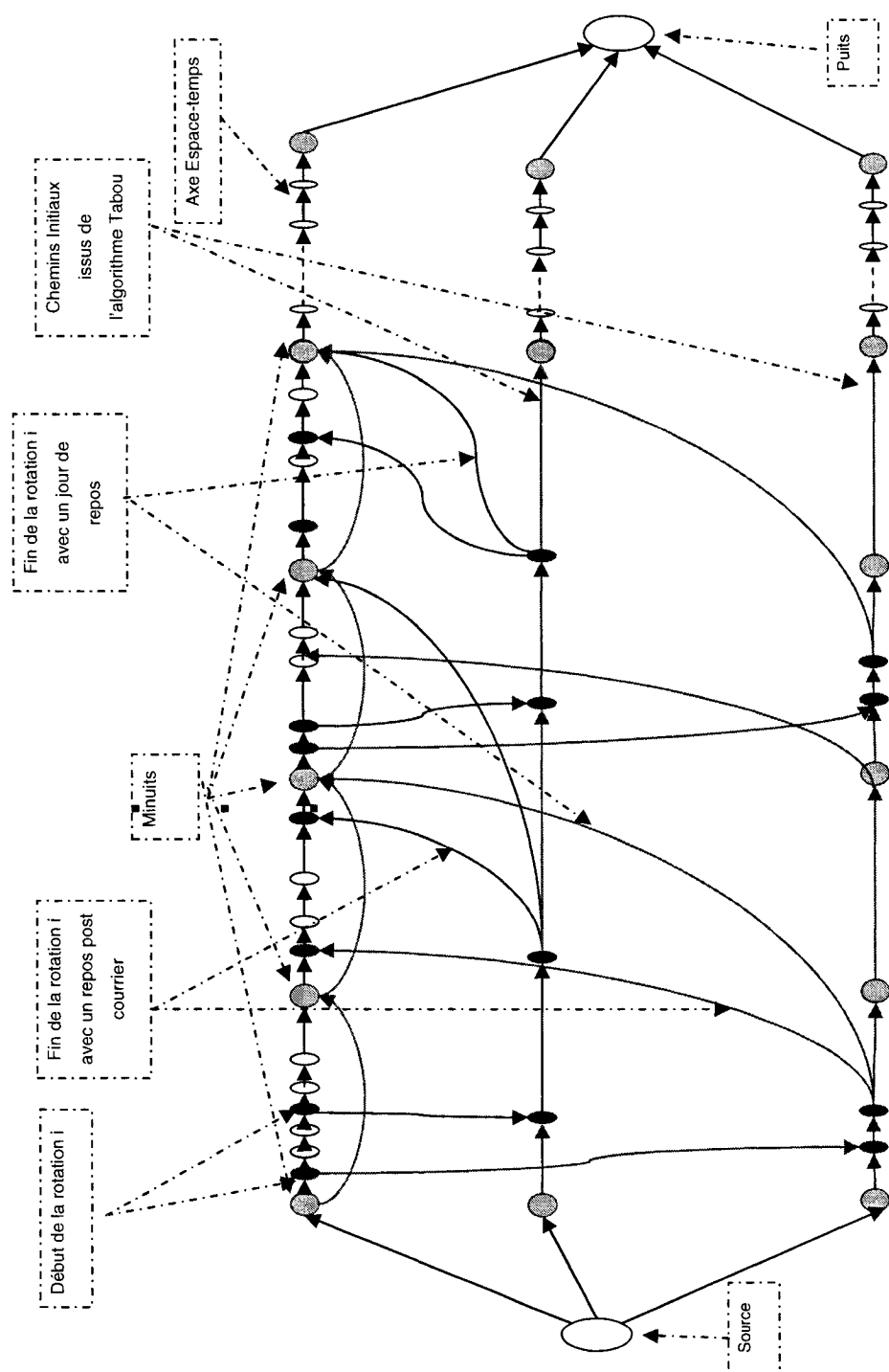
Le modèle des réseaux utilisés pour la méthode d'agrégation dynamique de contraintes présenté dans la figure 4.3 ressemble sensiblement à celui de la génération de colonnes classique. La seule différence majeure est que la solution fournie par l'algorithme Tabou est représentée dans le modèle par des

chemins initiaux. Pour ce faire, l'arc d'une rotation a été subdivisé en deux arcs. Le premier arc fait la liaison entre le nœud de début de la rotation dans l'axe espace-temps et celui du chemin initial et le deuxième arc modélise le début et la fin en comptabilisant la consommation des crédits et des jours travaillés pour la rotation en question. D'un autre côté, un certain nombre de nœuds Minuit apparaissent dans les chemins initiaux dépendamment de l'horaire de chaque employé.

Il faudrait noter que pour l'uniformisation de la solution, on a utilisé la même approche que celle utilisée lors de la génération de colonnes classique. La seule différence notable entre les différents réseaux réside donc dans les plages de validité des ressources et le coût qui leur est associé.

