

HEC MONTRÉAL
École affiliée à l'Université de Montréal

**Heuristiques pour le problème conjoint de récupération des appareils et des
passagers en transport aérien**

par
Karine Sinclair

Thèse présentée en vue de l'obtention du grade de Ph.D. en administration
(option Gestion des opérations et de la logistique)

Mars 2016

©Karine Sinclair, 2016

HEC MONTRÉAL

École affiliée à l'Université de Montréal

Cette thèse intitulée :

Heuristiques pour le problème conjoint de récupération des appareils et des passagers en transport aérien

Présentée par :

Karine Sinclair

a été évaluée par un jury composé des personnes suivantes :

HEC Montréal

Président-rapporteur

Jean-François Cordeau

HEC Montréal

Codirecteur de recherche

Gilbert Laporte

HEC Montréal

Codirecteur de recherche

Nom université

Membre du jury

Nom université

Examineur externe

HEC Montréal

Représentant du directeur de HEC Montréal

Résumé

L'utilisation des réseaux en étoile par de nombreuses compagnies aériennes implique qu'une petite perturbation peut avoir un impact significatif sur l'ensemble du réseau. Les perturbations peuvent être causées, entre autres, par des vols retardés ou annulés, la non-disponibilité d'un appareil ou d'un équipage, ou une diminution de la capacité d'atterrissage et de décollage causée par des conditions météorologiques défavorables. Ces perturbations peuvent avoir un impact important sur les opérations planifiées et sur la rentabilité des compagnies aériennes. Ainsi, lorsque de telles perturbations se produisent, les compagnies aériennes doivent effectuer des activités de récupération afin de rétablir l'horaire planifié le plus rapidement possible. Les étapes de récupération sont généralement effectuées de façon séquentielle, c'est-à-dire que la récupération des appareils est d'abord effectuée, suivie de la récupération des équipages et finalement de la récupération des passagers. Toutefois, résoudre les problèmes de récupération de façon séquentielle entraîne une sous-optimisation des solutions. La résolution simultanée de plusieurs étapes de récupération permet donc une meilleure optimisation de la récupération. Cette thèse présente différentes heuristiques pour résoudre le problème conjoint de récupération des appareils et des passagers, tel que présenté lors du Challenge ROADEF 2009.

La première méthode de résolution présentée est une heuristique de recherche à grand voisinage (RGV). L'heuristique alterne entre trois phases : construction, réparation et amélioration. De nombreuses améliorations ont été apportées à chaque phase afin d'explorer de façon plus approfondie l'espace de solutions, telles une phase de destruction et de création des rotations d'appareils. L'heuristique tente également d'accommoder un plus grand nombre de passagers en créant de nouveaux segments de vol et en les insérant dans des rotations d'appareils possédant des périodes de disponibilité. L'heuristique développée a été testée sur les instances du Challenge ROADEF 2009 et a permis d'obtenir les meilleures solutions connues pour 21 des 22 instances.

La seconde méthode de résolution est une heuristique de post-optimisation basée sur la génération de colonnes. Un modèle de programmation mixte a été développé pour représenter le problème conjoint de récupération des appareils et des passagers et différentes opérations sont effectuées afin de réduire la taille du modèle. L'heuristique de post-optimisation est exécutée après l'heuristique RGV décrite au paragraphe précédent. Le problème maître restreint inclut tous les arcs de vols et tous les arcs de passagers obtenus dans la solution RGV et la

relaxation linéaire du problème restreint est résolue. Ensuite, un nombre de variables possédant des coûts réduits négatifs sont incluses dans la relaxation linéaire du modèle qui est résolue à nouveau. L’heuristique arrête après un nombre d’itérations déterminé. Le modèle peut également être modifié pour résoudre les plus grandes instances et ce, en ne considérant que les arcs de passagers. L’heuristique a été testée sur les instances du Challenge ROADEF 2009 et a permis une amélioration significative des solutions pour l’ensemble des instances.

La dernière méthode de résolution présentée intègre l’heuristique de post-optimisation décrite au paragraphe précédent dans un cadre d’horizon roulant. Nous proposons également de nouvelles instances réalistes. Ces instances sont très grandes et comportent un grand nombre de perturbations d’appareils, de vols et de capacités aéroportuaires. Le cadre d’horizon roulant permet, à chaque itération, de ne considérer qu’une partie des itinéraires de passagers. Le nombre limité d’itinéraires considérés permet de résoudre les très grandes instances avec un temps de calcul acceptable et permet de réduire significativement le temps nécessaire pour résoudre les instances du Challenge ROADEF 2009. L’heuristique développée se comporte très bien lorsque le nombre de perturbations est grand.

Mots clés : récupération intégrée, récupération des appareils, récupération des passagers, recherche à grand voisinage, génération de colonnes, horizon roulant.

Abstract

The use of hub-and-spoke networks in the commercial airline industry implies that small disruptions can have a significant impact on the networks. Disruptions can be caused by delayed or cancelled flights, unavailable aircraft or crews, or reductions in the airport departure and arrival capacity caused by unfavorable weather conditions. These disruption can have a substantial impact on the planned operations and expenses. When such disruptions occur, airlines must devise recovery operations so as to return to the planned operations as quickly as possible. The airline recovery problem is usually solved in a sequential manner. The first problem addressed is the aircraft recovery problem, followed by the crew recovery problem, and finally, the passenger recovery problem. However, solving the recovery problems in a sequential manner can lead to sub-optimal solutions. Solving these problems simultaneously will lead to a better optimization of the recovery problem. This thesis presents different heuristics for solving the joint aircraft and passenger recovery problem, as described in the 2009 ROADEF Challenge.

The first method proposed is a large neighbourhood search heuristic (LNS), which alternates between three phases : construction, repair and improvement phase. Many improvements were added within each phase so as to perform a more thorough search of the solution space, such as the destruction and creation of aircraft rotations. The heuristic also attempts to accommodate additional passengers by creating new flight legs and inserting them in aircraft rotations with slack capacity. The heuristic was tested on the instances from the 2009 ROADEF Challenge and was able to obtain the best known solutions for 21 of the 22 instances.

The second solution method is a column generation based post-optimization heuristic. The joint aircraft and passenger recovery problem is presented as a mixed integer programming model and different operations are performed so as to reduce the size of the model. The post-optimisation heuristic is executed after the LNS heuristic described in the previous paragraph. The restricted master problem includes all aircraft and passenger arcs found in the LNS solutions and the linear programming relaxation of the restricted problem is solved. A number of variables with negative reduced costs are then included in the linear relaxation of the model which is solved again. The heuristic stops after a predetermined number of iterations. Then the mixed integer programming model with a limited number of columns is solved. The model can also be modified, by only considering passenger arcs, so as to solve

larger instances. The heuristic was tested on the instances from the 2009 ROADEF Challenge and was able to obtain the best known solutions for all the instances.

The last method proposed integrates the post-optimization heuristic described in the previous paragraph in a rolling horizon framework. We also present new realistic instances which are very large and consider a great number of aircraft, flight and airport capacity disruptions. The rolling horizon framework considers, at each iteration, only part of the passenger itineraries. Because of the limited number of itineraries considered, the heuristic can solve the new very large instances within a reasonable computing time and can also significantly reduce the computing time for the 2009 ROADEF Challenge instances. The heuristic is very effective when there are many disruptions.

Keywords : integrated recovery, aircraft recovery, passenger recovery, large neighbourhood search, column generation, rolling horizon.

Table des matières

Résumé	iii
Abstract	v
Liste des figures	xi
Liste des tableaux	xiii
Remerciements	xiv
1 Introduction	1
2 Revue de la littérature	5
2.1 La planification tactique des opérations	5
2.1.1 La création des horaires de vol	5
2.1.2 L'affectation des différents types d'appareils aux segments de vol	7
2.1.3 La création des itinéraires d'appareils	13
2.1.4 La création des rotations d'équipages	16
2.1.5 L'affectation des rotations aux membres d'équipages	21
2.2 La récupération des opérations	23
2.2.1 La récupération des appareils	24
2.2.2 La récupération des équipages	28
2.2.3 La récupération des passagers	31
2.2.4 La récupération intégrée	33
2.3 Horizon roulant	35
2.4 Challenge ROADEF 2009	38
3 Improvements to a Large Neighbourhood Search Heuristic for an Integrated Aircraft and Passenger Recovery Problem	43
3.1 Introduction	44
3.2 Problem description	47
3.3 Solution method	49

3.3.1	The large neighbourhood search heuristic	50
3.3.2	The improved large neighbourhood search heuristic	53
	3.2.2.1 Multi-commodity flow problem with additional constraints . .	56
	3.2.2.2 Shortest path problem	57
3.4	Computational results	63
3.5	Conclusions	71
4	A Column Generation Post-Optimization Heuristic for the Integrated Aircraft and Passenger Recovery Problem	73
4.1	Introduction	74
4.2	Problem description	77
	4.2.1 Airports	78
	4.2.2 Aircraft	78
	4.2.3 Passengers	79
	4.2.4 Recovery period	80
	4.2.5 Disruptions	80
	4.2.6 Mixed integer linear programming model	82
4.3	Solution methodology	84
	4.3.1 Large neighbourhood search heuristic	85
	4.3.2 Column generation	87
	4.3.3 Model reduction	87
	4.3.4 Multi-commodity flow problem	88
4.4	Computational results	89
4.5	Conclusions	98
4.6	Appendix 1	100
4.7	Appendix 2	103
5	A Rolling Horizon Heuristic for Aircraft and Passenger Recovery	109
5.1	Introduction	110
5.2	Problem description	114
	5.2.1 Airports	114
	5.2.2 Aircraft	114
	5.2.3 Flight legs	115
	5.2.4 Passenger reservations	115
	5.2.5 Disruptions	116
	5.2.6 Problem formulation	116

5.3	Solution methodology and mathematical model	117
5.3.1	Column generation post-optimization heuristic	117
5.3.2	Rolling horizon	123
5.4	Computational results	124
5.4.1	The 2009 ROADEF Challenge Instances	124
5.4.2	Creation of very large instances derived from the 2009 ROADEF Chal- lenge instances	126
5.4.3	Parameter setting	126
5.4.3.1	Number of iterations and percentage of variables with negative reduced cost	127
5.4.3.2	Fixed rolling horizon period	127
5.4.3.3	Additional time period for flights	128
5.4.3.4	Lenght of the rolling horizon	129
5.4.4	Results	130
5.5	Conclusions and future research	132
5.6	Appendix 1	137
5.7	Appendix 2	139
6	Conclusion	143
6.1	Contributions	143
6.2	Perspectives de recherche future	145
	Bibliographie	152

Liste des figures

1.1	Revenus et bénéfices des compagnies aériennes américaines de 2000 à 2014	2
1.2	Rapport des différents coûts sur les revenus des compagnies aériennes américaines de 1995 à 2014	2
3.1	Overview of the LNS solution method	50
3.2	Overview of the additional steps in the improved LNS	53
3.3	Example of a longest path graph	54
3.4	Redirecting passengers	61
3.5	Example delaying subsequent passenger departures	63
4.1	Set of aircraft F partitioned into families B_l , models T_g and configurations V_h	79
4.2	Demand nodes of the network	81
4.3	Overview of the LNS heuristic	85
4.4	Average solution cost vs computing time	95
5.1	Rolling horizon framework	123
5.2	Improvement with respect to best solution value for parameter Δ'	128
5.3	Computing time in seconds for parameter Δ'	128
5.4	Improvement with respect to best solution value for parameter Θ	129
5.5	Computing time in seconds for parameter Θ	129

Liste des tableaux

3.1	Final ranks and scores in the 2009 ROADEF Challenge	64
3.2	Characteristics of the B instances	65
3.3	Characteristics of the XA, XB instances	65
3.4	Characteristics of the X instances	66
3.5	Recovery costs for the B instances (part 1)	67
3.6	Recovery costs for the B instances (part 2)	67
3.7	Recovery costs for the XA instances	68
3.8	Recovery costs for the XB instances	68
3.9	Recovery costs for the X instances	68
3.10	Average gap with best solution with 5 minutes or 10 minutes of computing time	69
3.11	Characteristics of the solution costs	69
3.12	Recovery costs for the B instances (part 1)	69
3.13	Recovery costs for the B instances (part 2)	69
3.14	Recovery costs for the XA instances	70
3.15	Recovery costs for the XB instances	70
3.16	Recovery costs for the X instances	70
3.17	Characteristics of the solution costs	70
4.1	Characteristics of the B instances	89
4.2	Characteristics of the XA, XB and X instances	90
4.3	Characteristics of the aggregated passengers and aircraft	91
4.4	Recovery costs for the APRP	92
4.5	Variation in the different costs	92
4.6	Recovery cost for the MC-APRP	93
4.7	Characteristics of the solution costs	94
4.8	Recovery costs for the multi-commodity flow problem for parameter set P2 .	94
4.9	Recovery costs for parameter sets P2	96

4.10	Characteristics of the solution costs	96
4.11	Cost variations and % of total cost decrease for the B instances	97
4.12	Cost variations and % of total cost decrease for the XB instances	98
4.13	Recovery costs for the MC-APRP instances X01-X04	98
4.14	Recovery costs for the MC-APRP instances X01-X04	98
4.15	Best solutions for B and X instances	99
4.16	Recovery costs for parameter set P3	100
4.17	Recovery costs for parameter set P4	100
4.18	Recovery costs for parameter set P5	101
4.19	Recovery costs for parameter set P6	101
4.20	Recovery costs for parameter set P7	102
4.21	Recovery costs for B01	103
4.22	Recovery costs for B02	103
4.23	Recovery costs for B03	103
4.24	Recovery costs for B04	104
4.25	Recovery costs for B05	104
4.26	Recovery costs for B06	104
4.27	Recovery costs for B07	105
4.28	Recovery costs for B08	105
4.29	Recovery costs for B09	105
4.30	Recovery costs for B10	106
4.31	Recovery costs for XA02	106
4.32	Recovery costs for XA04	106
4.33	Recovery costs for XB01	107
4.34	Recovery costs for XB02	107
4.35	Recovery costs for XB03	107
4.36	Recovery costs for XB04	108
5.1	Characteristics the B01–B05 instances	124
5.2	Characteristics the B06–B10 instances	125
5.3	Characteristics of the XA instances	125
5.4	Characteristics of the XB instances	125
5.5	Characteristics of the X instances	125
5.6	Average solution improvement and computing times	127
5.7	Average solution improvement wrt LNS 10 minutes and LNS 40 minutes . . .	130

5.8	Average solution improvement wrt LNS 10 minutes and LNS 40 minutes . . .	130
5.9	Average solution improvement wrt LNS 10 minutes and LNS 40 minutes . . .	130
5.10	Average solution improvement wrt LNS 10 minutes and LNS average time	131
5.11	Rolling horizon heuristic solution costs	133
5.12	Rolling horizon heuristic solution costs	134
5.13	Improvement wrt the best known solution	135
5.14	Rolling horizon heuristic, column generation heuristic and LNS heuristic solution costs	136
5.15	Very large X01 instances	137
5.16	Very large X03 instances	138
5.17	Rolling horizon heuristic solution costs for X01-P instances	139
5.18	Rolling horizon heuristic solution costs for X0175P instances	139
5.19	Rolling horizon heuristic solution costs for X0150P instances	140
5.20	Rolling horizon heuristic solution costs for X0130P instances	140
5.21	Rolling horizon heuristic solution costs for X03-P instances	141
5.22	Rolling horizon heuristic solution costs for X0375P instances	141
5.23	Rolling horizon heuristic solution costs for X0350P instances	142
5.24	Rolling horizon heuristic solution costs for X0330P instances	142

Remerciements

Je voudrais remercier mes directeurs de recherche, MM. Jean-François Cordeau et Gilbert Laporte, pour leur aide tout au long de ma thèse. Leur expérience et leurs connaissances m'ont gradement aidée. Je les remercie également pour leur patience et leur soutien.

Je voudrais également remercier M. Serge Bisaillon pour son aide précieuse ainsi que M. Walter Rei, membre externe de mon comité, pour les suggestions qui ont permis d'améliorer la qualité de ma thèse.

Pour terminer, je voudrais remercier ma famille et mes amis pour leur soutien et leurs encouragements.

Chapitre 1

Introduction

L'industrie du transport aérien est en croissance. En effet, le nombre de passagers est passé de 310 millions en 1970 à 3,2 milliards en 2014, dont 762 millions de passagers transportés par les compagnies américaines [1], et l'IATA prévoit que le nombre de passagers atteindra 7,3 milliards en 2034 [2]. Toutefois, bien que le nombre de passagers et les revenus des compagnies aériennes soient en croissance, l'industrie du transport aérien est caractérisée par de faibles marges de profit. Entre 1995 et 2014, les revenus des compagnies aériennes américaines ont presque doublé, atteignant 126 milliards en 2014, tandis que la perte nette moyenne entre 2000 et 2014 était de 2,47 milliards [1]. Les principales sources de dépenses des compagnies aériennes sont les coûts de l'essence, de la main-d'oeuvre et de la maintenance, qui représentaient 70% des revenus en 2014. Entre 2000 et 2014, les coûts de l'essence ont presque triplé, atteignant leur maximum en 2012 avec plus de 36% des revenus, tandis que les coûts de la main-d'oeuvre ont significativement diminué depuis 2002, représentant en 2014 moins de 30% des revenus. La figure 1.1 présente les revenus et pertes des compagnies aériennes américaines entre 2000 et 2014 tandis que la figure 1.2 présente le pourcentage des principaux coûts par rapport aux revenus des compagnies américaines de 1995 à 2014.

Les faibles marges de profit de l'industrie ont favorisé l'utilisation de la recherche opérationnelle lors de la planification tactique des opérations. La taille et la complexité des problèmes de planification expliquent également l'importance du rôle que joue la recherche opérationnelle dans la planification tactique des opérations. En effet, de nombreuses compagnies aériennes effectuent un très grand nombre de vols quotidiens. Chaque jour, Air Canada et Jazz, son transporteur régional, exploitent en moyenne 1 300 vols, tandis qu'Air France effectue plus de 1 800 vols. Ainsi, pour s'assurer de bien gérer les ressources nécessaires au

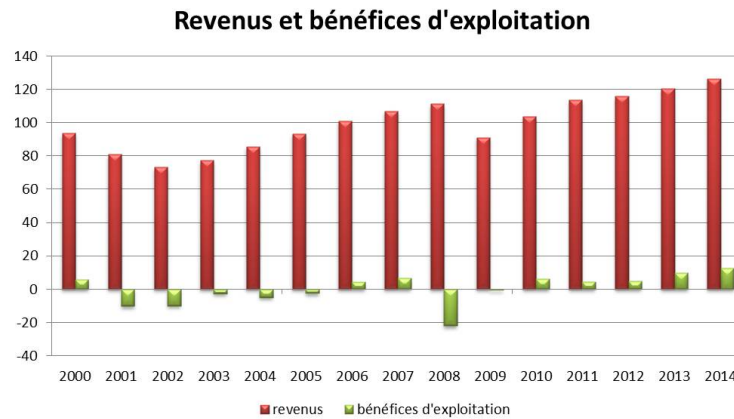


FIGURE 1.1: Revenus et bénéfices des compagnies aériennes américaines de 2000 à 2014

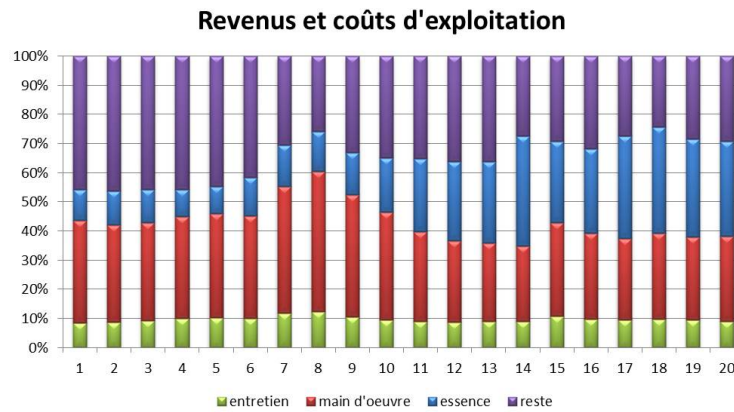


FIGURE 1.2: Rapport des différents coûts sur les revenus des compagnies aériennes américaine de 1995 à 2014

bon fonctionnement de leur réseau, les compagnies aériennes doivent utiliser des systèmes de planification performants. Selon Barnhart [14], l'industrie du transport aérien fournit un environnement propice à l'utilisation des techniques et des modèles de la recherche opérationnelle. De plus, l'utilisation massive des technologies de l'information explique en partie le développement soutenu de la recherche opérationnelle dans cette industrie. Selon Filar [32], la forte compétition résultant de la déréglementation explique l'utilisation intensive de méthodologies de recherche opérationnelle pour la gestion du rendement et les opérations de planification dans l'industrie du transport aérien.