Les ressources utilisées pour ce modèle sont les mêmes que celles utilisées pour la génération de colonnes classique. Cependant, l'ajout d'une nouvelle ressource a été nécessaire afin de limiter le nombre d'incompatibilités pour chaque horaire (voir El Hallaoui, 2006). Les tableaux 4.4 et 4.5 illustrent la plage de validité et la consommation des ressources (à l'exception de la nouvelle ressource) pour chacun des nœuds et arcs du réseau. Les plages de validité de la nouvelle ressource sont toutes les mêmes pour tous les nœuds, soit  $[0, \kappa]$  où  $\kappa$  indique le nombre maximum d'incompatibilités tolérés dans la phase courante. La consommation de cette ressource vaut 1 sur tous les arcs permettant de dévier de la partition courante et 0 sinon.





**Figure 4.3 : Exemple de réseau utilisé pour l'agrégation dynamique de contraintes**

Tableau 4.4 : Plages de validité des ressources

Plage de la ressource Noeuds du réseau	Nombre maximal de crédits	Nombre minimal de crédits	Nombre maximal de jours de congé	Nombre minimal de jours de congé	Nombre de jours travaillés consécutifs	Ajustement des jours de congé
Source	$[0, \text{Max}^{\text{credits}}_z]$	$[\text{Min}^{\text{credits}}_z, \text{Min}^{\text{credits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[\text{Min}^{\text{congé}}_z, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}_z]$	$[0, 0]$
Minuit (Chemin initial) Minuit (espace-temps)	$[0, \text{Max}^{\text{credits}}_z]$	$[0, \text{Min}^{\text{credits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}_z]$	$[0, 0]$
Début_Tâche (espace-temps) Fin_Post_courrier (espace-temps) Fin_Avec_Congé (espace-temps)	$[0, \text{Max}^{\text{credits}}_z]$	$[0, \text{Min}^{\text{credits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}_z]$	$[0, 1]$
Début_Tâche (Chemin initial) Fin_Tâche (Chemin initial)	$[0, \text{Max}^{\text{credits}}_z]$	$[0, \text{Min}^{\text{credits}}_z]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, \text{Min}^{\text{congé}}_z]$	$[0, \text{Max}^{\text{travaillés}}_z]$	$[0, 0]$
Puits	$[0, \text{Max}^{\text{credits}}_z]$	$[0, 0]$	$[0, \text{Max}^{\text{congé}}_z]$	$[0, 0]$	$[0, \text{Max}^{\text{travaillés}}_z]$	$[0, 0]$

**Tableau 4.5 : Consommation des ressources**

<div> <div>Ressource consom- mée</div> <div>Arcs du réseau</div> </div>	Nombre maximal de crédits	Nombre minimal de crédits	Nombre maximal de jours de congé	Nombre minimal de jours de congé	Nombre de jours travaillés consécutifs	Ajustement des jours de congé
Arcs d'attente	0	0	0	0	+1 si départ d'un Minuit et 0 sinon	+1 si départ d'un Minuit et 0 sinon
Minuit - Minuit	0	0	1	- 1	Remise à 0	0
Début_Tâche (espace-temps) - Début_Tâche (Chemin initial)	0	0	0	0	0	-1
Début_Tâche (Chemin initial) - Fin_Tâche (Chemin initial)	Crédits de la tâche i	- (Crédits de la tâche i)	0	0	+ 1 pour chaque minuit rencontré par la tâche	0
Fin_Tâche (Chemin initial) - Fin_Post_courrier (espace-temps)	0	0	0	0	+1 si passage par Minuit et 0 sinon	0
Fin_Tâche (Chemin initial) - Fin_Avec_Congé (espace-temps)	0	0	1	- 1	Remise à 0	0
Source - Minuit	0	0	0	0	0	0
Minuit - Puits	0	0	0	0	0	0

## **CHAPITRE 5 : EXPÉRIMENTATIONS NUMÉRIQUES**

Afin de valider les différentes étapes du développement de la nouvelle approche adoptée, on a utilisé un jeu de données correspondant au mois d'août 2005 provenant d'un client de la compagnie Kronos Inc. Notons que les règles de validité des horaires ont été précisés dans la section 4.1.3. Ce jeu de données comporte 2924 rotations et 565 employés. Afin d'effectuer plusieurs tests pour différentes instances, on procède à un choix aléatoire parmi les données initiales afin de créer des jeux de données de différentes tailles. Par conséquent, nous avons réalisé des tests sur des instances comportant 291, 506, 700, 974, 1187, 1507, 2165 et 2924 rotations et 55, 92, 134, 188, 228, 290, 417, 565 employés, respectivement. Notons que les tests pour la génération de colonnes et l'agrégation dynamique de contraintes ont été effectués sur une machine PC Linux 2.6 GHz et les tests de la méthode Tabou sur un PC Windows Pentium4 2.0 GHz. Dans ce chapitre, on va présenter initialement les résultats des différentes approches de résolution en l'occurrence, la méthode Tabou, la génération de colonnes et l'agrégation dynamique. Puis, on achève par une comparaison entre les trois méthodes d'un point de vue des temps de calcul et de la qualité des solutions produites.

### **5.1 Résultats de l'algorithme Tabou**

L'algorithme Tabou a été conçu principalement pour fournir une solution initiale pour l'algorithme d'agrégation dynamique de contraintes. Par contre, cet algorithme pourrait aussi être utilisé seul comme méthode de résolution pour

tout le problème. Dans cette section, nous présentons les résultats dans ces deux contextes.

### 5.1.1 Approche pour trouver une solution initiale

Pour ces tests, l'algorithme Tabou est stoppé 2500 itérations après avoir trouvé une première solution réalisable. Il faudrait cependant noter que les paramètres de l'algorithme ont été ajustés pour trouver la meilleure solution possible pour le nombre d'itérations fixé.

Les résultats obtenus pour chacune des instances testées sont présentés dans le tableau 5.1.

**Tableau 5.1 : Algorithme Tabou pour solution initiale - temps de calcul**

Nombre de rotations/employés	Nombre d'itérations total	Temps total (s)	Temps de la 1 <sup>ère</sup> solution réalisable (s)
291/55	6317	25.3	10.5
506/92	5789	35.4	14.7
700/134	6366	45.5	22.7
974/188	6325	61.2	31.4
1187/228	7089	81.3	44.9
1507/290	6338	95.2	50.2
2165/417	6150	131.4	68
2924/565	7227	207.9	120.9

L'analyse des résultats du tableau 5.1 démontre la linéarité du temps total ainsi que le temps de la première solution de l'algorithme et ce par rapport à la taille des instances considérées. Cela démontre la robustesse de l'algorithme Tabou utilisé dans le cadre de ce mémoire.

Dans les tableaux 5.2 et 5.3, nous présentons la qualité des solutions élaborées par l'algorithme Tabou par rapport à l'uniformisation des crédits travaillés et des jours de congé. Dans ces tableaux, les valeurs associées à la première solution réalisable trouvée par l'algorithme sont placées entre parenthèses et les autres valeurs correspondent à la solution finale obtenue. Pour chaque instance, on indique les valeurs minimale, maximale et moyenne du nombre de crédits ou de jours de congé par horaire et l'écart type correspondant.

**Tableau 5.2 : Algorithme Tabou pour solution initiale – statistiques sur le nombre de crédits**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Écart Type Cr
291/55	(3931) 4162	(5077) 4983	4528	(308.7) 192.6
506/92	(3903) 3935	(5081) 5094	4523	(340.7) 221.8
700/134	(3933) 3917	(5097) 5063	4520	(327.6) 217.7
974/188	(3902) 4004	(5090) 5079	4502	(307.1) 235.7
1187/228	(3914) 4066	(5072) 5094	4509	(327.1) 221.4
1507/290	(3904) 3925	(5097) 5097	4509	(322.3) 224.3
2165/417	(3900) 4037	(5099) 5098	4509	(325.6) 214.4
2924/565	(3900) 3930	(5099) 5099	4506	(334.9) 232

**Tableau 5.3 : Algorithme Tabou pour solution initiale – statistiques sur les jours de congé**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Nombre total des jours de congé	Écart Type J.C.
291/55	(10) 11	(18) 14	(13.05) 13.00	(718) 715	(1.64) 0.66
506/92	(10) 11	(18) 15	(13.57) 13.46	1249 1239	(1.97) 0.81
700/134	(10) 10	(19) 15	(13.57) 13.47	(1805) 1790	(1.89) 0.87
974/188	(10) 11	(20) 18	(13.55) 13.44	(2549) 2527	(1.87) 0.81
1187/228	(10) 10	(18) 16	(13.58) 13.48	(3092) 3075	(1.89) 0.89
1507/290	(10) 10	(18) 15	(13.48) 13.38	(3898) 3867	(1.81) 0.92
2165/417	(10) 10	(20) 16	(13.47) 13.39	(5607) 5571	(1.78) 0.86
2924/565	(10) 10	(20) 17	(13.50) 13.44	(7619) 7583	(1.87) 0.93

Ces résultats montrent une amélioration notable de la qualité entre la première solution et la solution finale. En effet, on remarque que les valeurs minimales et maximales sont resserrées autour de la valeur moyenne. Cependant, on remarque que le nombre de jours de congé totaux diminue chose qui pourrait éventuellement être évité en introduisant une pénalité au niveau de la fonction objective de l'algorithme Tabou. D'un autre côté, la valeur des écarts types est relativement stable mais encore on remarque une diminution de près de 35 % en moyenne pour les crédits travaillés et plus que 50 % selon les cas pour les jours de congé. Cela vient confirmer l'efficacité de l'algorithme Tabou pour

trouver une solution initiale d'assez bonne qualité dans un temps relativement court.

### 5.1.2 Approche en tant que méthode de résolution

Dans cette section, on considère l'utilisation unique de la méthode Tabou pour trouver une solution au problème traitée. La durée pendant laquelle on a exécuté l'algorithme est la somme de la durée trouvée par la méthode de l'agrégation dynamique (voir section 5.3) et de celle requise pour trouver la solution initiale par l'algorithme Tabou. Il faudrait noter qu'on observe une forte augmentation du nombre d'itérations pour les grandes instances. Mais, les paramètres de l'algorithme Tabou n'ont pas été fortement réajustés pour ce contexte. De ce fait, les résultats obtenus peuvent être améliorés avec une meilleure adaptation aux temps d'exécution choisis.