Étant donné la grande taille des problèmes de planification tactique des opérations, les différentes étapes de planification sont généralement effectuées de façon séquentielle. Toutefois,

puisque les différents niveaux sont interreliés, cette approche séquentielle produit des solutions optimales pour chaque sous-problème sans toutefois produire des solutions optimales pour l'ensemble du système. De plus, dans certains cas, la solution d'un sous-problème ne permet pas l'obtention de solutions réalisables aux niveaux subséquents. Depuis quelques années, plusieurs recherches ont été effectuées sur l'intégration des différents niveaux de planification. Une telle approche permet de réduire significativement les coûts d'opération et, par le fait même, d'augmenter la compétitivité des compagnies aériennes.

Par ailleurs, les opérations des compagnies aériennes sont régulièrement perturbées par des événements incontrôlables, tels des conditions climatiques défavorables, un entretien non planifié ou un retard entraînant un manque de personnel ou d'appareils. Ces perturbations engendrent dans le système des retards subséquents ou des annulations de segments de vol qui bouleversent les différents horaires optimisés des compagnies.

De plus, une grande partie des compagnies aériennes opèrent leur service sur des réseaux en étoile *hub-and-spoke*. Les opérations aux terminaux principaux sont généralement caractérisées par un grand nombre d'arrivées suivi d'un grand nombre de départs. Ce type de réseau permet aux compagnies de servir un plus grand nombre de marchés avec un nombre donné de segments de vol et permet de planifier les opérations de façon à assurer une utilisation très efficiente de leurs ressources critiques (appareils et équipages). Toutefois, ceci implique qu'une petite perturbation peut avoir un impact significatif sur l'exécution des opérations planifiées et il existe de nombreuses perturbations possibles. Ces perturbations peuvent être causées par un retard de vol, une annulation de vol, une indisponibilité d'un appareil ou d'un équipage, des conditions météorologiques défavorables, des mesures de protection contre le terrorisme ou une congestion du trafic aérien. Selon Rosenberger et al. [52], les conditions météorologiques sont responsables de 75% des perturbations. Ces dernières peuvent entraîner la fermeture d'un aéroport ou le ralentissement des opérations (diminution du nombre de décollages et d'atterrissages). L'utilisation des réseaux en étoile implique qu'un retard ou une annulation de vol dans le système peut perturber plusieurs autres vols planifiés au courant de la journée et même parfois, durant plus d'un jour. Ball et al. [11] ont estimé le coût total des délais dans le transport aérien aux États-Unis à 32,9 milliards en 2007, dont 8,3 milliards encourus par les compagnies aériennes et 16,7 milliards par les passagers.

Il est donc important pour les compagnies aériennes d'effectuer des opérations de récupération qui permettent de reprendre les horaires planifiés le plus rapidement possible. Tout

comme pour la planification des opérations, la récupération effectuée suite à une perturbation comprend plusieurs étapes : la récupération de la flotte d'appareils, la récupération des équipages et la récupération des passagers. Encore une fois, une approche séquentielle entraîne soit une sous-optimisation du système ou l'obtention de solutions non réalisables.

La littérature sur les différents sous-problèmes de la planification des opérations et de la récupération des opérations est riche. Malgré l'impact que ces opérations de récupération peuvent avoir sur la rentabilité des compagnies aériennes, très peu de recherche porte sur la récupération intégrée des opérations. L'intégration des sous-problèmes permettrait une meilleure optimisation de la récupération. Également, très peu de recherche porte sur la récupération des opérations pour des problèmes réalistes, c'est-à-dire les très larges problèmes auxquels font face les compagnies aériennes.

Chapitre 2

Revue de la littérature

2.1 La planification tactique des opérations

La planification tactique des opérations est divisée en plusieurs étapes. Tout d'abord, les compagnies aériennes doivent concevoir les horaires, c'est-à-dire déterminer quels segments de vol seront effectués. La deuxième étape consiste à affecter les différents types d'appareils de la flotte aux segments de vol choisis à l'étape précédente. Une fois les types d'appareils affectés aux différents segments de vol, il faut créer, pour chaque appareil, une route (itinéraire d'avion) reliant les différents segments de vol, et ce, en s'assurant de respecter les contraintes d'entretien des appareils. Les deux dernières étapes de la planification consistent à créer les rotations d'équipages et à affecter ces rotations au personnel. Il est important de noter que pour les deux dernières étapes, les problèmes sont résolus pour deux groupes différents d'équipages, soit les agents de bord et les équipages techniques (pilotes, officiers et instructeurs).

2.1.1 La création des horaires de vol

La première étape de la planification tactique est la création des horaires de vol et est effectuée plusieurs mois, parfois jusqu'à un an, avant l'implantation des horaires. Le problème est de déterminer quels segments de vol seront exploités par la compagnie et à quels moments. Un segment de vol est défini par un numéro de vol, une origine, une destination et une heure de départ planifiée. Il est primordial pour les compagnies de choisir les segments de vol profitables, car ces derniers ont un impact significatif sur les revenus des compagnies aériennes. Pour s'assurer de choisir les segments profitables, les compagnies doivent recueillir et analyser des informations sur leurs marchés cibles (origine-destination) afin de développer

des estimations précises de la demande. Toutefois, plusieurs facteurs autres que la demande doivent être considérés. Selon Ahmed et al. [8], les vols de connexion, les temps de préparation entre deux vols, la disponibilité des appareils, les préférences des passagers, les horaires des compétiteurs et les disponibilités des créneaux d’atterrissage sont des paramètres qui doivent également être considérés lors de la création des horaires de vol.

De plus, la création des horaires de vol est influencée et reliée à la majorité des décisions des différents problèmes de la planification tactique, telles les décisions d’affectation de la flotte, de rotations d’équipages et d’entretien. Les décisions de toutes les autres compagnies aériennes influencent aussi la rentabilité des horaires de vol choisis. Selon Barnhart et al. [14], ces interdépendances ainsi que le besoin de données précises sur, entre autres les demandes des passagers, les coûts de débordement et le pourcentage de récupération des passagers expliquent l’utilisation limitée de la recherche opérationnelle dans la planification des horaires de vol. Toutefois, selon Barnhart et al. [14], cette tendance semble se renverser et l’optimisation est de plus en plus utilisée à cette étape.

La création des horaires de vol est généralement divisée en deux étapes : la planification de la fréquence et la conception des horaires. La première étape consiste à déterminer pour chaque marché (origine-destination) la fréquence de services appropriée. Ensuite, les différents services proposés doivent être répartis aux différentes heures de la journée en s’assurant de considérer les effets de réseau, c’est-à-dire les transferts entre les différents segments de vol.

Bien que l’impact financier des décisions de création d’horaires soit important, cette étape est généralement effectuée manuellement. Cependant, plusieurs modèles et algorithmes ont été développés pour optimiser localement les horaires de vol. Ainsi, des changements mineurs (ouvrir un nouveau segment de vol ou fermer un segment de vol existant) sont apportés à l’horaire de vol existant. Bien que l’optimisation soit locale, elle a, selon Lohatepanont et al. [42], un impact significatif sur l’utilisation de la flotte ainsi que sur les revenus.

Pour résoudre le problème de création des horaires de vol, Yan et al. [66] ont développé un modèle de multi-flot en nombres entiers dans un réseau qui a pour objectif la suppression des segments de vol non rentables. Le modèle considère plusieurs types d’appareils et est représenté sur un réseau espace-temps. Ce modèle de base est ensuite utilisé pour développer un ensemble de modèles stratégiques en utilisant les modifications suivantes : la suppression de segments de vol multiples, l’ajustement des temps de départ et la location d’appareils.

Pour résoudre le modèle, les auteurs utilisent un algorithme basé sur une relaxation lagrangienne et une méthode de sous-gradient. L'algorithme a été testé sur 25 instances créées à l'aide de données d'une compagnie aérienne taïwanaise. Les auteurs concluent que lorsque le nombre de contraintes est petit, l'algorithme développé converge rapidement.

Lohatepanont et al. [42] ont développé deux modèles (*Integrated Schedule Design and Fleet Assignment Model* et *Approximate Schedule Design and Fleet Assignment Model*) qui optimisent simultanément la sélection des segments de vol et l'affectation des types d'appareils. Les modèles utilisent comme données une liste maîtresse de vols indiquant les vols obligatoires et les vols optionnels ainsi que la demande moyenne des itinéraires. Les modèles considèrent l'effet de récupération des passagers et l'interdépendance des segments de vol à l'aide de l'approche développée par Barnhart et al. [16]. Pour résoudre les modèles, les auteurs utilisent une relaxation linéaire du problème maître restreint et une méthode de branch-and-bound. Les deux modèles développés permettent de diminuer substantiellement les coûts d'opération. Par ailleurs, les solutions obtenues avec le modèle approximatif sont de meilleure qualité, puisque la complexité réduite de ce modèle permet de générer un plus grand nombre de solutions entières.

Le problème de création des horaires de vol considérant l'aspect stochastique de la demande a été étudié par Yan et al. [65]. Le modèle ne considère qu'un seul type d'appareil et que des vols directs. Les auteurs ont développé un modèle de programmation stochastique à deux étapes. Les variables de décision de la première étape sont les variables de flot de la flotte tandis que celles de la deuxième étape sont les variables de flot des passagers. Un modèle déterministe utilisant une demande moyenne est développé pour évaluer la borne inférieure du modèle stochastique. Pour résoudre le modèle, les auteurs ont développé deux méthodes heuristiques, une basée sur les arcs et l'autre basée sur les chemins. Des données provenant d'une compagnie aérienne taïwanaise ont été utilisées pour tester les méthodes de résolution. Pour les cinq cas étudiés, les solutions obtenues à l'aide des heuristiques sont très près des solutions optimales.

2.1.2 L'affectation des différents types d'appareils aux segments de vol

Les compagnies aériennes opèrent une flotte constituée de différents types d'appareils. Le problème d'affectation des types d'appareils aux segments de vol consiste à déterminer

quel type d'appareil effectuera les différents segments de vols. Chaque appareil est défini par un numéro de série, un modèle (type d'appareil) et une configuration de sièges. Après avoir déterminé les segments de vol que la compagnie opérera, celle-ci doit affecter les différents types d'appareils de la flotte à tous ces segments. Deux objectifs peuvent être utilisés pour résoudre ce problème, soit la maximisation de la contribution de la flotte ou la minimisation des coûts d'affectation. La maximisation de la contribution de la flotte est le revenu maximal d'un segment de vol, sans considérer la capacité, moins les coûts d'affectation. Les coûts d'affectation sont les coûts d'opération du type d'appareil, les coûts variables liés au nombre de passagers et les coûts de débordement. Les coûts de débordement sont le résultat de l'aspect périssable du transport aérien. En effet, lorsque les avions quittent le sol, les sièges non utilisés représentent une perte de revenu pour les compagnies. Puisque le rapport coûts fixes/coûts variables est très grand et afin de réduire les coûts d'opération, les compagnies doivent s'assurer de remplir au maximum les appareils. Toutefois, la demande non satisfaite, c'est-à-dire le nombre de passagers qui ne peuvent être desservis par un segment de vol, représente également une perte de revenus pour les compagnies. Les coûts de la demande non satisfaite et les coûts de débordement dépendent du pourcentage de récupération de passagers de la compagnie, c'est-à-dire, le pourcentage de passagers qui voyageront sur un autre segment de vol de la compagnie. Les compagnies doivent donc, pour chaque segment de vol, choisir le type d'appareil dont la capacité pourra répondre le plus précisément possible à la demande.

L'utilisation de réseaux en étoile implique que de nombreux passagers devront effectuer au moins une escale durant leur itinéraire. Ces escales augmentent les déplacements nécessaires ainsi que les manipulations de bagages. Une séquence combinée est formée lorsque le même appareil et le même numéro de vol sont utilisés pour l'ensemble des segments de vol d'un itinéraire. Bien qu'il soit possible de créer une séquence combinée pour plus de deux segments de vol, les itinéraires comportant trois segments de vol ou plus sont relativement rares et ne sont généralement pas considérés lors de l'élaboration des séquences combinées. Bien entendu, les passagers préfèrent ce type d'itinéraire car il leur permet de rester à bord de l'avion entre deux séquences de vol. Par conséquent, les compagnies aériennes peuvent demander un prix plus élevé pour ce type d'itinéraire. La formation des séquences combinées est généralement effectuée lors de l'affectation des types d'appareils et les compagnies aériennes doivent, à cette étape, choisir celles qui sont les plus rentables, c'est-à-dire celles qui permettent d'augmenter le plus les profits.

Tous les problèmes d'affectation de flotte d'appareils doivent respecter trois groupes de contraintes : les contraintes de balancement, les contraintes de capacité et les contraintes de couverture. Les contraintes de balancement assurent qu'à chaque nœud (aéroport), le nombre d'appareils entrants équivaut au nombre d'appareils sortants. Les contraintes de capacité limitent, pour chaque type, le nombre d'appareils utilisés à la quantité disponible et les contraintes de couverture garantissent que chaque segment de vol est couvert par exactement un type d'appareil.

En plus de ces trois groupes, plusieurs autres contraintes liées aux aéroports et aux caractéristiques des différents appareils doivent être respectées pour représenter adéquatement la réalité spécifique des différentes compagnies. Les compagnies doivent considérer les différentes caractéristiques des appareils lors de l'affectation aux segments de vol, telles la vitesse des appareils, la consommation d'essence, la distance maximale franchissable, la présence d'équipements spécifiques nécessaires, le niveau de bruit, les coûts d'entretien et les temps de préparation. Il faut également considérer les contraintes qui limitent le nombre d'appareils pouvant passer la nuit à un aéroport spécifique, les restrictions sur les heures d'atterrissage ou de décollage d'appareils bruyants et les restrictions sur la taille des appareils pouvant atterrir à certains aéroports (dépendant de la longueur des pistes d'atterrissage). De plus, tout comme pour le problème de création d'horaires de vol, le problème d'affectation de la flotte est interrelié aux autres problèmes de planification tactique de la compagnie.

Afin de limiter la complexité des modèles développés pour résoudre le problème d'affectation de la flotte, plusieurs suppositions sont généralement faites. Tout d'abord, les modèles considèrent généralement que la demande est statique. Or, l'industrie du transport aérien est caractérisée par une demande variant en fonction des jours de la semaine. Toutefois, prendre en considération cette variation entraîne une augmentation de la complexité du problème. Par conséquent, les modèles considèrent généralement la même affectation tous les jours de la semaine, c'est-à-dire que l'horaire est répété tous les jours et que le même type d'appareil est affecté au même segment de vol tous les jours de la semaine.

De plus, les modèles considèrent généralement la demande de transport comme étant déterministe et utilisent une demande agrégée et des prix moyens pour toutes les classes (Barnhart et al. [16]). Toutefois, afin d'estimer correctement les coûts de débordement, il est nécessaire de considérer les demandes et les prix en fonction des différentes classes de passagers. De plus, les coûts de débordement dépendent également de l'interdépendance entre les différents seg-

ments de vol, c'est-à-dire qu'une demande non satisfaite sur un marché (origine-destination) peut modifier la demande sur un autre marché relié (dont l'origine est la destination du précédent marché). Toutefois, les coûts de débordement sont habituellement considérés comme étant spécifiques aux segments de vol, plutôt que spécifiques aux itinéraires des passagers. Pour ces raisons, ils ne sont calculés qu'approximativement dans la majorité des modèles. Pour terminer, les temps de vol et les temps au sol sont généralement considérés comme étant déterministes, bien qu'en réalité, la congestion, les conditions météorologiques et les procédures de sécurité engendrent de grandes variations dans les temps de vol et les temps au sol.

Pour résoudre le problème d'affectation de la flotte, Abara [3] a développé un modèle de programmation linéaire en nombres entiers. Le modèle est représenté sur un réseau de connexions où les nœuds représentent le temps d'arrivée ou de départ des segments de vol et les arcs représentent les connexions entre les arrivées et les départs (plus les arcs de début et de fin). Le modèle développé permet de traiter les problèmes où tous les segments de vol sont obligatoires ainsi que les problèmes où certains segments de vol peuvent être éliminés. L'objectif est de maximiser les profits en respectant les contraintes de balancement, de capacité et de couverture. Toutefois, les contraintes de balancement et de capacité sont exprimées à l'aide de pénalités dans la fonction objectif. Le modèle a été utilisé chez American Airlines pour évaluer plusieurs décisions, telles l'emplacement des bases d'équipages et la maximisation de l'utilisation de certains types d'appareils. Le modèle a permis une augmentation des revenus de 1% ainsi qu'une réduction des coûts d'opération de 0,4%.

Hane et al. [35] ont développé un modèle pour résoudre le problème d'affectation de la flotte et sont parvenus à obtenir une solution en moins d'une heure pour un problème comportant 150 villes, 2 500 vols et une flotte de onze appareils. Le modèle est résolu à l'aide d'une relaxation linéaire améliorée et d'un branch-and-bound basé sur un ordre de priorité pour les branchements. Tout d'abord, la taille du réseau est réduite en regroupant les arrivées consécutives et les départs consécutifs. Ensuite, le nombre d'arcs est réduit en imposant une structure d'île à certaines villes et les connexions manquées sont éliminées. Les auteurs comparent les solutions obtenues à l'aide de la méthode des points intérieurs et de la méthode du *dual steepest edge simplex* du solveur OSL. La première méthode permet d'obtenir de meilleurs résultats pour les problèmes de flotte comportant sept appareils, tandis que la deuxième offre de meilleurs résultats pour les problèmes de flotte comportant onze appareils. Avant de débiter la phase du branch-and-bound, la solution fractionnaire optimale

est utilisée pour fixer les variables à la borne la plus proche. Pour le branch-and-bound, le branchement choisi est celui qui possède l'ordre de priorité le plus élevé et qui minimise la dégradation maximale. Cette technique permet de réduire grandement le nombre de noeuds explorés dans le branch-and-bound ainsi que le temps de traitement. De plus, le modèle de base peut être facilement adapté pour intégrer plusieurs autres contraintes, telles les limites sur le nombre d'appareils de grande taille permis aux aéroports et les limites sur le bruit aux aéroports.

Rushmeier et al. [53] ont modélisé le problème d'affectation de la flotte dans un réseau de connexions où chaque noeud représente un événement (un vol effectué par un type d'appareil) et chaque arc représente une activité (activité de vol ou d'attente). Par conséquent, chaque arc de vol représente une connexion entre deux événements pouvant être exécutés consécutivement par le même appareil. Pour limiter le nombre de connexions possibles, les événements sont divisés en sous-ensembles connectés de vols entrants et sortants. L'utilisation des sous-ensembles permet de réduire significativement le temps de traitement sans toutefois réduire la qualité des solutions. Afin de respecter les contraintes de ressources opérationnelles, des termes de pénalité sont introduits dans la fonction objectif. Les contraintes de ressources peuvent être le nombre d'appareils de chaque type utilisés, le nombre d'heures de vol de chaque équipage, le nombre maximum d'équipages arrivant ou quittant une station, etc. Pour résoudre le problème, les auteurs ont développé un modèle de programmation mixte avec une fonction objectif linéaire par morceaux. Sept horaires provenant de USAir ont été utilisés pour tester le modèle. Tout d'abord, l'algorithme du simplexe dual de CPLEX est utilisé pour résoudre la relaxation linéaire du problème. Par la suite, toutes les variables dont la valeur est supérieure à 0,99 sont fixées à un et les autres à zéro et un processus de branch-and-bound est utilisé. Les auteurs rapportent qu'après deux heures de traitement, les écarts entre la valeur de la solution de la relaxation linéaire et celle du modèle de programmation mixte varient entre 0,3 et 6%.

Pour résoudre le problème d'affectation de la flotte, Barnhart et al. [16] ont développé un modèle qui prend en considération les effets de réseau et qui permet d'estimer plus précisément le débordement et la récupération des passagers. Les auteurs démontrent qu'il existe une interdépendance importante entre les différents segments de vol influençant le débordement et la récupération des passagers, interdépendance qui n'est pas prise en compte dans les modèles de base d'affectation de la flotte. Pour résoudre le problème, les auteurs utilisent une relaxation linéaire suivie d'un branch-and-bound. De plus, pour diminuer le temps de

traitement, une réduction de coefficients est effectuée sur certaines contraintes. Les auteurs utilisent des données de réseaux complets provenant d'une grande compagnie aérienne américaine pour tester leur modèle et les résultats sont comparés aux résultats obtenus avec le modèle de base d'affectation de la flotte. Les auteurs démontrent que l'intégration des effets de réseaux et de l'effet de récupération des passagers a un impact significatif sur les coûts d'affectation ainsi que sur les coûts d'opération. De plus, le modèle permet une amélioration du facteur de charge du système, c'est-à-dire une meilleure correspondance entre la charge de passagers et la capacité des appareils.