Les résultats sont présentés dans les tableaux 5.4 à 5.6 qui sont similaires aux tableaux 5.1 à 5.3 de la section précédente. La dernière colonne des tableaux 5.5 et 5.6 indique les gains en pourcentage des écarts types par rapport à ceux des tableaux 5.2 et 5.3.

**Tableau 5.4 : Algorithme Tabou seul - temps de calcul**

Nombre de rotations/employés	Nombre d'itérations total	Temps total (s)	Temps de la 1 <sup>ère</sup> solution réalisable (s)
291/55	15777	75	13.7
506/92	13092	100	13.8
700/134	20985	185	20.8
974/188	20363	225	46.1
1187/228	32471	360	187.9
1507/290	25341	500	91.14
2165/417	51054	1300	77.6
2924/565	88327	3100	138.9



**Tableau 5.5 : Algorithme Tabou seul – statistiques sur le nombre de crédits**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Écart Type Cr	Gain (%)
291/55	4229	5084	4528.3	174.94	8.8
506/92	4246	5046	4523.1	141.62	36.2
700/134	4198	5033	4520.9	172.42	20.7
974/188	4223	5061	4502.2	153.25	34.8
1187/228	4152	4997	4509.2	161,86	27.1
1507/290	4189	5087	4509.9	155.1	30.8
2165/417	4083	5082	4509.6	159.25	25.8
2924/565	4150	5099	4506.4	166.8	28.1

D'après les tableaux des résultats, on constate qu'en donnant beaucoup plus de temps à la méthode Tabou, elle permet d'améliorer la qualité des solutions trouvées du point de vue de l'uniformisation du nombre de crédits travaillés. Notons cependant que les paramètres de l'algorithme ont été ajustés légèrement, en donnant plus de poids à l'uniformisation des crédits au détriment du nombre de jours de congé et ce car une étude du comportement de l'algorithme a prouvé que, pour un nombre plus important d'itérations, l'algorithme est plus robuste en équilibrant un peu plus l'uniformisation des crédits travaillés. D'un autre côté, la détérioration relative de l'écart type pour les jours de congé n'est pas très significative car les valeurs minimales et maximales du nombre de jours de congé pour chaque instance restent sensiblement les mêmes pour les deux approches.

**Tableau 5.6 : Algorithme Tabou seul – statistiques sur le nombre de jours de congé**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Nombre total des jours de congé	Moyenne	Écart Type J.C.	Gain (%)
291/55	10	15	718	13.05	0.77	-16.6
506/92	10	16	1240	13.47	0.93	-14.8
700/134	11	17	1798	13.51	1.08	-24.1
974/188	10	16	2535	13.48	1.03	-27.1
1187/228	11	17	3085	13.53	1.01	-13.4
1507/290	10	16	3885	13.44	0.97	-5.4
2165/417	10	16	5572	13.39	0.98	-13.9
2924/565	10	18	7591	13.45	1.05	-12.9

## **5.2 Résultats de la génération de colonnes**

Pour résoudre les problèmes par génération de colonnes, nous avons fait appel au logiciel GENCOL qui a été développé au GERAD. Les résultats obtenus sont présentés dans les tableaux 5.7 à 5.9. Dans le tableau 5.7, on indique pour chaque instance le temps total, les temps requis pour le problème maître et les sous-problèmes, le nombre de nœuds de branchement et le nombre de variables fractionnaires dans la solution de la relaxation linéaire.

**Tableau 5.7 : Génération de colonnes – temps de calcul**

Nombre de rotations/employés	Temps Total (s)	Temps du problème maître (s)	Temps des sous-problèmes (s)	Nombre de nœuds de branchement	Nombre de variables fractionnaires
291/55	48	9.5	33.4	87	292
506/92	637.6	149.4	420	146	507
700/134	934	334.2	500.1	194	694
974/188	3072.8	1362.6	1291.3	270	975
1187/228	8073.1	3065.6	3807.3	328	1179
1507/290	13926.8	6265.0	5349.0	404	1488
2165/417	56535.8	29156.2	19020.1	540	2154
2924/565	183175.8	77305.3	71855.8	693	2914

On remarque que les temps de calcul totaux augmentent rapidement avec la taille des instances. On remarque aussi que la répartition des temps de calcul entre le problème maître et les sous-problèmes est équitable. D'un autre côté, le nombre de nœuds de branchement est relativement élevé et ce du fait que les solutions des relaxations linéaires sont très fractionnaires.

Les tableaux 5.8 et 5.9 présentent une étude qualitative des solutions trouvées pour la méthode de génération de colonnes. Dans chacun des tableaux, la dernière colonne présente les gains des écarts types par rapport à ceux des tableaux 5.5 et 5.6.

**Tableau 5.8 : Génération de colonnes – statistiques sur le nombre de crédits**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Écart Type Cr	Gain (%)
291/55	4288	4719	4528.3	85.49	51.1
506/92	4381	4734	4523	92.62	34.6
700/134	4288	4832	4520	109.98	36.2
974/188	4301	4758	4502	91.18	40.5
1187/228	4380	4805	4509	87.31	46.1
1507/290	4266	4737	4509	101.89	34.3
2165/417	4289	4795	4509	97.22	38.9
2924/565	4267	4850	4506	112.33	32.6

**Tableau 5.9 : Génération de colonnes – statistiques sur le nombre de jours de congé**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Nombre total des jours de congé	Écart Type J.C.	Gain (%)
291/55	11	16	13.14	723	1.01	-31.1
506/92	11	16	13.59	1251	0.99	-6.4
700/134	11	16	13.67	1819	1.02	5.5
974/188	11	16	13.66	2568	0.89	13.6
1187/228	12	16	13.70	3124	0.83	17.8
1507/290	11	16	13.69	3959	1.00	-3.1
2165/417	11	16	13.70	5701	1.03	-5.1
2924/565	11	17	13.76	7764	1.06	-0.9

On remarque qu'il y a, à nouveau, une stabilité dans les écarts types ainsi que pour les valeurs minimales et maximales. Notons que la qualité des solutions reste meilleure que celle obtenue par l'algorithme Tabou surtout pour l'uniformisation du nombre de crédits.

### ***5.3 Résultats de l'agrégation dynamique de contraintes***

La méthode d'agrégation dynamique de contraintes utilisée a été implantée dans le logiciel GENCOL. Notons que, dans un souci d'équité, les mêmes valeurs de paramètres que celles utilisés pour la génération de colonnes ont été retenues. La version de l'agrégation dynamique utilisée est celle avec multiphase et le processus de résolution s'arrête à la phase 2 (i.e., aucune colonne avec plus de 2 incompatibilités ne peut être générée). Notons que des tests préliminaires ont été effectués avec la version incluant la phase  $\infty$  et l'analyse des résultats a montré que la qualité des solutions trouvées est semblable à celle obtenue après la phase 2. D'un autre côté, le gain en temps de calcul est substantiel et justifie pleinement l'option adoptée. À titre d'exemple, le temps de calcul total pour l'instance la plus grande passe de 50 heures à moins d'une heure.

Les résultats sont présentés dans les tableaux 5.10 à 5.12 qui sont similaires aux tableaux 5.7 à 5.9 de la section précédente. Dans le tableau 5.10, la troisième colonne donne la proportion de temps de résolution dédiée au calcul de la solution initiale par la méthode Tabou. Dans les tableaux 5.11 et 5.12, la dernière colonne présente les gains des écarts types par rapport à ceux des tableaux 5.8 et 5.9.

**Tableau 5.10 : Agrégation dynamique de contraintes – temps de calcul**

Nombre de rotations /employés	Temps Total (s)	% du temps passé pour la solution initiale	Temps du problème maître (s)	Temps des sous-problèmes (s)	Nombre de nœuds de branchement	Nombre de variables fractionnaires
291/55	47.9	34.5	1.3	43.3	52	165
506/92	59.9	37.1	1.0	53.9	64	181
700/134	139.2	24.6	2.6	122.1	107	251
974/188	165.0	27.0	3.7	143.6	136	261
1187/228	281.0	22.4	6.6	236.7	156	344
1507/290	396.8	19.3	6.8	341.3	191	374
2165/417	1143.3	10.3	35.5	992.2	317	617
2924/565	2817.3	6.8	61.8	2345.4	440	608

On remarque qu'il y a une chute des temps du problème maître ainsi que des temps des sous-problèmes. Cela est dû à un nombre plus petit d'itérations résultant de l'utilisation de la solution initiale de l'algorithme Tabou, au fait que les phases 0, 1 et 2 sont moins coûteuses pour les sous-problèmes car la limite sur le nombre d'incompatibilités admissibles réduit de beaucoup le nombre de chemins admissibles, et aussi à l'arrêt prématuré de la génération de colonnes après la phase 2. D'un autre côté, notons que le nombre de nœuds de branchement est nettement inférieur car les solutions des relaxations linéaires obtenues par l'agrégation dynamique de contraintes sont beaucoup moins fractionnaires que celles obtenues par génération de colonnes. Le fait que les

solutions relaxées sont moins fractionnaires permet de prendre de meilleures décisions heuristiques de branchement et donne de meilleures solutions entières. À titre d'exemple, pour la plus petite instance, l'agrégation dynamique de contraintes a confectionné les horaires en utilisant uniquement le réseau avec les plages de validité les plus strictes alors que la méthode de génération de colonnes a utilisé un réseau ayant des plages de validité plus larges à 5 reprises. Finalement, la décroissance du pourcentage de temps passé dans l'élaboration de la solution initiale montre la validité des choix qu'on a pris pour le développement des solutions initiales par la métaheuristique Tabou.