Une variante du problème est le problème d'affectation de la flotte avec fenêtres de temps. Dans ces problèmes, l'horaire utilisé comme donnée n'est pas définitif et peut être modifié. Ainsi, l'utilisation des fenêtres de temps dans le problème d'affectation permet d'apporter des changements raisonnables à l'horaire déterminé à l'étape précédente.

Le modèle de programmation en nombres entiers développé par Rexing et al. [51] pour résoudre le problème d'affectation de la flotte avec fenêtres de temps est représenté sur un réseau espace-temps. Chaque fenêtre de temps est discrétisée en un ensemble de temps de départs possibles qui sont représentés par des copies de l'arc de vol initial et seulement une copie peut être choisie. Des changements dans l'horaire des vols peuvent modifier les demandes des passagers, surtout lorsque l'on considère deux vols consécutifs desservant un même marché. Toutefois, afin de s'assurer de ne pas modifier les demandes des passagers, Rexing et al. [51] utilisent de petites fenêtres de temps. Le modèle développé est une variante du modèle de base et par conséquent, la qualité des solutions obtenues est nécessairement supérieure ou égale aux solutions du modèle sans fenêtres de temps. Trois méthodes sont utilisées pour réduire la taille du réseau, soit l'agrégation de nœuds, l'élimination d'arcs superflus et la création de structures d'île. Pour résoudre le problème, les auteurs utilisent deux approches, soit un algorithme direct utilisant le solveur CPLEX et un algorithme itératif permettant de réduire grandement l'espace mémoire nécessaire. Initialement, l'algorithme itératif ne considère qu'un seul arc pour chaque combinaison vol-appareil. Puis à chaque itération, de nouveaux arcs sont ajoutés lorsque ceci permet d'améliorer la solution. Les algorithmes ont été testés sur des instances provenant d'une grande compagnie aérienne américaine avec différentes grandeurs de fenêtres de temps. Les mêmes solutions ont été obtenues à l'aide des deux algorithmes et l'algorithme direct est plus rapide. Les auteurs rapportent que l'utilisation de fenêtres de temps de plus ou moins 10 minutes permet d'économiser plus de 65 000\$ quotidiennement. Bien que l'algorithme itératif soit moins rapide, puisqu'il nécessite moins

d'espace mémoire, les auteurs concluent qu'il serait intéressant de l'utiliser dans la résolution de problèmes d'affectation intégrés avec les problèmes de création d'itinéraires d'appareils ou de création des rotations d'équipages.

Bélanger et al. [20] considèrent également le problème périodique d'affectation de la flotte avec fenêtres de temps. Les fenêtres de temps permettent une flexibilité sur les temps de départ des appareils et le calcul des revenus varie en fonction du temps. Afin d'éviter de modifier les demandes des passagers, les auteurs imposent une pénalité lorsque les heures de départ de deux vols consécutifs desservant un même marché sont trop rapprochées. De plus, pour s'assurer de minimiser le nombre d'avions, le coût fixe lié à l'utilisation d'un appareil est très grand. Les auteurs ont développé un modèle non linéaire de multi-flots dans un réseau, représenté sur un réseau espace-temps. Un algorithme de branch-and-price est utilisé pour résoudre le problème. Pour tester l'algorithme, des instances basées sur des données provenant d'une compagnie aérienne nord-américaine ont été créées. Pour l'ensemble des instances, l'utilisation des fenêtres de temps permet une grande réduction du nombre d'appareils nécessaires et, dans la grande majorité des cas, une augmentation des profits.

Une autre variante du problème est l'affectation dynamique de la flotte. L'affectation dynamique permet d'ajuster la capacité offerte sur chaque segment de vol à la demande des passagers. Puisqu'il est possible pour les équipages d'être affectés à différents appareils d'une même famille, il est possible d'interchanger l'affectation des appareils dans cette famille. Pour résoudre le problème, Berge et al. [18] ont développé un modèle de multi-flots dans un réseau, représenté sur un réseau espace temps. L'objectif est de maximiser les profits en respectant les contraintes de capacité, de conservation de flots et de couverture. Les auteurs ont développé deux méthodes heuristiques pour résoudre le problème. La première est une méthode séquentielle de flot à coût minimum qui utilise un algorithme de plus court chemin à chaque itération pour deux types d'appareils. La seconde méthode débute avec une solution réalisable et à chaque itération échange deux affectations si ceci améliore la solution. L'approche a été testée sur vingt instances basées sur des réseaux réels de compagnies aériennes et a permis une amélioration des profits variant entre 1 et 5%.

2.1.3 La création des itinéraires d'appareils

À l'étape précédente, chaque segment de vol a été affecté à un type d'appareil. Toutefois, ceci ne permet pas de savoir quel appareil parmi le type choisi sera utilisé pour chaque seg-

ment de vol. La création des itinéraires d'appareils *maintenance routing* permet de créer les routes pour tous les appareils de chaque type, et ce, en respectant les contraintes d'entretien. Les compagnies aériennes sont soumises aux règles d'entretien gouvernementales ainsi qu'aux règles internes, qui varient d'une compagnie à l'autre. Les agences gouvernementales imposent différents types d'entretien qui varient en fonction de leur envergure, de leur fréquence et de leur durée. Les différents types d'entretien des appareils sont généralement les suivants (Sriram et al., [59]) :

- Entretien de type A, ou entretien de routine : Ces entretiens doivent être faits après 65 heures de vol et comprennent une inspection visuelle des systèmes majeurs des appareils, tels les moteurs et les freins d'atterrissage. Ces entretiens durent environ six heures.
- Entretien de type B : Ces entretiens sont effectués après 300 à 600 heures de vol et incluent une inspection visuelle plus approfondie ainsi que la lubrification de toutes les parties mobiles de l'appareil. Ces entretiens durent environ 12 heures.
- Entretien de type C et D : Ces entretiens sont effectués à un intervalle variant de un an à quatre ans. Pour ce type d'entretien, l'appareil doit être retiré de la flotte disponible pour environ un mois.

Bien que les compagnies aériennes doivent respecter les règles gouvernementales, elles possèdent généralement des règles internes plus strictes et les entretiens de type A sont normalement effectués tous les trois à quatre jours, tandis que les entretiens de type B sont réalisés tous les trois mois. Puisque la planification tactique est généralement faite pour une période de trois mois, seul les entretiens de type A et B sont pris en considération dans le problème de création des itinéraires d'appareils.

Ces deux types d'entretien doivent être effectués par un personnel spécialisé, et par conséquent, ils doivent être réalisés à des stations d'entretien spécifiques. Ces stations peuvent généralement accepter plusieurs types d'appareils pour les entretiens de type A. Afin de maximiser l'utilisation des appareils, pour les vols domestiques, les compagnies réalisent les entretiens de type A la nuit, car les vols domestiques sont effectués majoritairement durant le jour. Ainsi, les entretiens de nuit ne nécessitent pas le retrait des appareils de la flotte disponible.

Toutefois, la considération des entretiens de type B rend le problème plus complexe. Les

entretiens de type B de chaque type d'appareil ne peuvent être effectués qu'à une seule station d'entretien. Par conséquent, il n'est pas possible d'effectuer un entretien sur plusieurs appareils d'un certain type au même moment. De plus, puisque chaque type d'appareil nécessite des pièces d'équipement différentes, chaque station ne peut effectuer l'entretien que sur un type d'appareil.

Le problème de création d'itinéraires d'appareils a été étudié par Feo et al. [31]. Les auteurs ont développé un modèle de multi-flots dans un réseau avec contraintes d'intégralité. L'objectif du modèle est de minimiser les coûts d'entretien en respectant les contraintes de conservation de flots, de couverture, de capacité des installations d'entretien et d'intervalle de temps entre les entretiens. Afin de réduire la taille du problème, ce dernier est décomposé en problèmes individuels pour chaque type d'appareil. De plus, les contraintes de capacité des installations sont relaxées. Les auteurs ont développé une heuristique à deux étapes pour résoudre le problème. Tout d'abord, un horaire est créé pour chaque appareil sans toutefois assurer le respect des contraintes d'entretien. Ensuite, les bases d'entretien minimisant les coûts sont choisies. À chaque itération, les données sont perturbées pour assurer l'exploration de différents voisinages. À la deuxième étape, une heuristique gloutonne est utilisée pour effectuer des échanges entre des paires d'horaire. Pour tester l'algorithme, 25 instances ont été générées aléatoirement pour un seul type d'appareil. L'algorithme permet de réduire le nombre d'installations nécessaires de 22 à 17.

Gopalan et al. [34] ont développé deux modèles pour résoudre le problème de création d'itinéraires considérant deux types d'entretiens, un effectué au maximum après trois jours sans entretien et l'autre effectué à tous les N jours (N équivalant au nombre d'appareils d'un type dans la flotte). Tout d'abord, les segments de vol sont regroupés en lignes de vol qui spécifient l'origine et la destination des appareils à chaque jour. Le premier modèle, le modèle à horizon infini, considère que les lignes de vol se répètent à tous les jours (lignes de vol identiques). Pour résoudre ce modèle, les lignes de vol sont initialement fixées et un algorithme est utilisé pour déterminer si le graphe résultant permet d'obtenir un cycle eulérien respectant les conditions d'entretien. Dans le second modèle, le modèle à horizon fini, les lignes de vol peuvent varier de jour en jour et l'horizon de planification est de N jours. Un algorithme de décomposition de chemins en deux étapes permet de déterminer les itinéraires respectant les contraintes d'entretien. Un exemple numérique est présenté, mais la performance des algorithmes n'est pas démontrée.

Sriram et al. [59] ont développé deux modèles pour résoudre le problème de création des itinéraires. Les auteurs considèrent également la réaffectation de la flotte puisque celle-ci peut entraîner des pertes de revenus ou des augmentations de coûts d'opération. Le problème est défini sur un horizon cyclique de sept jours et considère une flotte d'appareils hétérogène. Tout comme pour Gopalan et al. [34], seules l'origine au début de la journée et la destination à la fin de la journée sont considérées et les activités d'entretien sont effectuées la nuit. Deux types d'entretiens sont également considérés. Les modèles sont représentés sur un réseau espace-temps et l'objectif est de minimiser les coûts d'entretien ainsi que les pénalités imposées pour les réaffectations non appropriées de la flotte d'appareils. Le premier modèle est basé sur le nombre de jours séparant deux entretiens, tandis que le second modèle est basé sur le nombre d'heures de vol. Toutefois, la deuxième façon de modéliser augmente largement la taille du problème. Les auteurs ont développé une heuristique pour résoudre le problème. Tout d'abord, une liste d'appareils et de noeuds est créée, puis par ordre, le meilleur itinéraire cyclique est trouvé pour chaque noeud et les arcs associés sont retirés du réseau. Ensuite les listes sont modifiées aléatoirement et les étapes sont répétées. L'heuristique a été testée sur 44 instances générées aléatoirement. Pour les 26 instances de plus petite taille, le modèle est également résolu avec le solveur CPLEX et l'heuristique permet d'obtenir des solutions à moins de 4,6% des solutions optimales. De plus, le temps de traitement nécessaire pour résoudre chaque instance à l'aide de l'heuristique est inférieur à 100 secondes tandis qu'il atteint 16 000 secondes lorsque la plus grande instance est résolue avec CPLEX.

2.1.4 La création des rotations d'équipages

En raison de l'importance des coûts d'équipages qui représentent les coûts d'opération les plus élevés après les coûts de carburant, les problèmes de création des rotations d'équipages et d'affectation des rotations au personnel ont été largement étudiés. L'objectif du problème est de générer un ensemble de rotations de coût minimum afin de couvrir tous les segments de vol planifiés. Le problème de rotations d'équipages est très difficile à résoudre car il existe un très grand nombre de combinaisons possibles d'affectation et les contraintes (gouvernementales et de contrats de travail) devant être respectées peuvent être très complexes.

Une rotation d'équipage comprend un ensemble de séquences de vol réalisables qui débutent et se terminent à une base de la compagnie, généralement un terminal principal. Une rotation est formée de journées de travail et de périodes de repos. Les rotations de vols domestiques

sont généralement de courte durée (entre un et quatre jours) tandis que les rotations de vols internationaux peuvent durer plus d'une semaine. Les journées de travail peuvent être formées d'une séquence de vol ou d'un ensemble de séquences de vol, entrecoupées par des périodes courtes de repos (périodes de connexion) permettant le transfert entre deux vols. Lorsqu'un équipage travaille sur deux séquences de vol qui n'utilisent pas le même appareil, le temps de connexion nécessaire est plus grand. De plus, parfois deux séquences de vol consécutives peuvent nécessiter un repositionnement. Le repositionnement peut être effectué par transport terrestre lorsque les aéroports de connexions sont rapprochées ou par transport aérien. Ces vols de repositionnement peuvent également être effectués pour retourner les membres de l'équipage à leur base.

Un grand nombre de contraintes provenant des agences gouvernementales et des contrats de travail doivent être prises en considération lors de la création des rotations. Bien entendu, les règles gouvernementales varient d'un pays à l'autre mais, de façon générale, les règles gouvernementales limitent (AhmadBeygi et al. [7]) :

- le nombre d'heures maximal d'une journée de travail ;
- le temps minimal et maximal entre deux séquences de vol ;
- le temps minimal de repos entre deux journées de travail consécutives ;
- le nombre maximal de journées de travail dans une rotation.

Par exemple, les règles de la FAA stipulent qu'une journée de travail ne peut excéder 12 heures et qu'une période de repos de huit heures doit être allouée après une journée de travail de 12 heures. De plus, durant les sept jours précédents une rotation, le nombre maximal d'heures travaillées ne peut dépasser 60 heures. Finalement, une rotation ne peut durer plus de 14 jours et après cinq journées de travail consécutives, une période de repos de 48 heures est obligatoire.

Les règles de Transport Canada sont les suivantes :

- Le temps total de service de vol ne dépasse pas 14 heures par période de 24 heures consécutives.
- Des périodes de repos qui permettent au membre d'équipage de conduite (technique) de dormir au moins neuf heures par période de 24 heures consécutives dans des installations adéquates.

- Une de ces périodes de repos permet au moins cinq heures consécutives de sommeil entre 20 h et 6 h, heure locale.
- Le membre d'équipage de conduite reçoit au moins cinq périodes de repos par période de 24 heures consécutives pour chaque période de 30 jours consécutifs.

De plus, il est permis d'augmenter le nombre d'heures dans une journée de travail en raison de circonstances imprévues si certaines conditions sont satisfaites (ex : une augmentation du temps de repos équivalent à la période supplémentaire de travail).

Les contraintes provenant des contrats de travail sont plus nombreuses et peuvent être plus complexes. Par exemple, le nombre d'heures de repos suite à une journée de travail peut dépendre du nombre d'heures de repos de la période précédente ou suivante. Il existe également des contraintes sur le nombre d'heures de vol et le nombre de segments de vol permis durant une journée de travail. En plus de ces contraintes, il faut considérer entre autres, les périodes de vacances, les périodes d'entraînement, la composition de l'équipage et les qualifications de l'équipage.

De plus, la structure de coûts (salaires) est non linéaire, c'est-à-dire que le coût d'une rotation est le maximum entre a) la somme des coûts des journées de travail et b) un pourcentage fixe de la durée totale de la rotation. Le coût d'une journée de travail est également le maximum entre a) le temps total de vol, b) un pourcentage fixe de la durée totale de la journée et c) un minimum garanti (AhmadBeygi et al. [7]). Par conséquent, les coûts d'une journée de travail ne dépendent pas uniquement du temps de vol, mais également du temps et du coût (indemnité quotidienne) des repos et du coût de repositionnement. Le coût de repositionnement équivaut au revenu perdu si un membre de l'équipage utilise un vol de la compagnie ou au coût du billet d'avion s'il utilise un vol d'une autre compagnie. L'objectif des compagnies est de minimiser les coûts de crédit, c'est-à-dire la différence entre les coûts réels payés et les coûts reliés au temps de vol. Une grande différence entre ces coûts peut être causée par des périodes de repos longues ou fréquentes pendant les journées de travail, de longues périodes de repos entre les journées de travail ou des repositionnements (Anbil et al. [9])

Pour effectuer un segment de vol, il est nécessaire d'affecter un certain nombre d'employés en fonction du type d'appareil considéré. Il existe deux types d'équipages, soit l'équipage technique et les agents de bord. Les membres de l'équipage technique sont les pilotes, les officiers

et les instructeurs. Ces derniers peuvent remplacer les pilotes et les officiers et les pilotes peuvent remplacer les officiers. Les agents de bord peuvent travailler sur plusieurs familles d'appareils, tandis que l'équipage technique est qualifié pour piloter une famille d'appareils et peut piloter l'ensemble des types d'appareils appartenant à cette famille, par exemple les modèles A318, A319, A320 de la famille AirbusSmall. Le problème de rotations des équipages techniques peut donc être considéré séparément pour chaque famille d'appareils sans diminuer la qualité des solutions. Par conséquent, le problème de rotations des équipages doit être résolu séparément pour les deux types d'équipages.

En plus des contraintes légales mentionnées précédemment, d'autres groupes de contraintes doivent être respectées. Les contraintes d'équilibrage des équipages assurent que les rotations respectent le nombre de membres d'équipages disponibles à chaque base. Ces contraintes sont généralement exprimées en fonction d'un nombre minimal et maximal d'heures de vol disponibles. D'autres contraintes peuvent également être ajoutées. Les contraintes de repositionnement limitent le nombre total de repositionnements permis durant une rotation, tandis que les contraintes de changement d'avion limitent le nombre de changements possibles durant une journée de travail. Toutefois, ces deux dernières contraintes peuvent apparaître dans la fonction objectif plutôt que dans les contraintes. Ainsi, le nombre total de changements ou de repositionnements n'est pas limité, mais des pénalités sont ajoutées au coût de la fonction objectif et le modèle tentera de diminuer le plus possible ces pénalités.

Pour résoudre le problème de création des rotations d'équipages, AhmadBeygi et al. [7] ont développé un modèle de programmation mixte incluant des variables binaires de connexion et de marqueur. Les premières variables déterminent si deux vols sont consécutifs et les deuxièmes indiquent les vols qui débutent et terminent les rotations. L'objectif du modèle est de minimiser le coût réduit des rotations qui est représenté par le coût réel de la rotation moins la somme de la valeur duale de tous les segments de vol de la rotation. Le coût non linéaire des rotations est calculé dans les contraintes de coûts. Des contraintes assurant la légalité des vols et des rotations doivent également être respectées en plus des contraintes de balancement. Les auteurs ont développé le modèle pour résoudre le problème quotidien de rotation d'équipage, c'est-à-dire le problème qui considère que les vols sont répétés à tous les jours. Toutefois, le modèle peut être facilement modifié pour résoudre le problème hebdomadaire et le problème de transition. Deux instances, la première comprenant 143 vols et quatre bases d'équipages et la seconde comprenant 329 vols et quatre bases d'équipages, ont été utilisées pour tester l'efficacité de la méthode de résolution. CPLEX a été utilisé pour ré-

soudre la relaxation linéaire du problème maître. La première instance a été résolue de deux façons, soit en générant les colonnes optimales (celle dont le coût réduit est le plus négatif) ainsi qu'en générant des colonnes réalisables (première colonne de coût réduit négatif). Bien que la valeur de la fonction objectif soit la même pour les deux méthodes, la deuxième permet de réduire grandement le temps de traitement nécessaire et cette méthode a été utilisée pour résoudre la seconde instance. Les auteurs ont démontré que le temps nécessaire pour obtenir la solution optimale est grand, mais que des solutions de bonne qualité (à 1% de l'optimalité) sont obtenues rapidement.