**Tableau 5.11 : Agrégation dynamique de contraintes – statistiques sur le nombre de crédits**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Moyenne	Écart Type Cr	Gain (%)
291/55	4394	4619	4528	67.33	21,2
506/92	4383	4665	4523	71.71	22.6
700/134	4381	4694	4521	72.23	34.3
974/188	4280	4715	4502	73.41	19.5
1187/228	4381	4739	4509	65.98	24.4
1507/290	4358	4730	4509	72.33	29.0
2165/417	4380	4727	4509	75.66	22.1
2924/565	4316	4740	4506	76.79	31.6

**Tableau 5.12 : Agrégation dynamique de contraintes – statistiques sur le nombre de jours de congé**

Nombre de rotations/employés	Valeur Minimale	Valeur Maximale	Nombre total des jours de congé	Moyenne	Écart Type J.C.	Gain (%)
291/55	12	14	721	13.10	0.77	23.7
506/92	12	15	1247	13.55	0.74	25.2
700/134	12	16	1799	13.52	0.79	22.5
974/188	12	16	2532	13.47	0.80	10.1
1187/228	12	15	3082	13.51	0.77	7.2
1507/290	12	16	3881	13.42	0.83	17.0
2165/417	12	16	5587	13.43	0.83	19.4
2924/565	12	16	7597	13.47	0.88	16.9

#### **5.4 Comparaison des trois méthodes**

Le tableau 5.13 présente une récapitulation des résultats obtenus par les trois approches de résolution. Notons que pour l'agrégation dynamique de contraintes, le temps présenté correspond au temps total incluant celui de la solution initiale par l'algorithme Tabou.



**Tableau 5.13 : Étude comparative des trois approches de résolution**

Nombre de rotations /employés	Méthode Tabou			Génération de colonnes			Agrégation dynamique de contraintes		
	Temps (s)	Écart type Cr	Écart type J.C.	Temps (s)	Écart type Cr	Écart type J.C.	Temps (s)	Écart type Cr	Écart type J.C.
291/55	75	174.9	0.77	48	85.49	1.01	73.2	67.3	0.77
506/92	100	141.6	0.93	637.6	92.62	0.99	95.3	71.7	0.74
700/134	185	172.4	1.08	934	109.98	1.02	184.7	72.2	0.79
974/188	225	153.2	1.03	3072.8	91.18	0.89	226.2	73.4	0.80
1187/228	360	161,8	1.01	8073.1	87.31	0.83	362.3	65.9	0.77
1507/290	500	155.1	0.97	13926.8	101.89	1.00	492	72.3	0.83
2165/417	1300	159.2	0.98	56535.8	97.22	1.03	1274.7	75.6	0.83
2924/565	3100	166.8	1.05	183175.8	112.33	1.06	3025.2	76.7	0.88

L'analyse des résultats du tableau 5.13 montre que la méthode de l'agrégation dynamique de contraintes est nettement supérieure aux autres méthodes pour les instances considérées. En effet, par rapport à l'algorithme Tabou, elle offre une meilleure qualité de solution pour des temps similaires. Par rapport à la génération de colonnes, l'approche de résolution adoptée présente des gains en temps de calcul considérables qui atteignent 98.3% et ce jumelé à une qualité de solution améliorée.

Le tableau 5.14 présente une revue des améliorations de la méthode de l'agrégation dynamique par rapport à la génération de colonnes et ce du point de vue des temps de résolution et de la qualité des solutions.

**Tableau 5.14 : Gain de la combinaison de la méthode Tabou associée à l'agrégation dynamique vs génération de colonnes**

Nombre de rotations/employés	Facteur de réduction du temps de calcul	% réduction pour l'écart type du nombre de crédits	% réduction pour l'écart type pour les jours de congés
291/55	0.6	21,2	23.7
506/92	6.7	22.6	25.2
700/134	5.0	34.3	22.5
974/188	13.5	19.5	10.1
1187/228	22.3	24.4	7.2
1507/290	28.3	29.0	17.0
2165/417	44.3	22.1	19.4
2924/565	60.5	31.6	16.9

L'analyse des résultats du tableau 5.14 montre que les facteurs de réduction sont impressionnants et réaffirment la validité de l'approche de résolution adoptée. D'un autre côté, la croissance des facteurs de résolution avec la taille des instances jumelée avec une qualité de la solution améliorée permet d'envisager de résoudre des problèmes de plus grande taille. En effet, la croissance plus lente du temps de calcul avec l'agrégation dynamique permet d'espérer résoudre en quelques heures des problèmes de 3000 à 6000 rotations vis-à-vis 1500 rotations pour la génération de colonnes.

Puisque le processus de résolution de l'algorithme MPDCA utilisé est stoppé à la phase numéro deux, l'optimalité de la solution trouvée n'est pas garantie. De ce fait, les solutions obtenues sont heuristiques et peuvent être améliorées au détriment du temps de calcul. Mais, la qualité des solutions trouvées est acceptable et devance celles trouvées par la version heuristique de la génération de colonnes.

D'un autre côté, malgré le fait que la performance de l'algorithme Tabou peut être améliorée, on n'aura aucune garantie d'optimalité à la fin du processus de résolution.