Pour résoudre le problème de création des rotations d'équipages, Desaulniers et al. [28] ont développé un modèle de multi-flots en nombres entiers dans un réseau avec contraintes de ressources. L'objectif du modèle est de minimiser le coût des flots, des ressources et des variables supplémentaires qui sont utilisées pour relaxer les contraintes de couverture des segments de vols et les contraintes globales. Le modèle est résolu à l'aide d'un algorithme de branch-and-bound basé sur une extension de la décomposition de Dantzig-Wolfe. Cette décomposition permet d'obtenir un problème maître de partitionnement d'ensemble et des sous-problèmes du plus court chemin. Ceci permet également d'isoler tous les aspects non-linéaires dans les sous-problèmes qui sont résolus à l'aide d'un algorithme de programmation dynamique. Le modèle a été testé sur 11 instances provenant d'Air France et comprenant entre 154 et 1157 segments de vol. Pour l'ensemble des instances, l'écart entre la valeur de la solution obtenue avec l'algorithme et la valeur de la solution de la relaxation linéaire est inférieure à 0,5% et, à l'exception d'une instance, les solutions ont été obtenues en moins d'une heure.

Barnhart et al. [17] ont développé un modèle approximatif pour résoudre le problème de création des rotations d'équipages pour les vols de longue durée. Le modèle résultant est un modèle de circulation de flots dans un réseau dont les contraintes de temps hors base et les contraintes de localisation de début et de fin des rotations sont relaxées. Ceci permet de réduire grandement le nombre de variables. Le modèle approximatif ainsi qu'une procédure de sélection de vols de repositionnement développée par Barnhart et al. [15] ont été testés sur cinq instances provenant d'une compagnie aérienne américaine. L'utilisation de la procédure avec le modèle approximatif permet de réduire d'un ordre de grandeur le temps nécessaire pour générer les vols de repositionnement. La solution obtenue est ensuite utilisée comme solution de départ pour la résolution du problème à l'aide d'une méthode de branch-and-bound. Les auteurs démontrent que l'utilisation de la procédure et du modèle approximatif

permet d'améliorer grandement les solutions, permettant des économies annuelles variant de 1,5 à 10 millions \$.

2.1.5 L'affectation des rotations aux membres d'équipages

La dernière étape de la planification tactique des opérations est la création des horaires mensuels des équipages qui sont formés de plusieurs rotations créées à l'étape précédente. L'objectif du problème est de générer un ensemble d'horaires mensuels qui minimisent les coûts et qui répondent aux normes de qualité de vie des employés (établies par la compagnie). La qualité d'un horaire dépend entre autres de la difficulté des séquences de vol, du nombre de vols de nuit et du nombre de fuseaux horaires traversés. Il existe plusieurs contraintes devant être respectées, dont :

- des contraintes limitant le nombre maximum d'heures de travail durant la période couverte par l'horaire ;
- des contraintes imposant un nombre minimum d'heures de repos entre deux rotations ;
- des contraintes s'assurant que les employés affectés aux rotations possèdent les qualités nécessaires. Les qualités nécessaires peuvent être les qualifications techniques, l'emplacement des bases, les langues parlées ou les visas requis lors de vols internationaux.

Les horaires mensuels peuvent être créés selon trois méthodes différentes : la méthode des enchères *bidline*, la méthode de *rostering* et la méthode des enchères préférentielles *preferential bidding*. Pour la méthode des enchères, les horaires sont d'abord construits et les employés choisissent leur horaire par ordre d'ancienneté. Cette méthode est utilisée dans la majorité des compagnies aériennes d'Amérique du Nord. La méthode *rostering* consiste à créer des horaires personnalisés pour les employés sans toutefois les consulter. Cette méthode tente de répartir équitablement la charge de travail entre les employés et de considérer les horaires des mois précédents (l'horaire mensuel d'un employé contenant beaucoup de vols difficiles sera suivi d'un horaire mensuel contenant beaucoup de vols faciles). Pour terminer, la méthode des enchères préférentielles est un mélange des deux méthodes précédentes. Des horaires personnalisés sont créés en fonction des préférences des employés. L'ancienneté des employés est considérée puisqu'un employé plus ancien se verra octroyer un plus grand nombre de préférences qu'un nouvel employé. Les préférences peuvent être la volonté d'obtenir un vol international, d'obtenir certains jours précis de congé ou autres.

Cappanera and al. [24] formulent le problème de *rostering* comme un problème de multi-flots binaires. Le problème est représenté sur un graphe temporel acyclique où les arcs représentent les activités et les noeuds représentent les temps de début et de fin des activités. L'objectif du modèle est de maximiser soit le nombre d'activités ou la durée des activités couvertes, tout en respectant les contraintes de conservation de flots, de capacité sur les arcs, de jours de travail, de ressources, d'activités assignées et de tâches interdites. Pour resserrer la formulation linéaire, les auteurs ont présenté des familles d'inégalités valides qui peuvent être regroupées en deux ensembles. Le premier est traité implicitement dans la phase préliminaire de traitement et le second doit être explicitement inclus dans le modèle. L'approche proposée a été testée sur sept instances provenant d'une compagnie aérienne italienne résolues à l'aide du solveur CPLEX. L'utilisation de la phase préliminaire de traitement permet d'obtenir de bons résultats avec un temps de traitement raisonnable.

Jarrah et al. [38] ont modélisé le problème de création d'horaires par enchère comme un problème de partitionnement d'ensemble dont l'objectif est de maximiser le nombre de temps de crédit couvert tout en minimisant le nombre total d'horaires. L'approche heuristique utilisée choisit tout d'abord un sous-ensemble de rotations, puis génère les horaires possibles pour ces rotations. Afin de restreindre le nombre d'horaires générées, deux techniques sont utilisées pour éliminer le plus tôt possible les horaires partiels qui ne pourront respecter les contraintes de temps de crédit minimum. Le problème de partitionnement d'ensemble est résolu à l'aide du solveur OSL de IBM. Les instances testées sont basées sur des données réelles d'une compagnie aérienne américaine et l'objectif est d'obtenir des solutions de bonne qualité avec un temps de traitement court. Les auteurs sont parvenus à obtenir des solutions satisfaisantes, c'est-à-dire des solutions avec des temps de crédit non couverts variant entre 0 et 3,51%. De plus, le temps de traitement de l'instance de plus grande taille est de 11 minutes comparativement à 15 jours lorsque le problème est résolu manuellement.

2.2 La récupération des opérations

Les opérations planifiées sont très souvent perturbées. En 2007, près d'un vol sur quatre atterrissant aux États-Unis accusait un délai de plus de 15 minutes ([12]). L'utilisation des réseaux en étoile implique qu'une petite perturbation, telle un retard ou une annulation de vol, peut avoir un impact significatif sur le reste du réseau. Ainsi, un temps d'embarquement trop long sur un vol, une grève des employés au sol ou un temps de préparation d'appareil allongé peuvent entraîner des répercussions dans l'ensemble du réseau. L'indisponibilité d'un appareil causée par un entretien non planifié ainsi qu'une diminution significative du nombre d'atterrissages et de décollages aux aéroports ou la fermeture d'un aéroport causée par des intempéries ont des répercussions encore plus importantes sur le réseau et peuvent durer plusieurs jours.

Lors de perturbations dans les opérations, les compagnies doivent rétablir les opérations normales le plus rapidement possible, généralement au plus tard le jour suivant. Elles disposent d'une période de récupération qui définit à quel moment les opérations planifiées devront être rétablies. Durant cette période, les opérations de récupération pour les appareils, les équipages et les passagers doivent être planifiées. De plus, une fois la période de récupération terminée, les appareils et les équipages doivent être localisés à l'emplacement déterminé par les opérations planifiées.

Suite à une perturbation, le planificateur peut avoir recours à une ou plusieurs des options suivantes : annuler des vols, retarder des vols (retarder intentionnellement un vol permet de réduire le nombre de connexions manquées et par conséquent réduire le temps de retard total des passagers), modifier l'affectation d'un ou plusieurs appareils ou équipages, modifier les itinéraires des avions ou utiliser des équipages de réserve (Ball et al. [13]). Les contraintes des problèmes d'affectation de la flotte, de création des itinéraires d'avions et de rotations des équipages doivent toujours être satisfaites. De plus, plusieurs objectifs peuvent être poursuivis et parfois les compagnies tenteront de minimiser une combinaison de deux ou de plusieurs des objectifs suivants : minimiser les coûts d'exploitation de la récupération : les coûts opérationnels des vols créés, les coûts de retard ou d'annulation, moins les coûts opérationnels des vols annulés, minimiser le nombre d'annulations, minimiser le nombre de vols retardés, minimiser le retard maximum, minimiser le ressenti passager ou minimiser le retard total des passagers (Luo et al. [43]). Les problèmes de récupération doivent normalement être résolus très rapidement, c'est-à-dire en quelques minutes. De plus, beaucoup d'informations sur la

situation actuelle doivent être considérées afin de s'assurer de respecter les contraintes. En effet, il est nécessaire de connaître le temps de vol de tous les appareils pour respecter les contraintes d'entretien, le temps de vol des membres des équipages pour respecter les règles des contrats de travail, les temps de porte à porte estimés pour tous les segments de vol et l'information sur tous les passagers pour s'assurer qu'ils se rendent à destination.

Tout comme pour les problèmes de planification tactique, la taille et la complexité des problèmes impliquent que la récupération est généralement effectuée de façon séquentielle. En effet, puisque les opérations de récupération doivent être planifiées en temps réel, l'optimisation globale des problèmes serait beaucoup trop longue. Par conséquent, le problème de récupération des appareils est généralement résolu en premier lieu. C'est à cette étape que sont prises les décisions concernant la réaffectation des appareils de la flotte, les modifications d'itinéraires des appareils et les annulations ou retards intentionnels de segments de vol. Ensuite, le problème de récupération des équipages est résolu, et ce, en réaffectant un sous-ensemble d'équipages ou en utilisant des équipages de réserve. Pour terminer, le problème de récupération des passagers est résolu. Seuls les passagers dont l'itinéraire a été affecté par les perturbations sont considérés et les itinéraires des passagers non affectés ne peuvent être modifiés, c'est-à-dire qu'il n'est pas possible de donner le siège d'un passager non affecté à un passager affecté.

Une revue exhaustive des différentes méthodologies de résolution des problèmes de récupération a été réalisée par Clausen et al. [26]. Une brève description des étapes de la planification tactique est effectuée et une classification des problèmes de récupération des appareils, des équipages, des passagers et de récupération intégrée est faite en fonction des méthodologies utilisées.

2.2.1 La récupération des appareils

Le problème consiste à déterminer l'heure de départ des vols, les annulations de vol et les nouveaux itinéraires des appareils, et ce, en minimisant la somme des coûts de retard et d'annulation. Les itinéraires des appareils peuvent être modifiés de plusieurs façons : repositionnement d'un appareil sans passagers ; déviation du vol vers un aéroport voisin ; déviation du vol vers la destination suivante *over-flying* ; échanges de segments de vol entre différents appareils ; retard intentionnel de segment de vol ; annulation de segment de vol.

Les décisions liées à ces différentes options doivent respecter les contraintes d'entretien, les contraintes de couvre-feu (qui varient selon les aéroports), les contraintes d'équilibrage et les contraintes de repositionnement. Pour s'assurer de respecter les contraintes d'entretien, l'itinéraire des appareils nécessitant un entretien durant la période de recouvrement (appareils critiques) ne sera généralement pas modifié. Les contraintes de balancement augmentent la complexité des décisions d'annulation, puisque l'annulation d'un segment de vol donné implique généralement l'annulation du segment de vol inverse. Pour pouvoir annuler un seul segment de vol, il est nécessaire d'utiliser un appareil de réserve. Toutefois, la quantité de ces appareils est très limitée. Pour terminer, les contraintes de repositionnement garantissent qu'à la fin de la période de récupération, les appareils sont repositionnés de façon à permettre la reprise des opérations planifiées.

L'objectif du problème de récupération des appareils varie selon les modèles développés, mais généralement les modèles tentent de minimiser la somme des coûts de retard et des coûts d'annulation. Les coûts de retard et d'annulation utilisés dans les modèles sont les coûts opérationnels. Toutefois, il est possible d'inclure dans ces coûts, des coûts liés au resenti passager.

Les coûts opérationnels d'un vol peuvent comprendre les coûts d'exploitation liés au type d'appareil et à la durée du vol (essence, entretien, équipage), les coûts d'exploitation liés au nombre de passagers et les coûts de services aéroportuaires. Les coûts opérationnels d'un retard ou d'une annulation peuvent comprendre un remboursement du prix du billet en cas d'annulation et une compensation monétaire en cas d'annulation. La valeur de la compensation dépend généralement de la durée prévue de l'itinéraire et une indemnité quotidienne (collation, repas, nuit d'hôtel). La valeur des indemnités dépend de la durée du retard et également de la durée de l'itinéraire (le retard permis avant le paiement d'une indemnité dépend de la durée prévue de l'itinéraire).

Teodorović et al. [62] ont développé un modèle de réseau pour résoudre le problème de pénurie d'appareils. L'objectif du modèle est la minimisation du délai total des passagers. Le modèle considère une flotte homogène d'appareils et est résolu à l'aide d'une méthode de branch-and-bound. Une instance comportant trois appareils et huit vols est résolue à l'optimalité. Toutefois, aucune instance réaliste n'a été solutionnée.

Jarrah et al. [39] ont développé deux modèles de flots sur le réseau permettant de résoudre le problème de pénurie d'appareils occasionné par des problèmes mécaniques, des conditions

météorologiques défavorables ou des retards sur des vols entrants. Les deux modèles développés prennent en considération des retards et des annulations multiples, la possibilité d'échange d'appareils entre les segments de vol et l'utilisation d'appareils de rechange. Toutefois, les modèles ne considèrent que des flottes homogènes d'appareils. L'objectif des deux modèles est de minimiser les coûts de retards et d'annulations. Le premier modèle résout le problème en retardant un certain nombre de vols jusqu'au retour de l'appareil en pénurie, tandis que le second modèle détermine de façon optimale l'ensemble des vols qui devront être annulés. Les deux modèles ont été testés sur des données provenant de United Airlines et le premier modèle permet, pour certaines instances, d'obtenir une réduction de coût allant jusqu'à 90%. Toutefois, ce modèle n'est efficace que lorsque le volume de vols est suffisamment élevé. Pour les deux modèles le temps de traitement est minimal.

Pour la récupération des flottes d'appareils d'un seul type, Argüello et al. [10] ont développé un GRASP. L'algorithme est composé d'une phase de construction et d'une phase de recherche locale. Lors de la recherche locale, l'algorithme choisit aléatoirement une solution dans une liste de candidats, examine les solutions voisines de cette dernière et insère la plus attrayante dans la liste des candidats. Pour créer les solutions voisines, les trois opérations suivantes sont effectuées : l'augmentation des routes, l'échange partiel des routes et l'annulation d'un circuit. L'algorithme a été testé sur 6068 instances provenant des données de Continental Airlines. Chaque instance comportent 42 vols, 16 appareils et 13 aéroports. La solution optimale a été obtenue dans 5,8% des cas, tandis que pour 90% des instances testées, la solution obtenue se situait à moins de 10% de la borne inférieure.

Cao et al. [22] définissent le problème de récupération d'appareils à l'aide d'un modèle de programmation quadratique prenant en considération les décisions de retarder intentionnellement certains vols ainsi que les décisions d'annuler certains vols. De plus, le modèle considère les effets de réseau entre les différents aéroports et quantifie les répercussions des retards et des annulations sur les aéroports subséquents. Les rotations d'équipages ne sont pas traitées et dans le modèle initial, seules les flottes homogènes d'appareils sont considérées. Toutefois, certaines extensions du modèle sont traitées pour prendre en compte la possibilité d'utiliser des appareils de surplus, de repositionner des appareils et d'utiliser des appareils de types différents. Les auteurs identifient également certaines propriétés spéciales du modèle développé. Cao et al. [23] développent un algorithme approximatif de programmation linéaire pour résoudre le modèle. Les propriétés spéciales du modèle permettent d'obtenir des solutions acceptables avec un temps de traitement raisonnable. De plus, l'algorithme

approximatif retourne un ensemble d'optima locaux de bonne qualité. Ceci peut permettre, lors de la prise de décision, de prendre en compte plusieurs autres facteurs non considérés dans le modèle.

Rosenberger et al. [52] ont développé un modèle de partitionnement d'ensemble pour résoudre le problème de récupération d'appareils. Le modèle assure que les segments de vol soit affectés exactement à une route ou annulés. L'objectif est de minimiser les coûts d'affectation des segments de vol et les coûts d'annulation. Afin de s'assurer que les opérations planifiées soient reprises après la période de perturbation, une pénalité est imposée lorsque le dernier segment de vol d'un appareil n'est pas le segment de vol planifié. Avant de résoudre le problème, l'ensemble des routes réalisables sont générées. Toutefois, puisque le nombre total de routes réalisables peut devenir très grand pour des problèmes réalistes, une méthode heuristique est utilisée pour déterminer un sous-ensemble d'appareils qui pourront être utilisés lors de la création des routes. L'approche développée a été testée sur trois instances de taille réelle et permet de résoudre le problème en quelques secondes.

Eggenberg et al. [30] ont développé un modèle de réseau pour le problème de récupération des appareils. Un algorithme de programmation dynamique est utilisé pour générer des réseaux qui crée, de façon itérative, les noeuds et les arcs, et un réseau est créé pour chaque ressource. Ensuite, un algorithme de génération de colonnes est utilisé. L'algorithme est testé sur des instances réelles provenant d'une compagnie aérienne ainsi que sur des instances créées à partir de données réelles et permet d'obtenir, lorsque les maintenances ne sont pas considérées, des solutions optimales avec un temps de calcul raisonnable. Lorsque les maintenances sont considérées, toutes les instances sont résolues avec un écart d'optimalité inférieur à 2%.

Dozic et al. [29] ont développé une heuristique de trois phases pour résoudre le problème de récupération des appareils. La première phase consiste à générer une solution réalisable en annulant temporairement une boucle contenant les vols perturbés. Lorsqu'il est impossible de retirer une boucle de vols, le vol perturbé ainsi que tous les vols suivants dans la rotation sont annulés. La deuxième phase tente de réintégrer les vols temporairement annulés aux rotations d'appareils. Finalement, la dernière phase consiste à retirer une partie d'une rotation et à essayer de l'ajouter à une autre rotation, ou interchanger des parties de rotation entre deux appareils. L'heuristique retourne une liste de solutions réalisables.

Finalement, Xiuli et al. [64] ont développé une heuristique hybride qui combine un GRASP et un algorithme de recherche avec tabous.

2.2.2 La récupération des équipages

Puisque toutes les étapes de la récupération sont interreliées, la récupération des appareils entraîne nécessairement des perturbations dans les rotations et les horaires des équipages ainsi que des bouleversements dans l'itinéraire des passagers. Les annulations de vol, les retards intentionnels, les modifications des itinéraires d'avions, ainsi que les maladies et les grèves des membres de l'équipage causent un manque de personnel aux emplacements planifiés. Étant donné que pour les équipages techniques (pilotes), les coûts sont généralement plus élevés et qu'il est plus difficile de les affecter (en raison des qualifications nécessaires), les modèles développés dans la littérature portent plus souvent sur ce type d'équipage.

Trois options peuvent être utilisées pour la récupération des équipages : le repositionnement, l'utilisation d'équipages de réserve et une modification de l'affectation planifiée. Les coûts de repositionnement sont importants puisque les employés doivent utiliser des sièges de passagers. Les compagnies tenteront donc de minimiser le nombre de repositionnements nécessaires. Selon Lettovsky et al. [41], un membre de l'équipage peut être repositionné pour les raisons suivantes : lorsqu'un segment de vol inclus dans sa rotation est annulé (l'employé doit donc voyager en tant que passager jusqu'à l'origine du prochain segment de vol de sa rotation et ainsi reprendre son horaire planifié ou retourner à la base s'il ne reste plus de segments de vol dans sa rotation) lorsqu'un segment de vol est dévié vers un aéroport voisin (dans ce cas, le repositionnement peut être effectué par transport terrestre et les coûts de repositionnement sont moindres) ; lorsqu'un segment de vol inclus dans sa rotation est annulé et qu'il y a un échange d'affectation (l'employé est repositionné pour travailler sur un segment de vol dont un membre de l'équipage est absent) ; lorsque le temps de vol restant n'est pas suffisant (l'employé voyage en tant que passager pour retourner à sa base).

L'utilisation des équipages de réserve peut également être envisagée afin de s'assurer que tous les segments de vol possèdent le nombre d'employés nécessaire. Toutes les compagnies aériennes possèdent un certain nombre d'employés de réserve qui sont payés un salaire mensuel minimum garanti, mais qui ne possèdent pas d'horaire de vol (ils ne sont affectés à aucun segment de vol). Lorsque ces employés sont affectés à un ou plusieurs segments de vol, des coûts supplémentaires sont encourus par les compagnies. De plus, les membres de l'équipage

initialement affectés aux segments de vol doivent également être payés. Cette option est donc dispendieuse.

La troisième option est certainement la plus intéressante pour les compagnies aériennes car c'est l'option dont les coûts sont les moindres, soit la réaffectation des membres de l'équipage vers d'autres segments de vol. Toutefois, bien que moins dispendieuse, cette option est plus complexe. En effet, pour que la réaffectation à un nouvel horaire soit possible, les différentes contraintes de légalité (temps maximum de vol et de travail, temps minimum de repos, temps maximum de la rotation et autres) doivent être respectées. La structure de coût de cette option est également complexe car le salaire payé aux membres de l'équipage est le maximum entre celui de l'horaire planifié et celui du nouvel horaire.

L'objectif du problème de récupération d'équipages est de créer les nouveaux horaires (nouvelles rotations) pour les équipages affectés par les perturbations et les équipages de réserve, et ce, en minimisant les coûts ainsi que le nombre d'horaires modifiés. Minimiser le nombre d'horaires modifiés permet une reprise plus rapide des opérations planifiées et il est donc important d'en tenir compte dans les modèles. Toutefois, minimiser les coûts et les horaires modifiés rend le problème plus complexe.

En plus de respecter les contraintes légales, il faut s'assurer que tous les segments de vol soient couverts une seule fois, c'est-à-dire qu'un et un seul équipage est affecté à chaque segment de vol. Bien entendu, aucun équipage n'est affecté aux segments de vol qui ont été annulés. De plus, il faut s'assurer que les membres de l'équipage assignés à un nouveau segment de vol soient qualifiés pour le type d'appareil affecté au segment de vol. Cette règle est plus contraignante pour les équipages techniques, puisque ces derniers sont qualifiés pour opérer les appareils d'une seule famille.

Pour résoudre le problème de récupération des équipages, Stojković et al. [60] ont développé un modèle dont l'objectif est de couvrir tous les segments de vols durant la période de récupération de façon à minimiser les coûts et minimiser les perturbations pour les membres d'équipages. Le problème est décomposé en un problème maître formulé comme un problème de partitionnement d'ensemble et un sous-problème de plus courts chemins. Plusieurs scénarios sont générés à partir d'une instance comprenant 106 segments de vols provenant d'une compagnie aérienne américaine et les scénarios sont résolus à l'aide des solveurs GENCOL et CPLEX. Afin d'augmenter la vitesse de résolution, les auteurs utilisent une méthode heuris-

tique permettant d'effectuer des branchements rapides. Toutes les instances ont été résolues rapidement, c'est-à-dire en moins de 20 minutes.

Letovsky et al. [41] formulent le problème de récupération des équipages comme un problème de recouvrement d'ensemble dont l'objectif est de minimiser les coûts des rotations qui couvrent le plus grand nombre de segments de vol tout en limitant l'impact des délais et des annulations sur les passagers. La relaxation linéaire du problème est d'abord résolue, puis une méthode de branch-and-bound est utilisée pour obtenir des solutions entières. Trois stratégies de branchement sont utilisées : le branchement sur les variables d'annulations, le branchement sur les variables de repositionnement et le branchement sur les suites de deux segments de vol. Trois différents scénarios d'une instance comprenant 1 296 vols, 177 rotations et deux bases d'équipages sont utilisés pour tester l'approche. Le premier scénario représente une perturbation mineure liée à un problème d'entretien, le second scénario représente un problème météorologique qui cause un raletissement à un aéroport et le troisième représente une perturbation majeure dans trois aéroports. Les résultats démontrent que des problèmes concernant des perturbations moyennes peuvent être résolus avec un temps de traitement raisonnable.

Pour résoudre le problème de récupération d'équipage qui survient dans les opérations quotidiennes, Medard et al. [45] ont développé un modèle de recouvrement d'ensemble qui intègre les étapes de création des rotations et d'affectation des rotations aux membres d'équipages. L'objectif du modèle est de minimiser les coûts d'affectation. Des pénalités sont imposées pour tout changements à l'horaire planifié. Le problème est représenté sur un réseau d'activité acyclique qui est ensuite réduit en un réseau d'activité pour les groupes d'équipages. Le problème est résolu par deux méthodes. Tout d'abord, une méthode d'énumération gloutonne avec recherche en profondeur est utilisée pour générer les horaires mensuels des équipages. La seconde méthode est l'utilisation d'un générateur des colonnes à coût réduit. Les deux approches ont été testées sur 19 instances comportant une ou plusieurs bases d'équipages. Pour les instances de petite à moyenne taille, la méthode de génération de colonne est plus performante. Toutefois, pour les problèmes de grande taille comportant plusieurs bases d'équipages, le temps de traitement nécessaire pour la génération de colonnes est très grand.

Abdelghany et al. [4] ont développé un modèle de programmation mixte pour résoudre le problème de récupération des équipages. L'objectif est de récupérer un nombre maximal de segments de vol perturbés tout en minimisant les coûts d'affectation et de retard. L'utilisa-

tion d'équipages de réserve, les échanges de segments de vol entre les équipages et l'utilisation de repositionnements sont permis. La méthode développée résout une séquence de problème d'affectation de façon à effectuer les récupérations des vols en ordre chronologique des temps de départ. Plusieurs méthodes de traitement préliminaire sont utilisées, tels le déplacement des problèmes vers les terminaux principaux, l'ajout d'équipages non perturbés et le calcul des coûts d'affectation. La méthode de résolution est testée sur une étude de cas comportant 18 perturbations. La séquence de problèmes d'affectation est résolue à l'aide du solveur CPLEX avec un temps de traitement inférieur à deux minutes.

Nissen et al. [47] ont développé un modèle de récupération d'équipages considérant des coûts d'équipages fixes, et par conséquent, qui s'applique aux transporteurs européens. Le modèle est représenté sur un réseau espace-temps où les noeuds représentent un aéroport à un moment spécifique et les arcs représentent les périodes de travail. L'objectif est de minimiser les coûts de réaffectation. Les contraintes liées aux journées de travail sont implicitement considérées dans le réseau, tandis que les contraintes liées aux rotations doivent être explicitement considérées. La méthode de résolution choisit un sous-ensemble d'équipages (et les segments de vols qui leur sont affectés) pouvant être utilisés lors de la récupération. Puis, l'ensemble des journées de travail légales sont générées et un nouveau réseau est créé. Une méthode de branch-and-price est utilisée et les branchements sont effectués sur deux vols consécutifs. Afin de tester la méthode, cinq instances ont été générées pour 14 scénarios de perturbation différents. La méthode de résolution développée permet de résoudre les problèmes avec un temps de traitement très court.

2.2.3 La récupération des passagers

Les perturbations des itinéraires de passagers peuvent être causées par des annulations de vol, des retards de vol entraînant des connexions manquées et les différentes opérations de récupération des appareils et des équipages. Il existe très peu de littérature sur la récupération des passagers.

Tel que mentionné précédemment, les coûts liés au ressenti passager peuvent être inclus dans la fonction objectif lors de la récupération des appareils. Le ressenti passager est très difficile à estimer et est indépendant des compensations monétaires offertes aux passagers. Il représente l'attitude négative du passager face au service offert et a un effet à long terme sur les revenus et les profits des compagnies. Toutefois, il est difficile de quantifier l'attitude

des passagers. Le ressenti passager, suite à une opération de récupération, dépend du temps total de retard, de l'annulation et du type de récupération qui aura été fait (déclassement ou surclassement). Il existe plusieurs options possibles afin de récupérer les passagers, soit la modification de l'itinéraire des passagers affectés, l'utilisation de segments de vol d'une autre compagnie aérienne, l'utilisation d'un autre mode de transport et l'annulation de l'itinéraire des passagers.

Modifier l'itinéraire d'un passager affecté signifie trouver un itinéraire alternatif débutant à l'endroit où la perturbation s'est produite et se terminant à la destination finale du passager. De plus, un temps minimum entre le moment de la perturbation et le début du nouvel itinéraire doit être considéré pour permettre au passager d'effectuer leur connexion. L'itinéraire alternatif d'un passager peut également être effectué sur des vols d'une autre compagnie aérienne.

L'utilisation d'un autre mode de transport et l'annulation de l'itinéraire sont les alternatives qui auront l'impact le plus important sur la satisfaction du passager. Un autre mode de transport est généralement utilisé lorsque la distance à parcourir est relativement courte. Toutefois, cette option allonge le temps de transport du passager, bien que le temps total de l'itinéraire puisse être plus court. De plus, le confort des passagers peut être diminué. Bien entendu, l'annulation de l'itinéraire d'un passager n'est possible que si le passager n'a pas quitté le point d'origine.

Plusieurs contraintes doivent être respectées. Tout d'abord, les contraintes de capacité garantissent que le nombre de passagers sur tous les appareils ne dépassent pas la capacité des appareils. Les contraintes d'affectations assurent que tous les passagers sont affectés à un itinéraire ou que leur itinéraire est annulé. De plus, il faut inclure des contraintes qui assurent que les sièges des passagers non perturbés ne sont pas assignés à des passagers perturbés.

Bratu et al. [21] ont développé deux modèles pour résoudre le problème de récupération des passagers en déterminant quels segments de vol seront retardés et quels segments seront annulés. L'utilisation d'équipages et d'appareils de réserve est permise. Le premier modèle a pour objectif de minimiser les coûts d'opération et les coûts liés aux passagers perturbés, tandis que le deuxième a pour objectif de minimiser les coûts d'opération et les coûts liés aux passagers retardés. Pour ce faire, les auteurs discrétisent les fenêtres de temps et génèrent une copie d'arc pour chaque minute de la fenêtre et un algorithme est utilisé pour

éliminer les arcs dominés. Les modèles sont résolus à l'aide du solveur OPL Studio et trois instances basées sur des données provenant d'une compagnie américaine sont testées, soit une comportant un niveau élevé de perturbation, une comportant un niveau moyen et une comportant un niveau faible. Pour les trois instances, les modèles développés permettent de réduire substantiellement les délais et les perturbations des passagers.

Pour résoudre le problème de récupération des passagers, Zhang et al. [70] ont développé un modèle permettant l'utilisation d'une stratégie de substitution du mode de transport. La stratégie permet de choisir entre retarder intentionnellement des segments de vol, annuler des segments de vol et substituer des segments de vol par un autre mode de transport (autobus). Les auteurs ont développé un modèle de programmation non linéaire en nombres entiers possédant des contraintes linéaires et une fonction objectif non linéaire. Un algorithme approximatif basé sur une relaxation des contraintes d'intégralité est utilisé. Le modèle est testé sur une instance comprenant 40 vols entrants et 40 vols sortants. Le modèle permet d'obtenir une solution 3% plus élevée que la borne inférieure et permet de réduire le nombre de passagers perturbés de 90 à 14.

2.2.4 La récupération intégrée

L'objectif de la planification tactique dans le transport aérien est l'optimisation de toutes les opérations. Lorsque survient une perturbation, mineure ou majeure, les opérations optimisées sont compromises et des opérations de récupération doivent être planifiées. Toutefois, ces opérations diminuent significativement les profits des compagnies qui doivent encourir des coûts d'opération supplémentaires. De plus, les perturbations affectent la satisfaction des passagers, ce qui peut se traduire en perte de revenu pour les compagnies. Étant donné la fréquence élevée des perturbations dans le transport aérien, une amélioration des solutions du problème de récupération aura un impact important sur la rentabilité des compagnies. Puisque la résolution séquentielle des différents problèmes de récupération entraîne une sous-optimisation des solutions, il est important de développer des modèles intégrant la récupération des appareils, des équipages et des passagers. Une résolution simultanée des trois problèmes pourrait permettre des réductions de coûts substantielles pour les compagnies aériennes. Quelques recherches portent sur la résolution intégrée des trois problèmes, tandis que d'autres recherches portent sur l'intégration de deux étapes de récupération.

La récupération intégrée des appareils, équipages et passagers

Petersen et al. [50] ont utilisé une méthode de décomposition de Benders pour résoudre le problème intégré de récupération. Le problème maître est le problème d'horaire, tandis que les problèmes de récupération d'appareil, de récupération d'équipages et de récupération des passagers sont les sous-problèmes. Afin de permettre une résolution, la taille du problème est réduite en ne considérant que les vols, les équipages et les passagers qui ont été perturbés. Ensuite, des coupes de Benders et la génération de colonnes sont utilisés simultanément. Leur méthode a été testée sur des données provenant d'une compagnie aérienne américaine et permet une réduction du temps total de retard des passagers.

Pour résoudre le problème intégré de récupération, Zhang et al. [69] ont développé un modèle de partitionnement d'ensemble et ont utilisé un algorithme basé sur un horizon roulant. Des données provenant d'une compagnie aérienne américaine ont été utilisées pour tester leur algorithme qui permet une réduction importante des retards. Toutefois, la taille des instances est petite avec 351 vols, 70 appareils du même type et 134 équipages.

La récupération des appareils et des équipages

Pour le problème de perturbations causées par des retards, Luo et al. [43] ont développé deux modèles qui considèrent à la fois la réaffectation d'appareils et la réaffectation d'équipages. Le premier modèle a pour objectif la minimisation du retard maximal et le deuxième, la minimisation du nombre total de vols retardés par plus de 15 minutes. Afin de résoudre le deuxième modèle, le nombre de variables binaires a été réduit. De plus, quatre inégalités valides ont été dérivées afin de resserrer les bornes de la relaxation linéaire et une heuristique basée sur le modèle restreint a été développée. L'heuristique a été testée sur une instance de 71 vols entrants provenant de American Airlines. Celle-ci a permis une amélioration de 46% de la solution avec un temps de traitement minimal.

Stojković et al. [61] ont développé une méthode heuristique pour résoudre le problème de récupération des appareils et des équipages suite à une perturbation mineure. L'objectif est de déterminer les nouveaux horaires en modifiant les temps de départ, d'arrivée, de durée de vol, d'entretien et de connexion des passagers, tout en considérant que les itinéraires des appareils et des équipages sont conservés. Un modèle de programmation linéaire est développé et les auteurs démontrent que le dual du problème est un problème de réseau pouvant être résolu en temps linéaire. Dix instances réalistes ont été générées et les problèmes pri-

mal et dual ont été résolus à l'aide du solveur CPLEX. Les résultats confirment que les temps d'exécution pour les problèmes primal augmentent de façon polynomiale en fonction de la taille des problèmes, tandis qu'ils augmentent de façon linéaire pour les problèmes dual.

Pour résoudre le problème de récupération intégré incluant la récupération des appareils, des membres d'équipages techniques et des agents de bord, Abdelghany et al. [5] ont développé un système à plusieurs étapes intégrant un modèle de simulation et un modèle de programmation mixte. À chaque étape, le modèle de simulation détermine toutes les violations futures qui seront occasionnées par les perturbations et une banque de ressources disponibles est ensuite créée. Quatre options sont permises pour créer la banque, soit l'utilisation de ressources de réserve, de ressources perturbées, de ressources dont les vols ont été annulés et de ressources non perturbées. Puis, le modèle d'optimisation examine les échanges de ressources possibles et les retards intentionnels afin de minimiser les délais et les annulations. Une instance comprenant 522 appareils, 1 360 membres d'équipages techniques, 2 040 agents de bord et 1 100 segments de vol desservant 112 villes est utilisée pour tester le modèle et une perturbation en début de journée impliquant trois aéroport est simulée. Le système développé permet une réduction de 8,7% des délais avec un temps de traitement de 36 secondes.

La récupération des appareils et des passagers

Zegordi et al. [68] ont développé un algorithme de colonie de fourmis pour résoudre le problème de récupération des appareils tout en considérant dans la fonction objective les coûts des délais de passagers et d'annulations. Leur algorithme a été testé sur des instances de petite taille provenant de données réelles. Jafari et al. [36, 37] ont représenté le problème à l'aide d'un modèle de programmation linéaire mixte qui utilise des rotations d'appareils et des itinéraires de passagers plutôt que des segments de vols. Les mêmes instances que Zegordi et al. [68] ont été utilisées et les auteurs démontrent que de considérer les passagers dans la fonction objectif permet une diminution du coût total.

2.3 Horizon roulant

L'utilisation d'horizons roulants lors de la planification est très utile dans un environnement dynamique où l'information nécessaire à la prise de décision est imparfaite. Un cadre

de résolution basé sur un horizon roulant permet de ne considérer qu'une partie de l'horizon de planification à chaque itération. L'information sur la période considérée et une partie de l'information des périodes suivantes sont utilisées lors de l'optimisation du problème, mais seulement les décisions liées à la période considérée sont appliquées. Il existe une riche littérature sur l'utilisation d'horizons roulants pour résoudre des problèmes de gestion des opérations, tels la planification de la production, la gestion de stocks et la gestion de la capacité. Une revue exhaustive des différents problèmes de gestion des opérations résolus à l'aide d'un cadre d'horizon roulant a été effectuée par Chand et al. [25].

Toutefois, la littérature portant sur l'utilisation d'horizons roulants pour résoudre des problèmes de transport de passagers est très limitée et s'applique principalement aux problèmes où l'information disponible est incomplète comme pour les problèmes de transport ferroviaire de passagers où les paires origine-destination des passagers sont inconnues. Une approche heuristique a été développée par Törnquist [63] pour résoudre le problème de la gestion des perturbations dans le transport ferroviaire. L'heuristique modifie l'horaire initial en allouant les voies ferrées aux trains sans considérer quelles voies leur étaient initialement assignées. Les séquences de trains sont conservées, tout en permettant à certains trains de laisser passer un ou plusieurs autres trains sur la voie. Les instances utilisées pour tester l'heuristique ont été générées à partir d'information réelle sur le transport ferroviaire de la région de Norrköping en Suède et considère à la fois des trains de passagers, des trains de marchandises et des trains de service. L'heuristique permet d'obtenir, pour l'ensemble des instances testées, des solutions optimales ou très près de l'optimum, et ce, moyennant un temps de calculs très court.

Afin de résoudre le problème de récupération du matériel roulant dans le transport ferroviaire, Nielsen et al. [46] ont résolu un modèle de multi-flots intégré dans un cadre d'horizon roulant. Leur méthodologie a été testée sur des instances réelles provenant de Netherlands Railways et a permis d'obtenir des solutions de qualité avec un temps de calcul de quelques secondes, et ce, même pour les plus grandes instances.

Lors de la résolution des problèmes de récupération dans le transport aérien, toute l'information nécessaire à la prise de décision est disponible. Toutefois, l'utilisation d'une approche d'horizon roulant est appropriée en raison de la complexité et la taille des problèmes de récupération. Cette approche permet donc de résoudre de très grandes instances avec un temps de calcul acceptable.

Tel que mentionné à la section 2.2.4, Zhang et al. [69] ainsi que Abdelghany et al. [5] ont utilisée une approche d’horizon roulant pour résoudre des problèmes de récupération. Zhang et al. [69] ont développé un algorithme basé sur un horizon roulant pour résoudre le problème intégré de récupération des appareils, des équipages et des passagers dans le transport aérien. Les auteurs ont développé un modèle de programmation en nombres entiers basé sur un problème de partitionnement d’ensemble où les échanges ne sont permis qu’à l’intérieur de la période considérée ou des périodes précédentes. L’algorithme permet d’obtenir des diminutions de retard importantes et de diminuer le nombre de vols affectés de 10%, et ce, en quelques secondes de temps de calculs. Toutefois, la taille des instances testées est très petite. Abdelghany et al. [5] ont développé un système utilisant une approche d’horizon roulant qui intègre un modèle de simulation et un modèle de programmation mixte pour le problème conjoint de récupération des appareils et des équipages. À chaque itération, seulement une partie de la période de récupération est considérée et de nouvelles rotations d’appareils et d’équipages sont créées pour l’ensemble des segments de vols qui ne respectent pas les contraintes.

Jafari et al. [36] ont présenté un modèle d’affectation intégré à un cadre d’horizon roulant pour le problème conjoint de récupération des appareils et des passagers. Le modèle considère les retards de segments de vol, l’échange d’appareils, l’utilisation d’appareils de réserve, le repositionnement d’appareils, l’annulation d’itinéraires et la réaffectation des passagers. À chaque étape, le modèle tient compte de l’impact des décisions sur les horizons futurs et sélectionne les appareils qui ne causeront pas de perturbations aux étapes suivantes. Toutefois, les instances considérées sont petites et ne contiennent que très peu de perturbations.