En dernier lieu, les résultats présentés montrent qu'une métaheuristique jumelée avec une méthode exacte, en l'occurrence l'agrégation dynamique de contraintes, s'avère une méthode robuste et efficace pour la construction d'horaires mensuels pour les compagnies aériennes.

## CHAPITRE 6 : CONCLUSION

Dans ce mémoire, nous avons traité de la construction d'horaires pour les membres d'équipages d'une compagnie aérienne et l'objectif principal est d'uniformiser les horaires construits suivant les jours de congé et les heures travaillés. Vu la difficulté du problème, nous avons utilisé une nouvelle approche de résolution en l'occurrence l'agrégation dynamique de contraintes. Cette approche nécessite toutefois une solution initiale ce qui nous a amenés à développer une heuristique générique basée sur le coloriage de graphe et utilisant la recherche locale avec liste Tabou.

Il faudrait noter que l'approche de résolution adoptée est considérée à l'avant-garde dans le domaine de la recherche opérationnelle. En effet, l'agrégation dynamique de contraintes est considérée comme un nouveau paradigme et le fait de la jumeler avec une métaheuristique en l'occurrence l'algorithme Tabou est une première en tant que telle. Cette approche a prouvé sa robustesse et son efficacité de part les tests qu'on a effectués dans ce mémoire. En effet, on a observé une réduction par un facteur de 60 les temps de calcul par rapport à l'approche traditionnelle de génération de colonnes et ce avec des solutions de qualité nettement améliorées. D'un autre côté, pour des temps semblables, l'agrégation dynamique de contraintes donne des solutions de qualité supérieure à celles d'une métaheuristique Tabou.

Finalement, les travaux initiés dans ce mémoire ouvre une porte à une application plus approfondie de l'agrégation dynamique de contraintes au problème de construction d'horaires mensuels pour les compagnies aériennes.

En effet, dans le cadre de ce mémoire les employés sont pris anonymement sans tenir compte des préférences ainsi que des compétences de chacun. Une prochaine étape pour tester l'efficacité de l'agrégation dynamique serait d'inclure les préférences des employés lors de la confection des horaires et ce pour la construction d'horaires personnalisés dans un cadre d'équité ou dans un cadre de séniorité stricte.

## RÉFÉRENCES

ACHOUR, H. (2005). Fabrication d'horaires personnalisés avec priorités pour les équipages aériens, Thèse de doctorat, École Polytechnique de Montréal, Canada.

AHUJA, R. K. et JAMES, B. (2002). "Very Large-Scale Neighborhood Search in Airline Fleet Scheduling". *SIAM News*, 35, 9.

BALL, M. et ROBERTS, A. (1985). A Graph Partitioning Approach to Airline Crew Scheduling. *Transportation Science*, 19(2), 107-126.

BAKER, E.K., BODIN, L.D., FINNEGAN, W.F. et PONDER, R.J. (1979). Efficient Heuristic Solutions to an Airline Crew Scheduling Problem. *AIE Transactions*, 11(2), 79-85.

BARNHART, C., JOHNSON, E., NEMHAUSER, G., SAVELSBERGH, M. et VANCE, P. (1998a). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research*, 46, 316-329.

BARNHART, C., JOHNSON, E., NEMHAUSER, G. et VANCE, P. (1999). Crew Scheduling. "Handbook of Transportation Science". *R. W. Hall. Kluwer Scientific Publishers*, 493-521.

BARNHART, C. et SHENOI, R.G. (1998b). An Approximate Model and Solution Approach for the Long-Haul Crew Pairing Problem. *Transportation Science*, 32(3), 221-231.

BELANGER, N., DESAULNIERS, G., SOUMIS, F., DESROSIERS, J. et LAVIGNEC, J. (2006). Weekly airline fleet assignment with homogeneity, *Transportation Research*. 40 (4), 306-318.

CHRISTOU, I., ZAKARIAN, A., LIU, J. et CARTER, H. (1999). A Two-Phase Genetic Algorithm for Large-Scale Bidline-Generation Problems at Delta Air Lines. *Interfaces* 29(5), 51-65.

COMBS, T.E., (2002). A Combined Adaptive Tabu Search and Set Partitioning Approach for the Crew Scheduling Problem with an Air Tanker Crew Application. Ph.D. Dissertation, Department of Operational Sciences, Air Force Institute of Technology.

DANTZIG, G. et WOLFE, P. (1960). The Decomposition Algorithm for Linear Programming. *Operations Research*, 8, 101-111.

DEIRMENDJIAN, L. (1999). Réduction des réseaux pour le problème d'horaires de personnel aériens. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., MARC, S., RIOUX, B., SOLOMON, M.M., et SOUMIS, F. (1997a). Crew Pairing at Air France. *European Journal of Operational Research*, 97, 245-259.

DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., SOLOMON, M.M., et SOUMIS, F. (1997b). Daily Aircraft Routing and Scheduling. *Management Science*, 43, 841-855.

DESAULNIERS, G., DESROSIERS, J., GAMACHE, M. et SOUMIS, F. (1998a). Crew Scheduling in Air Transportation. "Fleet Management and Logistics", Éditeurs : T.G. Crainic et G. Laporte, , Kluwer Academic Publishers, Boston, 169-185.