Saddoune et al. [54] ont développé une heuristique d’horizon roulant pour le problème d’affectation des rotations aux membres d’équipages. L’horizon de planification est divisé en périodes qui se recoupent et à chaque période le problème est résolu par génération de colonne. Les solutions obtenues ont été comparées aux solutions obtenues avec la méthode en trois phase généralement utilisée. L’heuristique d’horizon roulant permet d’obtenir de meilleures solutions car il permet la répétition des numéros de vols. De plus, l’heuristique nécessite un temps de calcul inférieur et produit des solutions plus régulières.

Une approche d’horizon roulant intégrant une heuristique de recherche avec tabous a été proposée par Furini et al. [33] pour résoudre le problème de d’ordonnancement des appareils.

Les auteurs ont présenté deux modèles en nombres entiers et ont proposé plusieurs règles de partition des séquences d'appareil. L'heuristique a été testée sur des instances réelles provenant de l'aéroport Milano Linate et a permis l'obtention de la solution optimale pour toutes les instances.

2.4 Challenge ROADEF 2009

Bien qu'il existe plusieurs recherches portant sur l'intégration d'au moins deux des trois problèmes de récupération, les instances considérées sont généralement de petite taille et ne contiennent qu'un nombre très limité de perturbations. Toutefois, le problème de récupération des appareils et des passagers a été étudié lors du Challenge ROADEF 2009 et les instances proposées sont de taille considérable et comportent un grand nombre de perturbations. La variante introduite lors du Challenge est décrite par Palpant et al. [48]. Au total, neuf équipes ont participé à la finale de la compétition, dont seulement quatre ont obtenu au moins une des meilleures solutions connues. Bisailon et al. [19] ont développé une heuristique de recherche à grand voisinage qui alterne entre trois phases : construction, réparation et amélioration. Les deux premières phases essaient de créer une solution réalisable en retardant des vols, en annulant des vols ou en retirant des boucles de segments de vols. La troisième phase essaie d'améliorer la solution en effectuant des grands changements dans l'horaire des vols. Ces auteurs ont obtenu la première place lors du Challenge.

Mansi et al. [44], l'équipe qui a terminé en deuxième position, ont proposé une heuristique de deux étapes. La première étape consiste à trouver une solution réalisable en résolvant une relaxation du problème. Si aucune solution réalisable n'est obtenue, un algorithme de programmation dynamique est utilisé pour trouver des rotations d'appareils réalisables. La deuxième étape cherche à améliorer la solution en utilisant une stratégie d'oscillation qui alterne entre une phase de destruction et une phase de construction. La phase de destruction annule des séquences de vols et les itinéraires qui utilisaient ces séquences, puis réassigne les passagers perturbés à des vols existants, tandis que la phase de construction génère des rotations d'appareils et des itinéraires de passagers et les attribue simultanément aux appareils et aux passagers.

Pour la même compétition, Jozefowicz et al. [40] ont développé une heuristique de trois phases. Dans la première phase les perturbations sont intégrées à l'horaire planifié et une solution réalisable est créée en retirant des vols et des itinéraires de passagers. La seconde phase

consiste à réintégrer, lorsque possible, les vols annulés aux rotations d'appareils. Finalement, la troisième phase cherche à créer des nouveaux vols pour réaccommoder les passagers dont les itinéraires ont été annulés.

Pour terminer, Peekstok et al. [49] ont développé un algorithme de recuit simulé. Initialement, l'algorithme accepte les solutions non réalisables relativement aux appareils, aux aéroports et aux passagers et introduit un second terme dans la fonction objectif afin de quantifier le niveau de non-conformité. Toutefois, lorsque les solutions deviennent réalisables relativement aux appareils et aux aéroports, elles ne peuvent plus redevenir non réalisables.

Une autre recherche sur le sujet a été faite après le Challenge. Acuna-Agost [6] a développé une heuristique de post-optimisation pouvant être combinée à une méthode de résolution du problème de récupération des appareils. Deux algorithmes sont utilisés pour réduire le nombre de variables et de contraintes en identifiant les noeuds du réseau qui sont incompatibles ou sous-optimaux. Cette heuristique, combinée à l'heuristique de Jozefowicz et al. [40], permet une grande amélioration des solutions.

La définition de la variante du problème présenté lors du Challenge ROADEF 2009 ainsi que les instances introduites offrent une opportunité de développer de nouvelles méthodes de résolution. Les trois prochains chapitres proposent des méthodes heuristiques pour résoudre le problème complexe de récupération intégrée des appareils et des passagers, tel que défini lors du Challenge, ainsi que la création de nouvelles instances de très grandes tailles comportant un nombre important de perturbations.

Le chapitre 3 de cette thèse présente un algorithme de recherche à grand voisinage (RGV). La méthode de résolution est basée sur l'heuristique développée par Bisailon et al. [19] qui alterne entre trois phases : construction, réparation et amélioration. De nombreux changements ont été apportés à chacune des phases afin d'explorer plus exhaustivement l'espace de solution.

Dans la phase de construction l'heuristique crée un réseau pour chaque appareil disponible et tente de recréer les arcs de vols annulés à l'aide d'un algorithme du plus long chemin. De plus, lorsque la capacité des aéroports est réduite, l'algorithme considère les vols en ordre décroissant de coût d'annulation. Après la phase de réparation, s'il subsiste des passagers dont l'itinéraire est annulé, l'heuristique tente de réacheminer ces passagers en utilisant les

appareils disponibles, soit en résolvant un problème de multi-flot avec contraintes additionnelles ou à l'aide d'un algorithme du plus court chemin.

Finalement, quatre étapes ont été ajoutées à la phase d'amélioration. Tout d'abord, afin de diversifier la recherche, l'heuristique détruit soit une partie d'une rotation ou la rotation complète. Ensuite, afin de réacheminer les passagers dont l'itinéraire a été annulé, l'algorithme essaie de créer de nouveaux vols, en utilisant les appareils disponibles, à l'aide d'un algorithme du plus court chemin. La deuxième amélioration considère les passagers se trouvant à un aéroport perturbé et essaie de les rediriger vers d'autres aéroports en utilisant les vols existants. Puis, l'heuristique tente de créer de nouveaux vols permettant de réacheminer ces passagers à leur destination finale. La troisième amélioration consiste à retarder des itinéraires de passagers subséquents afin de réacheminer des passagers additionnels. Finalement, la dernière amélioration tente de réparer les positions des appareils en les transférant, lorsque possible, à l'aéroport planifié.

L'algorithme a été testé sur les instances introduites lors du Challenge ROADEF 2009 et a permis d'obtenir les meilleures solutions connues pour 17 des 22 instances avec un temps de calcul de cinq minutes et 21 des 22 instances avec un temps de calcul de 10 minutes.

Le chapitre 4 présente une méthode de résolution qui combine l'heuristique RGV présentée au chapitre 3 et une heuristique de post-optimisation basée sur la génération de colonnes. Le problème conjoint de récupération des appareils et des passagers est représenté à l'aide d'un modèle de programmation mixte. Ensuite, la taille du modèle est réduite en regroupant les passagers et les appareils ayant les mêmes caractéristiques. Afin de permettre la résolution des plus grandes instances, le modèle de programmation mixte est modifié pour ne considérer que les passagers. Il en résulte un modèle de multi-flots.

Le problème est initialement résolu en appliquant l'heuristique RGV. Puis, les arcs d'appareils et de passagers obtenus avec l'heuristique RGV sont intégrés au problème maître restreint (PMR) initial. La relaxation linéaire du PMR est ensuite résolue et les coûts réduits des variables de passagers sont calculés. Les variables ayant un coût réduit négatif sont intégrées au modèle et la relaxation linéaire du PMR est à nouveau résolue. La phase de relaxation linéaire se termine lorsqu'un certain nombre d'itérations ont été effectuées. Pour terminer, soit le problème de programmation mixte ou le problème de multi-flot est résolu.

L’heuristique a été testée sur les 22 instances du Challenge ROADEF 2009 et a permis une amélioration significative des meilleures solutions connues pour l’ensemble des instances avec un temps de calcul raisonnable.

Le chapitre 5 présente une approche d’horizon roulant qui intègre l’heuristique de post-optimisation. À chaque itération, seulement une partie de la période de récupération est considérée et l’heuristique de post-optimisation est appliquée. Cette méthode permet d’obtenir de bons résultats tout en réduisant significativement le temps de calcul nécessaire. Ce chapitre présente également de très grandes instances créées à partir des instances du Challenge ROADEF 2009. Ces instances comportent un nombre important de perturbations d’appareils, de vols et d’aéroports. L’heuristique d’horizon roulant permet de résoudre ces très grandes instances avec un temps de calcul raisonnable et améliore les solutions obtenues avec l’heuristique RGV, particulièrement lorsque le nombre de perturbations d’aéroports est important.

Chapitre 3

Improvements to a Large Neighbourhood Search Heuristic for an Integrated Aircraft and Passenger Recovery Problem

Information sur le chapitre

Un article basé sur ce chapitre a été publié dans la revue *European Journal of Operational Research* : K. Sinclair, J.-F. Cordeau, G. Laporte. Improvements to a Large Neighbourhood Search Heuristic for an Integrated Aircraft and Passenger Recovery Problem. *European Journal of Operational Research* 233(1) : 234-245, 2014.

Abstract

Because most commercial passenger airlines operate on a hub-and-spoke network, small disturbances can cause major disruptions in their planned schedules and have a significant impact on their operational costs and performance. When a disturbance occurs, the airline often applies a recovery policy in order to quickly resume normal operations. We present in this paper a Large Neighbourhood Search heuristic to solve an integrated aircraft and passenger recovery problem. The problem consists of creating new aircraft routes and passenger itineraries to produce a feasible schedule during the recovery period. The method is based on an existing heuristic, developed in the context of the 2009 ROADEF Challenge, which alternates between three phases : construction, repair and improvement. We introduce

a number of refinements in each phase so as to perform a more thorough search of the solution space. The resulting heuristic performs very well on the instances introduced for the challenge, obtaining the best known solution for 17 out of 22 instances within five minutes of computing time and 21 out of 22 instances within 10 minutes of computing time.

Keywords : airline recovery, fleet assignment, aircraft routing, passenger itineraries, large neighbourhood search.

3.1 Introduction

In order to successfully manage their expensive resources, commercial passenger airlines must make use of efficient planning systems. For this reason, operational research plays a major role in the airline industry's tactical planning. As was documented by Barnhart et al. [14], this industry provides a favorable environment for the application of operations research (OR) models and techniques. The intensive use of computers also explains the importance of OR in the airline industry.

Because of their size and complexity, tactical planning problems are usually solved in a sequential order and are divided into five phases. Airlines must first determine which cities they will service and create a flight schedule. They then need to determine which type of aircraft will be assigned to the different flight legs of the schedule. In the third phase, airlines must create, for each aircraft, rotations that connect the flight legs while respecting maintenance constraints. Next, airlines must create crew pairings while respecting complex government and work related constraints. Finally, these crew pairings are combined to form monthly schedules for all crew members.

Since most commercial passenger airlines operate on a hub-and-spoke network, small disturbances can cause major disruptions in their planned schedules. Although disruptions can be caused by many factors such as cancelled or delayed flights, unavailable aircraft or crews, security measures and airport congestion, unfavorable weather conditions are the primary cause and according to Rosenberg et al. [52], they are responsible for 75% of all disturbances. Disruptions have a significant impact on the operational costs and profits of airlines. Therefore, when they occur, it is necessary for the airlines to re-establish the planned schedule as quickly as possible, usually by the following day. Also, recovery problems need to be solved

in a very short period of time, usually within minutes. Because of the size of the recovery problems and the time constraint, similar to the tactical planning, the recovery process is usually performed in a sequential way. First, the aircraft recovery problem is solved by either cancelling or delaying flights, modifying aircraft rotations or reassigning available aircraft while respecting maintenance, flow and location constraints. The second phase is crew recovery, which can be done by reassigning some crews, by deadheading crew members or by using reserve crews. Finally, the passenger recovery problem is solved. Although this sequential process can produce optimal solutions for the different phases, it rarely yields optimal solutions for the entire system. Hence, solving the problem globally, i.e. integrating the different recovery phases, should yield better solutions. Because disturbances occur frequently in the airline industry and their financial impact is substantial, it is necessary for the airlines to develop integrated approaches for the recovery problem.

Different methods have been developed to solve the aircraft recovery problem. Teodorović et al. [62] proposed a heuristic based on a network flow model and used a branch-and-bound method, while Jarrah et al. [39] also used network flow models, but their algorithm solves a shortest path problem repeatedly. Arguello et al. [10] applied a greedy randomized adaptive search procedure (GRASP) consisting of a construction phase and a local search phase. Cao et al. [22, 23] developed a linear programming approximation algorithm based on a quadratic programming model. Rosenberg et al. [52] solved a set partitioning problem and used a heuristic to reduce the number of feasible routes. More recently, Eggenberg et al. [30] presented a constraint specific network recovery model which they solved by column generation. As for the crew recovery problem, Stojković et al. [60] modeled the problem as a set partitioning problem while Lettovsky et al. [41] and Medard et al. [45] modeled it as a set covering problem. Abdelghany et al. [4] developed a mixed integer programming model. Other methods have been proposed to solve the crew recovery problem (see e.g. Nissen et al. [47]; Yu et al. [67]). To solve the passenger recovery problem, Bratu et al. [21] used network flow techniques, while Zhang et al. [70] developed an integer non-linear programming model.

Although, to our knowledge, only Petersen et al. [50] addressed the full recovery problem (i.e. aircraft, crew and passenger recovery) and solved it using a Benders decomposition scheme, several methods have been developed to solve two recovery problems jointly. Abdelghany et al. [5] used a simulation model and a resource assignment optimization model to solve the joint aircraft and crew recovery problem. Luo et al. [43] modeled this joint problem as an integer linear program and solved it with a heuristic based on a restricted version of the

model, while Stojković et al. [61] also modeled the problem as an integer linear program but solved it by network flow techniques. As for the joint aircraft and passenger recovery problem, Bratu and Barnhart [21] presented two models and developed an Airline Operations Control Center simulator to evaluate their impact on a major US airline's operations. Their models also consider crew recovery, but only use an approximation of reserve crews and do not consider the disrupted crews. Zegordi et al. [68] solved the aircraft recovery problem using an ant colony algorithm while taking into consideration disrupted passengers in the objective function. To solve the joint aircraft and passenger recovery problem, Jafari et al. [36] developed a mixed integer programming model. Finally, Ball et al. [13] and Clausen et al. [26] offer surveys devoted to airline recovery problems.

This paper presents a Large Neighborhood Search (LNS) heuristic for the joint aircraft and passenger recovery problem as described by Palpant et al. [48] in the context of the 2009 ROADEF Challenge. Our method improves the LNS developed by Bisailon et al. [19] in the context of this challenge in which they won first prize. Eight other teams qualified for the final and their approaches are described on the web site <http://challenge.roadef.org/2009>. Among the competing teams, three other approaches provided best solutions for at least two of the final instances. Jozefowicz et al. [40] developed a three-phase heuristic. The first phase integrates the disruptions to the initial plan and returns a feasible solution by removing flight legs and all disrupted itineraries. In the second phase, the cancelled itineraries from the previous phase are reassigned, when possible, to the existing rotations. Phase three attempts to create new flight legs to accommodate the remaining disrupted passengers using available aircraft. Hanafi et al. [44], the team that finished second in the competition, proposed a two-phase heuristic based on an oscillation strategy and mathematical programming. This heuristic first attempts to find a feasible solution close to the initial schedule by solving a relaxation of the problem. If no feasible solution is obtained because of maintenance constraints, a dynamic programming based algorithm is used to find suitable routes. The second phase, the strategic oscillation, alternates between constructive and destructive heuristics to improve solutions. The constructive phase generates feasible aircraft routes and passenger itineraries and then assigns them simultaneously to aircraft and passengers. The destructive phase deletes routes and cancels the corresponding passenger itineraries and then assigns the cancelled passengers to existing flights. Finally, Peekstok et al. [49] developed a simulated annealing algorithm. Initially, the algorithm accepts infeasibility with respect to airports, aircraft and passengers; however, once airport and aircraft feasibility is achieved, it does not allow the solution to become infeasible again for airports and aircraft. Although this team ranked sixth in the

competition, it obtained two of the best solutions for the 22 instances considered in the final.

Acuna-Agost [6] developed a network pruning algorithm that can be combined with aircraft recovery solution methods. The algorithm reduces the number of decision variables and constraints by identifying incompatible or suboptimal network nodes for each commodity. The algorithm was combined with the three-phase heuristic developed by Jozefowiez et al. [40] to solve the problems introduced in the 2009 ROADEF Challenge.

The contribution of this paper is to introduce several refinements to the LNS algorithm that lead to a much improved performance. The resulting heuristic can quickly provide very good solutions to the problem. We also show that it can be profitable to run the algorithm for a longer period of time in order to accommodate additional passengers.

The remainder of the article is organized as follows. Section 2 provides a description of the joint aircraft and passenger recovery problem. Section 3 briefly describes the LNS heuristic developed by Bisailon et al. [19] and presents our improvements to this heuristic. Computational results are reported in Section 4, followed by the conclusion and discussion of future research directions in Section 5.

3.2 Problem description

The joint passenger and aircraft recovery problem consists of creating new aircraft routes and passenger itineraries so as to generate a feasible schedule during the recovery period and return to normal operations as quickly as possible. The problem can be represented on a time-space network $G = (N, A)$, where each node in the set $N = \{1, \dots, n\}$ represents an airport at a specific time, and each arc in the set $A = \{(i, j); i, j \in N, i \neq j\}$ represents a flight leg or a connection between two flight legs at the same airport. Airports have restrictions on the maximum number of arrivals and departures allowed during each 60-minute period beginning on the hour. The parameter a_{ip} represents the arrival capacity at airport i during period p , whereas b_{ip} represents the departure capacity at airport i during period p .

The set F represents the fleet of aircraft operated by the airline, and each aircraft $f \in F$ is characterized by an identification number, a model and a cabin configuration. All aircraft of a given model have the same turn-round time, transit time, range and set of possible cabin configurations. An aircraft family is a subset of models sharing some common cha-

racteristics. Regular maintenance operations are performed on aircraft which then become unavailable for a certain period of time. Airlines must also respect the maximum number of flight hours left before the next maintenance check. The flight schedule is the set of all flights legs that an airline will operate during a certain period of time. Each aircraft f is assigned a sequence of flight legs called a rotation, which must satisfy continuity and turn-round time constraints. Passenger reservations are defined by a reservation number, the number of passengers, the average price, the itinerary and the nature of the itinerary (inbound or outbound). Let K be the set of all itineraries. Each itinerary $k \in K$ consists of one or more flight legs characterized by a cabin class (first, business or economy).

There exist four possible disruptions which can be classified into three groups : aircraft disruptions, airport disruptions and flight disruptions which can be a flight delay or a flight cancellation. Aircraft disruptions consist of the unavailability of an aircraft for a certain period of time, while airport disruptions correspond to a decrease in the number of arrivals and departures allowed at a given airport and a given period.

Given the initial aircraft routes, passenger reservations and a set of disruptions, the objective is to create new aircraft routes and passenger itineraries during the recovery period so as to minimize the weighted sum of the operating costs, the costs related to the disutility of passengers and the costs for non-compliant location of aircraft at the end of the recovery period. There are two types of operating costs : direct operating costs and delay and cancellation costs. Direct operating costs correspond to the operational cost c_{ijf} of aircraft f on arc (i, j) ; they relate to the cost of operating an aircraft (fuel, maintenance and crew) and the cost of services. In our problem, these costs are incurred only during flight and are considered to be independent of the number of passengers. We also consider two types of delay and cancellation costs. The first is the delay cost r_{ij} of arc (i, j) and the cancellation cost e_{ij} of arc (i, j) ; these are operating costs and relate to the different disbursements that an airline must make when such events happen, such as the cost of meals and drinks, lodging, reimbursement of tickets and financial compensations. The second type is associated with the disutility of passengers (i.e. the inconvenience perceived by passengers), and depends on the itinerary type (intercontinental, continental or domestic) and on the cabin class. We refer to these costs as legal delay and legal cancellation costs, where h_{ij} represents the legal delay cost of arc (i, j) and f_{ij} represents the legal cancellation cost of arc (i, j) . Legal delay costs are linear, whereas legal cancellation costs are constant. Downgrading costs, where g_{ijf} is the downgrading cost of aircraft f on arc (i, j) , are also associated with the disutility of

passengers. They are incurred when the cabin class of the new itinerary's flight is lower than the original itinerary's reference cabin class. These costs depend on the level of downgrading and on the itinerary type. Finally, the cost of non-compliant location of aircraft is a penalty imposed when the number of aircraft of each model and configuration at each airport at the end of the recovery period is different from that of the original schedule. There are three types of penalty costs considered, depending on the level of non-compliance. The penalty cost incurred if a required aircraft cannot be matched with an aircraft of the same family at the end of the recovery period is L_{fam} . The penalty cost incurred if a required model cannot be matched is L_{mod} . Lastly, if a required configuration cannot be matched, the penalty cost incurred is L_{conf} . Only those aircraft that have landed before the end of the recovery period are considered.

Apart from the airport capacity constraints, other operational constraints and functional constraints must be satisfied. The seating capacity constraint ensures that the number of passengers assigned to each cabin does not exceed its capacity, while the maintenance constraints ensure that the right aircraft is located at the assigned airport at the required time, and that the maximum flight hours left before maintenance is respected. Finally, turn-round time and connection time constraints ensure, respectively, that the time between two consecutive flights in a rotation is at least as large as the turn-round time and the time between two flights in an itinerary is at least as large as the 30-minute minimal connection time. Functional constraints relate to passenger re-accommodation and ensure that the destination of the modified itinerary is the same as that of the original itinerary, that the departure time of the first flight of the modified itinerary is at least as late as the time of the first flight of the original itinerary, and that the maximum delay does not exceed 18 hours for domestic and continental flights or 36 hours for intercontinental flights. The maximum delay constraint only applies to passengers who have not yet started their trip.

3.3 Solution method

Our solution method is based on the LNS heuristic developed by Bisailon et al. [19] in the context of the 2009 ROADEF Challenge. In this section, we first briefly describe their LNS heuristic and then the additional steps of our improved heuristic.

3.3.1 The large neighbourhood search heuristic

The large neighbourhood search heuristic of Bisailon et al. [19] alternates between three phases : construction, repair and improvement. The first two phases attempt to create a feasible solution, and are repeated until a certain amount of computing time has elapsed or a given number of iterations have been performed. The third phase attempts to improve the solution by considering large schedule changes. If computing time allows it, the full process is reiterated. Figure 3.1 (taken from Bisailon et al. [19]) presents an overview of the solution method.

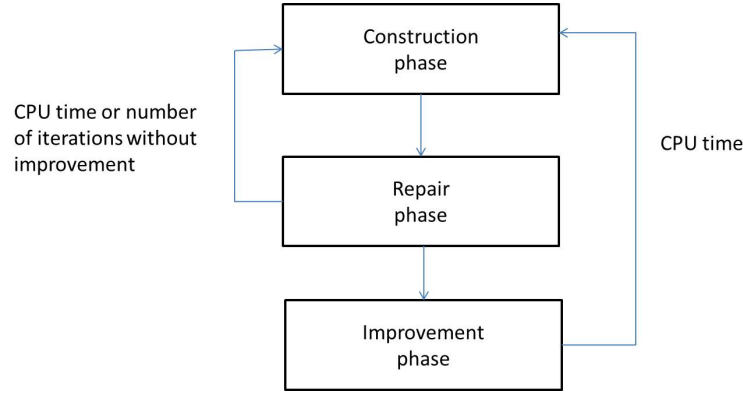


FIGURE 3.1: Overview of the LNS solution method

Construction phase

During the construction phase, the LNS attempts, in four steps, to find feasible solutions by delaying or cancelling flights. The first step considers flights that have been disrupted because of delays on previous flights in the rotation. To reach feasibility, the disrupted flight is delayed by increments of 60 minutes. If feasibility is achieved, the rotation is left unchanged. Otherwise, the flight yielding infeasibility is declared critical. Steps 2 and 4 consider, in order, flights that are cancelled and infeasible rotations due to insufficient capacity, either at the departure airport or at the arrival airport. To reach feasibility, Step 2 attempts to recreate similar flights, while Step 4 attempts to delay the flights by increments of 60 minutes. If feasibility is not reached, the LNS then attempts to remove a loop containing the disrupted flight, or to cancel flights starting from the disrupted flight to the end of the rotation. Step 3 considers infeasible rotations due to maintenance constraints. In order to reach feasibility, the LNS will either remove a loop scheduled before the required maintenance, or cancel all flights starting from the critical flight to the end of the rotation. Algorithm 1 presents

the pseudo-code summarizing the main steps of the LNS construction phase of Bisailon et al. [19].

Repair phase

The repair phase consists of three steps. First, using a greedy approach it identifies flights that still violate the airport capacity constraints after the construction phase and attempts to make them feasible by delaying them to a less congested time period. If this proves impossible, the LNS attempts to remove the smallest loop containing the flight. Again, if this is impossible, it will remove all flights starting from the critical flight to the end of the rotation. In the second phase, the LNS attempts to reinsert the cancelled sequences of the construction phase using available aircraft. Finally, the LNS attempts to accommodate passengers on cancelled itineraries by repeatedly solving a shortest path problem. The passengers are considered for re-accommodation in the order provided in the file. This problem is defined on a graph G_k , for a given itinerary $k \in K$, obtained from G . The graph G_k contains only those flight arcs that have an excess capacity and that could belong to a path from the origin of itinerary i to its destination within the relevant time horizon.

Improvement phase

The improvement phase attempts to improve the solution by local search. In this phase, the LNS attempts to delay flights, one at a time, by a certain amount of time Δ , in order to accommodate new passengers. Again, passengers are reassigned by solving a shortest path problem and whenever a solution yields an improvement, it replaces the current solution. When all flights have been considered, the value of Δ is increased and the process is reinitiated from the first flight. The value of Δ is repeatedly increased until it reaches a maximum value or until a certain amount of time has elapsed.

Algorithm 1 Construction phase

```
1 : randomly sort all aircraft in the set  $F$ 
2 : for all aircraft  $f \in F$  do
3 :   for all flights  $l$  that have become infeasible because of delays on previous flights do
4 :     set  $t = 60$ 
5 :     while delaying flight  $l$  by  $t$  minutes does not resolve the infeasibility and  $t \leq 960$  do
6 :       set  $t = t + 60$  and try delaying flight  $l$  by  $t$  minutes
7 :     end while
8 :   end for
9 :   for all flights  $l$  that have been cancelled do
10 :    if creating a flight similar to  $l$  makes the rotation feasible then
11 :      create flight
12 :    else if removing a loop containing flight  $l$  makes the rotation feasible then
13 :      remove loop
14 :    else
15 :      cancel flights from  $l$  to the end of the rotation
16 :    end if
17 :  end for
18 :  for all flights  $l$  causing a violation of the maintenance constraints do
19 :    if removing a loop makes the rotation feasible then
20 :      remove loop
21 :    else
22 :      cancel flights from  $l$  to the end of the rotation
23 :    end if
24 :  end for
25 :  for all flight  $l$  that cannot take place because of insufficient airport capacity do
26 :    set  $t = 60$ 
27 :    while delaying flight  $l$  by  $t$  minutes does not resolve the infeasibility and  $t \leq 960$  do
28 :      set  $t = t + 60$  and try delaying flight  $l$  by  $t$  minutes
29 :    end while
30 :    if rotation is still infeasible then
31 :      if removing a loop makes the rotation feasible then
32 :        remove loop
33 :      else if the origin of flight  $l$  is the desired final destination of aircraft  $f$  then
34 :        cancel flights from  $l$  to the end of the rotation
35 :      end if
36 :    end if
37 :  end for
38 : end for
```

3.3.2 The improved large neighbourhood search heuristic

The heuristic consists of the same three phases of the LNS algorithm, but additional steps have been introduced in each of them. Figure 3.2 presents an overview of the additional steps in each phase.

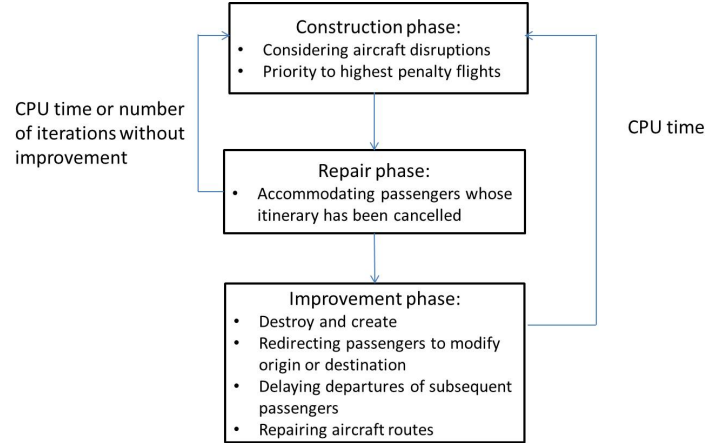


FIGURE 3.2: Overview of the additional steps in the improved LNS

Construction phase

When flights are cancelled, the itineraries of the passengers booked on these flights are cancelled and cancellation costs are incurred. Recreating these flights would eliminate or at least reduce the number of cancelled itineraries. Therefore, we have added a fifth step in the construction phase which considers cancelled flights due to aircraft disruptions. To ensure feasibility when an aircraft becomes unavailable for a period of time, the flights scheduled during this period are cancelled. The heuristic attempts to use available aircraft to recreate these flights by means of a longest path algorithm. In this step we create a network for each available aircraft, i.e. those that have a sufficient time gap between two consecutive flights. The source node represents the airport where and when the aircraft becomes available, and the sink node represents the destination airport of the available aircraft. The network contains the flight arcs that were previously cancelled, connection arcs between all the recreated flight arcs, as well as sink/source arcs between the source and sink nodes and all of the flight arcs and connection arcs. Only the recreated flight arcs have an associated cost and the connection arcs and source/sink arcs created must satisfy continuity, turn-round time and airport capacity constraints. A longest path algorithm is applied to each network and all the possible paths are kept in a list. We then choose, using a greedy approach, the

group of paths that improve the solution cost the most. Some flights may be created more than once if the new aircraft capacities are lower than the original aircraft capacity associated with the flight (e.g., if the initial aircraft associated with the cancelled flight had a capacity of 400 passengers, it is possible that two smaller aircraft, say of 220 passengers each, will both include this flight in their rotation).

Figure 3.3 depicts an example with five flight arcs (A1, A2, A3, A4, A5), three connection arcs (C1, C2, C3) satisfying the turn-round time and airport capacity constraints, as well as source/sink arcs (D1, D2) and (E1, E2, E3) connecting, respectively, the source and sink nodes to the flight and connection arcs. The source/sink arcs must also satisfy the turn-round time and airport capacity constraints. In this example, there are two possible paths. The first contains two source/sink arcs, one flight arc and one connection arc. The second path, in bold, contains two source/sink arcs, one connection arc and two flight arcs. The longest path is the latter with a cost of two and would be kept in the list of possible paths. We then choose, using a greedy approach, the paths that improve solution cost the most. Treating this as a multi-commodity flow problem could have provided better solutions, but since the aircraft have different transit times that vary between 20 and 90 minutes, creating a network using a transit time inferior to the maximum transit time could yield infeasible solutions. Similarly, creating a network using the maximum transit time, we would ensure feasibility, but this would unduly restrict the potential paths. Algorithm 2 presents a pseudo-code summarizing the additional steps of the construction phase. These steps replace steps 9 through 17 of the LNS construction phase described in Algorithm 1.

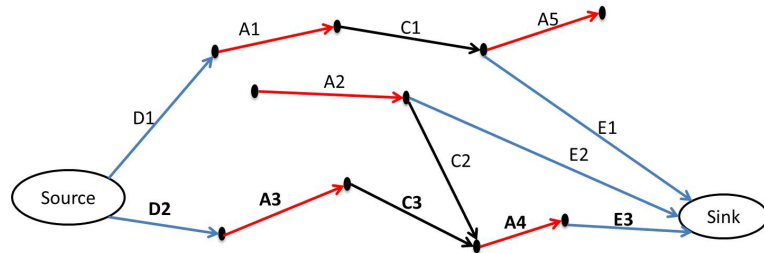


FIGURE 3.3: Example of a longest path graph

Some instances encounter temporary reductions in departures and arrivals at airports. In some cases, this causes a very large number of flights and passenger itineraries to be cancelled. When the number of allowed departures and arrivals is significantly lower than that of the initial schedule, it is important to ensure that the flights that could cause the highest penalty, if cancelled, obtain the available departure and arrival slots. For this reason, when

Algorithm 2 Additional steps of the LNS construction phase

```
1 : for all flights  $l$  that have been cancelled do
2 :   if creating a flight similar to  $l$  makes the rotation feasible then
3 :     create flight
4 :   else if removing a loop containing flight  $l$  makes the rotation feasible then
5 :     remove the loop and insert its flights in a disruption list
6 :   else
7 :     cancel flights from  $l$  to the end of the rotation and insert them in the disruption list
8 :   end if
9 : end for
10 : for all flights in the disruption list do
11 :   find a longest path
12 :   if inserting the longest path in the rotation of aircraft  $f$  improves cost then
13 :     insert the path in the list of possible paths.
14 :   end if
15 : end for
16 : sort all paths in the list of possible paths in decreasing order of cost improvement
17 : and choose the first path in which all flight arcs have not yet been chosen
```

we construct the initial rotations we give priority to international flights, then continental flights and finally domestic flights. Also, within these three groups we consider the aircraft in order of size, starting with the largest aircraft. In addition to the high cancellation cost, international flights should be given priority because once cancelled, they require longer periods of availability and, therefore, are much harder to recreate and assign to available aircraft.

Repair phase

The last step of the LNS repair phase attempts to accommodate passengers affected by cancelled itineraries, but there still remain passengers who cannot be accommodated on existing flights. Creating new flights could allow the algorithm to accommodate these passengers. Therefore, after the initial three steps of the LNS repair phase, we create a group of origin-destination (O-D) pairs with all cancelled itineraries and we try to redirect the passengers using the available aircraft. Two different methods are used depending on the size of the instance : a multi-commodity flow problem with additional constraints and a shortest path problem.

3.2.2.1 Multi-commodity flow problem with additional constraints

We first consider all aircraft at the same time and we solve a multi-commodity flow problem with additional constraints. The network contains the origin-destination arcs, connecting arcs between these arcs and the departure and arrival airports of the available aircraft as well as connecting arcs between the different origin-destination arcs. To minimize the size of the network, the transit time used to create these arcs is the largest transit time of all aircraft. Our model is represented on a time space network $G = (N, A)$, where each node in the set $N = \{1, \dots, n\}$ represents an airport at a specific time and each arc in the set $A = \{(i, j); i, j \in N, i \neq j\}$ represents a flight leg or a connection between two flight legs at the same airport. The recovery period is divided into a set P of 60-minute periods. To take into consideration the maximum number of departures and arrivals allowed at each airport during each period p , we define the set $I_i(p)$ of all arcs (i, j) departing from node i during period p and the set $O_j(p)$ of all arcs (i, j) arriving at node j during period p . We define a_{ip} and b_{jp} , respectively, as the arrival and departure capacity at airport i in period p . For each aircraft f , we define o_f as the origin airport of f and d_f as the destination airport of f . We define c_{ijf} as the sum of the operational cost and the downgrading cost of aircraft f on arc (i, j) , and r_{ij} as the sum of the cancellation, legal cancellation, delay and legal delay cost of arc (i, j) . The binary variables x_{ijf} are equal to 1 if and only if aircraft f uses arc (i, j) . Using this notation, the problem can be formulated as follows.

$$\text{Minimize } \sum_{i,j \in N} \sum_{f \in F} r_{ij} x_{ijf} + \sum_{i,j \in N} \sum_{f \in F} c_{ijf} x_{ijf} \quad (3.1)$$

subject to

$$\sum_{i \in N} x_{ijf} - \sum_{i \in N} x_{jif} = 0 \quad f \in F, j \in N \setminus \{o_f, d_f\} \quad (3.2)$$

$$\sum_{j \in N} x_{o_f j f} = 1 \quad f \in F \quad (3.3)$$

$$\sum_{(i,j) \in I_j(p)} \sum_{f \in F} x_{ijf} \leq a_{jp} \quad p \in P, j \in N \quad (3.4)$$

$$\sum_{(i,j) \in O_i(p)} \sum_{f \in F} x_{ijf} \leq b_{ip} \quad p \in P, i \in N \quad (3.5)$$

$$\sum_{f \in F} x_{ijf} \leq 1 \quad i, j \in N. \quad (3.6)$$

The objective is to minimize cancellation, delay, downgrading and operational costs. Constraints (3.2) and (3.3) are the flow conservation constraints, while constraints (3.4) and (3.5) are the airport capacity constraints, which limit the number of arrivals and departures at every airport during each period. Constraints (3.6) are assignment constraints which ensure that only one aircraft will be assigned to each flight leg.

3.2.2.2 Shortest path problem

When the size of the instance is large, it becomes impossible to repeatedly solve the multi-commodity flow problem within a reasonable time limit. We therefore create a network for each available aircraft and apply a shortest path algorithm to solve the problem. To begin, we create flight arcs between the origin and destination of the O-D pairs satisfying the airport capacity constraints. If we were to create these arcs for each aircraft, we could associate the actual cost to them, but for computing time reasons, the O-D pairs and these arcs are only created once and therefore are independent of the aircraft that will be used. Therefore, a cost of 1 is associated with these arcs because direct operating costs are dependent on the aircraft used and delay, whereas downgrading and cancellation costs are dependent on the number of passengers accommodated and consequently on the aircraft capacity. We then create a network for each available aircraft $f \in F$ (i.e. each aircraft whose time interval between two consecutive flights is sufficient to accommodate new flights). The source is the airport node at the beginning of the availability period of aircraft f and the sink is the airport node at the end of the period. We create arcs between the source node and the beginning of all flight arcs that satisfy the turn-round time and airport capacity constraints. We also create arcs between the end of all flight arcs and the sink node that satisfy the same constraints. The cost associated with these arcs are the direct operating costs of aircraft f . We solve the shortest path problem for all aircraft and keep the possible paths in a list. We then calculate the cost of each path and choose, using a greedy approach, the paths that improve the solution cost the most.

As previously mentioned, it is sometimes impossible to solve the multi-commodity flow pro-

blem several times on some medium to large instances because of the time constraint, but because it significantly improves solutions, the multi-commodity flow problem is solved at least once on each instance. The construction and repair phases are repeated several times until a certain time limit is reached or for a preset number of iterations. For the larger instances, the repair phase first solves the shortest path problem. Once the time limit is reached, we repeat once more the construction and repair phases, but we solve the multi-commodity flow problem with additional constraints instead of the shortest path problem.

Improvement phase

The bulk of the improvements made to the algorithm lies in this third phase. Although the heuristic also improves solutions by attempting to delay flights one at a time, in order to accommodate additional passengers, four new steps are performed, three at the beginning of the repair phase and one at the end.

3.2.3.1 Destroy and create

In order to diversify the search, we have added a destroy and create phase that attempts to improve the solution cost by destroying either part of an aircraft rotation or the complete rotation, and by then creating new flights to accommodate passengers whose itineraries have been cancelled. This is a time consuming step and because of the time limit, it is not completely executed when the size of the instance is too large. Two different destruction operations are applied. The first takes one aircraft at a time and cancels all flights departing after the beginning of the recovery period until the end of the recovery period. To create new flights we proceed as in Step 4 of the repair phase. We create a group of origin-destination pairs with all the cancelled itineraries, as well as a network for each available aircraft containing the origin-destination arcs and connecting arcs between the origin-destination arcs and the departure and arrival airports of the available aircraft. Again, as in Step 4 of the repair phase, we solve this problem by means of a shortest path algorithm. If the solution produced by the destruction operator improves the current solution or surpasses a certain threshold, we insert it in a list of good destruction operators. This phase is applied twice, first by using the initial solution for each aircraft and then by updating the solution if it improves the cost, and using the best solution for each aircraft destruction. We then cancel all rotations belonging to the list of good destruction operators and we solve a multi-commodity flow problem with additional constraints as in the repair phase. In most cases, solving this multi-commodity flow problem yields the best results, but for some instances, the best results are obtained by solving the shortest path while updating the solution. As previously mentioned, when the

number of aircraft and the recovery period are too large, it is impossible to fully execute this step. We therefore impose a time limit. Algorithm 3 presents a pseudo-code of the destroy and create step of the improvement phase.

The second destruction operator also takes one aircraft at a time, but cancels a small loop instead of the whole rotation. Again, the first flight of the loop must depart after the beginning of the recovery period. We have tested a few variants for the partial destruction of rotations :

- For all aircraft rotations we destroy the smallest loop starting from the first flight departing after the beginning of the recovery period, the destination of the last flight of the loop being the same as the origin of the first flight.
- We proceed as above, but several times for each aircraft rotation and starting with different points in time (not always with the first flight after the beginning of the recovery period).
- For those instances where the number of cancelled itineraries is high, we only consider flights having the same origin and destination as some of the passengers whose itineraries have been cancelled and destroying the smallest loop containing that flight.