DESAULNIERS, G., DESROSIERS, J., IOACHIM, I., SOLOMON, M.M., SOUMIS, F. et VILLENEUVE, D. (1998b). A Unified Framework for Time Constrained Vehicle Routing and Crew Scheduling Problems. "Fleet Management and Logistics". Éditeurs: T. Crainic and G. Laporte , , Kluwer, Norwell, 57-93.

DESAULNIERS, G., DESROSIERS, et SOLOMON, M.M. (2002) .Accelerating Strategies in Column Generation Methods for Vehicle Routing and Crew Scheduling Problems. "Essays and Surveys in Metaheuristics". Éditeurs: C.C. Ribeiro and P. Hansen, Kluwer, Norwell, 309-324.

EL HALLAOUI, I., (2006) Agrégation dynamique des contraintes de partitionnement en génération de colonnes. Thèse de doctorat, École Polytechnique de Montréal, Canada.

EL HALLAOUI, I., VILLENEUVE, D., DESAULNIERS, G. et SOUMIS, F. (2005). Dynamic Aggregation of Set-Partitioning Constraints in Column Generation. *Operations Research*, 53(4), 632-646.

FAHLE, T., JUNKER, U., KARIACH, S., KOHL, N., SELLMANN, M. et VAABEN, B. (2002). Constraint Programming Based Column Generation for Crew Assignment. *Journal of Heuristics* 8, 59 – 81.

FEDERICI, F. et PASCHINA, D. (1990). Automated Rostering Model. *AGIFORS Symposium Proceedings*, 12.



GAMACHE, M. (1995). Fabrication d'horaires mensuels pour les membres d'équipage en transport aérien. Thèse de doctorat, École Polytechnique de Montréal, Canada.

GAMACHE, M., SOUMIS, F., MARQUIS, G. et DESROSIERS, J. (1999). A Column Generation Approach for Large Scale Aircrew Rostering Problems. *Operations Research*, 47(2), 247-262.

GAMACHE, M., HERTZ, A., OUELLET, J. (2005). A Graph Coloring Model for a Feasibility Problem in crew scheduling. À paraître dans *Computers & Operations Research*.

GAMACHE, M., SOUMIS, F., VILLENEUVE, D., DESROSIERS, J. et GELINAS, E. (1998). The Preferential Bidding System at Air Canada. *Transportation Science*, 32(3), 246-255.

GERSHKOFF, I. (1989). Optimizing Flight Crew Schedules. *Interfaces*, 19(4), 29-43.

GONTIER, T. (1985). Longhaul Cabin Crew Assignment. *AGIFORS Symposium Proceedings*, 25, 49-66.

GLANERT, W. (1984). A Timetable Approach to the Assignment of Pilots to Rotations. *AGIFORS Symposium Proceedings*, 24, 369-391.

GRAVES, G.W., MCBRIDE, R.D., GERSHKOFF, I., ANDERSON, D. et MAHIDHARA, D. (1993). Flight Crew Scheduling. *Management Science*, 39(6), 736-745.

HAO, J.K., GALINIER, P. et HABIB, M. (1999). Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes. *Intelligence Artificielle*, 13(2), 283-324.

HOUSOS, E. et ELMROTH, T. (1997). Automatic Optimization of Subproblems in Scheduling Airline Crews. *Interfaces*, 27(5), 68-77.

IRNICH, S. et DESAULNIERS, G. (2005). Shortest Path Problems with Resource Constraints. "Column Generation". *Éditeurs: G. Desaulniers, J. Desrosiers and M.M. Solomon., Springer, NY*, 33-65.

KLABJAN, D. (2005). Large-scale Models in the Airline Industry. "Column Generation". *Éditeurs: G. Desaulniers, J. Desrosiers and M.M. Solomon., Springer, NY*, 163-196.

LEVINE, D. (1996). Application of a Hybrid Genetic Algorithm to Airline Crew Scheduling. *Computers and Operations Research*, 23(6), 547-558.

LOHATEPANONT, M. et BARNHART, C. (2004). Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment. *Transportation Science* 38 (1), 19-32.

MITCHELSON, C. (2003). Nouvel algorithme de découpage pour la construction d'horaires mensuels personnalisés dans un contexte d'équité. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

OUELLET, J. (2004). Une approche tabou pour le problème d'horaires de personnel en transport aérien. Mémoire de maîtrise, École Polytechnique de Montréal, Canada.

RUSSELL, D.J. (1989). Development of an Automated Crew Bid Generation System. *Interfaces*, 19, 44-51.

RYAN, D.M. (1992). The Solution of Massive Generalized Set Partitioning Problems in Air Crew Rostering. *Journal of Operational Research Society*, 43, 459-467.

STOJKOVIC, M et SOUMIS, F. (2001). An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem. *Management Science* 47 (9), 1290-1305.

WILSON, B. J. (1981). BA's Regular O.R. Crew Planning Models for the 1980's. *AGIFORS Symposium Proceedings*, 21, 257-270.