The create phase is similar to that of the first type of destruction, but only the shortest path problem is solved. Although for some instances the third variant provided good solutions, the second variant provided the best overall results.

Algorithm 3 Improvement phase - destroy and create (first type of destruction)

```
1 : for all aircraft  $f \in F$  do
2 :   cancel all flights starting after the beginning of the recovery period to the end of the rotation
3 :   for all cancelled itineraries  $i \in I$  do
4 :     create origin-destination pairs
5 :     for all origin-destination pairs do
6 :       for all aircraft  $f \in F$  do
7 :         solve a shortest path problem
8 :         if the shortest path improves solution cost then
9 :           insert the shortest path in the rotation of aircraft  $f$ 
10 :        end if
11 :      end for
12 :    end for
13 :  end for
14 :  if destruction improves solution cost then
15 :    replace best solution and insert the destruction in the list of good destructions
16 :  else if solution cost exceeds a certain threshold then
17 :    revert to best solution and insert the destruction in the list of good destructions
18 :  else
19 :    revert to best solution
20 : end for
21 : for all aircraft  $f$  in the list of good destructions do
22 :   cancel all flights starting after the beginning of the recovery period to the end of the rotation
23 : end for
24 : for all cancelled itineraries  $i \in I$  do
25 :   create origin-destination pairs
26 : end for
27 : for all origin-destination pairs do
28 :   for all aircraft  $f \in F$  do
29 :     create and solve the multi-commodity network flow problem
30 :   end for
31 : end for
```

3.2.3.2 Redirecting disrupted passengers to different airports to create new O-D pairs

Some instances contain temporary reductions in airport capacity. It is therefore difficult and sometimes impossible to create new flights to accommodate the disrupted passengers. Although some airports experience capacity reductions, other airports have available departure and arrival slots that could be used to accommodate the passengers affected by cancelled itineraries. This step attempts to accommodate additional passengers by redirecting them to a different airport. The passengers are redirected using flights with excess capacity that have either the same origin or the same destination. Again, we create a group of origin-destination pairs for the disrupted itineraries and we try to redirect them one at a time. If the disrupted airport is the origin airport, we examine all existing flights with excess capacity leaving this airport after the departure time of the O-D pair. The flights are treated in non-decreasing order of departure time and all flights with excess capacity are sorted in non-increasing order of capacity. Therefore, if two flights have the same excess capacity, the earlier flight will be given priority. We accommodate the passengers of the disrupted itineraries on one or more of these flights, regardless of their destination. We then create a new O-D pair originating at the arrival airport of the new flight assigned to the disrupted itinerary and ending at the original destination airport and we attempt to assign an available aircraft to this new O-D pair using a shortest path algorithm. To illustrate this, in Figure 3.4 we consider a group of 20 passengers whose origin is NCE and destination is CDG. There is an existing flight with an excess capacity of 50 departing from NCE after the 20 passengers' initial departure time and arriving at AVN. If it is possible to use an available aircraft at a given airport to create a flight between its current position X and AVN, a flight between AVN and CDG, and a final flight between CDG and its current position X, and if this improves the solution cost, these flights will be created and the 20 passengers will be accommodated on flight NCE-AVN and on flight AVN-CDG.

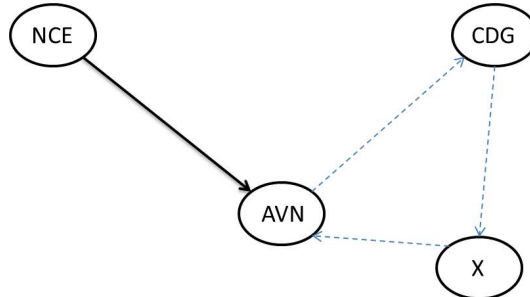


FIGURE 3.4: Redirecting passengers

When the disrupted airport is the destination airport, we proceed in the opposite way, i.e. we make an ordered list of all flights with excess capacity arriving at the destination airport before the maximum allowed delay, and we accommodate the passengers of the disrupted itineraries on one or more of the flights. We create a new O-D pair ending at the departure airport of the new flight and we attempt to assign an available aircraft to this new O-D pair.

3.2.3.3 Delaying subsequent passenger departures

This additional step is used for instances containing disruptions in airport capacities. Because of the maximum delay constraint, we cannot accommodate disrupted passengers on just any flight. We therefore attempt to accommodate additional passengers by delaying other passengers having the same origin and destination, but a later departure time, which means that it is possible to accommodate them on a later flight that would have violated the maximum delay constraint for the disrupted passengers. We again create origin-destination pairs for the disrupted itineraries. We then verify whether there exist flights with the same origin and destination arriving at the destination before the maximum allowed delay, even if their excess capacity is zero. If the passengers assigned to these flights have only one leg, we insert them in a list. We then verify whether there exist flights with excess capacity leaving later. If so, the passengers in the list are assigned to some of these flights and the original disrupted passengers will be assigned to the first flight. For example, in Figure 3.5 there are 20 passengers with a cancelled itinerary from CDG to NCE originally leaving at 10h. Flight 1, with origin CDG and destination NCE has no excess capacity and is leaving at 19h. Flight 2 (same origin and destination) has excess capacity of 50 and is leaving at 8h the following day. Because assigning the 20 passengers whose itineraries have been cancelled to flight 2 would violate the maximum delay constraint, we cannot do so. However, some passengers on flight 1 could be assigned to flight 2 without violating the maximum delay constraint and the 20 passengers whose itineraries have been cancelled could be assigned to flight 1 without violating the maximum delay constraint. We therefore assign these 20 passengers to flight 1 and assign 20 passengers from flight 1 to flight 2. Because the cost of cancellation is much larger than the cost of delays, this should always improve the solution cost. It is important to transfer only passengers whose final destination is NCE, otherwise the delay could cause the passengers to miss their connecting flight and their itinerary could be cancelled.

It is possible that airlines may not wish or may be unable to delay passengers whose itineraries have not been disrupted. It is important to note that this step is only applied to

instances with disruptions in airport capacities and that removing it from the algorithm would not have a very large impact on solution quality.

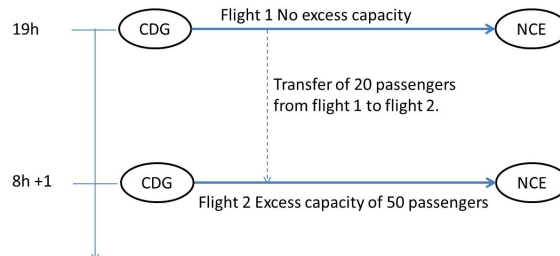


FIGURE 3.5: Example delaying subsequent passenger departures

3.2.3.4 Repairing aircraft positions

In order to avoid penalties for non-compliant location of aircraft at the end of the recovery period, we attempt to transfer all aircraft to the required airport, when possible. Although this step only yields small improvements on large instances, it can produce rather significant improvements on small ones.

3.4 Computational results

The LNS heuristic just described was implemented and tested on an Intel Core2 Quad Q9550 CPU with 4GB of memory. We have used the instances of the 2009 ROADEF Challenge to test our algorithm with a maximum computing time of 10 minutes. Recall that our heuristic is based on the LNS developed by BISAILLON et al. [19] for this competition, in which these authors ranked first. The parameters of this algorithm were not modified, but the additional steps of the heuristic increase the computation times of all phases. Therefore, in order to ensure that all instances go through the three phases at least once, we impose a time limit for each phase.

Table 3.1 provides the final score and the ranking of the nine teams in the ROADEF Challenge. During the competition, only BISAILLON et al. obtained results for the X01, X02, X03 and X04 instances which were not considered in the evaluation of the methods. These instances were later solved by JOZEFOWIEZ et al. [40] and we will therefore also report results for them. We present in Tables 3.2, 3.3 and 3.4 the characteristics of the B and X instances,

i.e. the recovery period, the number of flights, the number of aircraft, the number of airports, the number of flight disruptions, the number of airport disruptions and the number of aircraft disruptions.

We have calculated the marginal impact on the cost of each improvement procedure just described, by switching each of them off. In the construction phase, considering aircraft disruption and giving priority to the highest penalty flights has an average impact of 9.11% for the B instances and 13.97% for the X instances. The improvements made in the repair phase have an average impact of 2.18% for the B instances and 22.48% for the X instances. The improvement with the largest marginal impact on solution cost is the destroy and create step of the improvement phase, with an average of 11.66% for the B instances and 29.70% for the X instances, whereas the average impact on the solution cost of the redirecting passenger step is 0.18% with a significant impact on only three instances. Delaying departures of subsequent passengers has an average impact of 0.28%, and this impact is significant on only two instances (2.70% for instance XB02 and 1.94% for instance X02). Finally, the average impact on the solution cost of repairing aircraft routes is 0.01% for the B instances and 7.02% for the X instances.

TABLE 3.1: Final ranks and scores in the 2009 ROADEF Challenge

Team	Rank	Score
Bisaillon et al.	1	95.90
Hanafi et al.	2	92.73
Acuna-Agosta et al.	3	74.26
Eggermont et al.	4	72.01
Darlay et al.	5	70.62
Peekstok et al.	6	70.31
Jozefowicz et al.	7	64.02
Dickson et al.	8	42.02
Eggenberg et al.	9	20.48

TABLE 3.2: Characteristics of the B instances

	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10
Recovery period	36	36	36	36	36	36	36	36	36	36
Nb. aircraft	255	256	256	256	256	256	256	256	256	256
Nb. airports	45	45	45	45	45	45	45	45	45	45
Nb. flights	1,423	1,423	1,423	1,423	1,423	1,423	1,423	1,423	1,423	1,423
Nb. itin.	11,214	11,214	11,214	11,214	11,214	11,214	11,214	11,214	11,214	11,214
Flight disr.	230	255	229	230	0	230	255	229	230	77
Aircraft disr.	0	0	1	0	0	0	0	1	0	0
Airport disr.	0	0	0	1	34	0	0	0	1	34

TABLE 3.3: Characteristics of the XA, XB instances

	XA01	XA02	XA03	XA04	XB01	XB02	XB03	XB04
Recovery period	14	52	14	52	36	52	36	52
Nb. aircraft	85	85	85	85	256	256	256	256
Nb. airports	35	35	35	35	45	44	45	44
Nb. flights	608	608	608	608	1,423	1,423	1,423	1,423
Nb. itin.	1,943	3,959	1,872	3,773	11,214	11,214	11,565	11,565
Flight disr.	83	0	83	0	229	0	228	0
Aircraft disr.	3	3	3	3	3	1	4	4
Airport disr.	0	407	0	407	0	34	0	34

Tables 3.5 and 3.6 present the solution costs (in euros) of our algorithm for the B instances, whereas Tables 3.7, 3.8 and 3.9 present the solution costs for the X instances. We also show the solution costs for all teams who obtained at least one best solution for the B and the X instances, i.e. Bisailon et al., Hanafi et al., Jozefowiez et al. and Peekstok et al. The results presented for Jozefowiez et al. are not the results obtained during the competition, but the improved results presented in Jozefowiez et al. [40] using an Intel Core 2 Duo E6550 2.33Ghz CPU. Only Bisailon et al. and Jozefowiez et al. were able to obtain solutions for all 22 instances, and several teams were not able to find solutions for all 18 instances considered in the evaluation of the methods. We present the solution cost obtained when running the heuristic for 10 minutes, which was the maximum time allowed during the competition and therefore also the CPU time limit used by the other teams reported in our tables. To take the relative speed of different computers into account, we have considered the scores made available by CPU benchmarks (2013) [58]. Our computer had a score of 1200 whereas the computer used during the challenge had a score of 616, and the computer used by Jozefowiez et al. [40] had a score of 881. We therefore also present the solution cost obtained when running the heuristic for just 5 minutes. Tables 3.5–3.9 also show the gap between the best

solution and the solutions obtained with our heuristic, and report the number of passengers whose itinerary has been cancelled in our final solution.

Table 3.10 presents the average gap between the solution provided by our heuristic within 5 and 10 minutes of computing time and that of the other teams for the B instances, the X instances and all instances combined. Table 3.11 presents the number of feasible solutions, the number of best solutions and the average gap between the other teams' solution cost and ours. Because not all teams considered were able to solve all instances, the average gap was measured only on the instances they were able to solve. Our solution method provided 21 out of 22 best solution costs and proves to be stable and provide good solutions for all the instances within 10 minutes of computing time. Our solution method also provided 17 out of 22 best solution costs within 5 minutes of computing time.

Tables 3.12 through 3.16 report the best known solution for the instances including all solutions obtained by the nine teams competing in the Challenge and the results reported in Acuna-Agost [6]. This team did not report the results obtained within 10 minutes of computing time, but reported for each instance the computing time necessary to obtain their best solution (varying between 602.73 and 1073.80 seconds). These tables also report our solution costs for different execution times of our algorithm : 20, 30 and 60 minutes. All best solutions are obtained either by our algorithm or by the method proposed by Acuna-Agost.

Table 3.17 presents the number of best solutions and the average gap between our solution cost for different execution times and the solution cost reported by Acuna-Agost [6]. Our solution method, with an execution time of 20 minutes, provided 14 best solution costs out of the 22 instances, while Acuna-Agost obtained 8 best solution costs. Also, within the 20

TABLE 3.4: Characteristics of the X instances

	X01	X02	X03	X04
Recovery period	78	78	78	78
Nb. aircraft	618	618	618	618
Nb. airports	168	168	168	168
Nb. flights	2178	2178	2178	2178
Nb. itin.	28,308	28,308	29,151	29,151
Flight disr.	0	0	0	0
Aircraft disr.	1	1	1	1
Airport disr.	1	0	1	0

TABLE 3.5: Recovery costs for the B instances (part 1)

Team	B01	B02	B03	B04	B05
Bisaillon et al.	983731.75	1522452.75	1031825.30	1192519.20	15639190.80
Hanafi et al.	5813896.95	9950888.70	5569623.95	5775277.70	13139974.30
Jozefowiec et al.	971182.50	1220708.30	1007565.70	1101394.80	25302036.95
Peekstok et al.	1590791.95	2482349.85	1650348.50	1667929.00	9653780.05
LNS 5 min.	847943.85	1260552.35	887133.5	975590.00	10081015.50
% wrt best	12.69%	-3.16%	11.95%	11.42%	-4.42%
Nb passengers	26	63	55	39	4119
LNS 10 min.	843983.35	1225858.10	859809.05	968796.25	8717434.65
% wrt best	13.09%	-0.42%	14.66%	12.04%	9.70%
Nb passengers	23	60	48	35	2953

TABLE 3.6: Recovery costs for the B instances (part 2)

Team	B06	B07	B08	B09	B10
Bisaillon et al.	3789254.05	5488693.00	4069557.35	5906239.15	52355192.80
Hanafi et al.	9095248.10	19144460.30	10099607.00	10176173.55	34523605.00
Jozefowiec et al.	3218000.10	5039744.20	3509318.00	3967344.70	59289841.80
Peekstok et al.	5993131.95	8580429.2	6234247.00	5465108.55	38537692.15
LNS 5 min.	3019252.55	4965177.55	3103625.30	2974577.25	33566143.3
% wrt best	6.18%	1.48%	11.56%	25.02%	2.77%
Nb passengers	1075	1619	1053	860	27549
LNS 10 min.	2733781.50	4588857.70	3067543.75	2678741.60	32171455.95
% wrt best	15.05%	8.95%	12.59%	32.48%	6.81%
Nb passengers	942	1438	976	523	25757

minutes of computing time, our algorithm provided on average solution costs 13.64% lower than the solution costs obtained by Acuna-Agost. The number of best solutions provided by our algorithm with an execution time of 30 minutes increased to 16 with an average solution cost 19.18% lower than the solution costs obtained by Acuna-Agost. Finally, with an execution time of 60 minutes, our algorithm provided 17 of the best solution costs with an average solution cost 25.04% lower than the solution costs obtained by Acuna-Agost. Running the algorithm for a longer period of time proves to be profitable, especially for the X instances where the number of aircraft and airport disruptions is higher.

TABLE 3.7: Recovery costs for the XA instances

Team	XA01	XA02	XA03	XA04
Bisaillon et al.	462571.10	2238311.75	959080.90	5480962.75
Hanafi et al.	116195.20	1475322.10	285287.05	4112262.60
Jozefowicz et al.	150857.60	4787273.45	404964.20	9352557.15
Peekstok et al.	145591.00	2614075.45	INF	INF
LNS 5 min.	103140.90	1465251.25	262945.25	4429120.5
% wrt best	11.23%	0.68%	7.83%	-7.15%
LNS 10 min.	103140.90	1462311.50	262945.25	3972419.50
% wrt best	11.23%	0.88%	7.83%	3.40%
Nb passengers	1	158	24	2156

TABLE 3.8: Recovery costs for the XB instances

Team	XB01	XB02	XB03	XB04
Bisaillon et al.	1352823.05	17064421.50	6463354.30	53543381.45
Hanafi et al.	5985772.05	12716512.00	11124244.55	34331225.80
Jozefowicz et al.	1194006.65	24885515.20	4251062.90	57588009.55
Peekstok et al.	INF	11297822.20	INF	INF
LNS 5 min.	1082138.65	9917850.35	4660820.30	34743666.00
% wrt best	9.37%	12.21%	-8.79%	-1.18%
Nb passengers	98	5339	1933	28870
LNS 10 min.	1028223.70	9986653.95	4204317.95	33326556.40
% wrt best	13.88%	11.61%	1.10%	2.93%
Nb passengers	75	5432	1672	27311

TABLE 3.9: Recovery costs for the X instances

Team	X01	X02	X03	X04
Bisaillon et al.	1116142.85	806011.20	2682125.00	485904.75
Hanafi et al.	INF	INF	INF	INF
Jozefowicz et al.	283033.85	135872.00	1835571.95	590774.35
Peekstok et al.	INF	INF	INF	INF
LNS 5 min.	226289.75	35787.50	1472732.90	244507.00
% wrt best	20.05%	73.66%	19.77%	49.68%
Nb passengers	0	0	214	79
LNS 10 min.	222040.50	-18917.75.00	1316835.90	240656.75
% wrt best	21.55%	113.92%	28.26%	50.47%
Nb passengers	0	0	39	79

TABLE 3.10: Average gap with best solution with 5 minutes or 10 minutes of computing time

Instances	Average gap 5 min.	Average gap 10 min.
B instances	7.55%	12.49%
X instances	16.01%	23.26%
Total	11.95%	17.82%

TABLE 3.11: Characteristics of the solution costs

Teams	B			X			Total		
	#solved	#best	avg. gap	#solved	#best	avg. gap	#solved	#best	avg. gap
LNS 10 min.	10	9		12	12		22	21	
Bisaillon et al.	10	0	21.97%	12	0	52.87%	22	0	34.24%
Hanafi et al.	10	0	67.07%	8	0	24.10%	18	0	47.97%
Jozefowicz et al.	10	1	27.40%	12	0	44.47%	22	0	36.71%
Peekstok et al.	10	0	41.62%	3	0	28.27%	13	0	38.55%

TABLE 3.12: Recovery costs for the B instances (part 1)

Team	B01	B02	B03	B04	B05
Best known solution	797903.00	1020906.00	831642.00	907752.00	9653780.05
LNS 20 min.	813164.60	1203925.60	834790.55	957761.25	7871090.35
% wrt best	-1.88%	-15.20%	-0.38%	-5.51%	18.47%
LNS 30 min.	813164.60	1203925.60	834790.35	944419.95	7593532.40
% wrt best	-1.88%	-15.20%	-0.38%	-3.88%	21.34%
LNS 60 min.	813164.60	1203925.60	834790.55	898181.20	7536525.35
% wrt best	-1.88%	-15.20%	-0.38%	1.05%	21.93%

TABLE 3.13: Recovery costs for the B instances (part 2)

Team	B06	B07	B08	B09	B10
Best known solution	2671627.00	4184662.00	3038695.00	3484271.00	34523605.00
LNS 20 min.	2677841.15	4470070.90	3053223.75	2597649.55	31986001.90
% wrt best	-0.02%	-6.82%	-0.48%	25.45%	7.35%
LNS 30 min.	2575495.25	4383665.75	3017226.40	2558698.65	30915288.90
% wrt best	3.60%	-4.54%	0.71%	26.56%	10.45%
LNS 60 min.	2541716.95	4349898.75	2935127.10	2558698.65	30733381.45
% wrt best	4.86%	-3.80%	3.41%	26.56%	10.98%

TABLE 3.14: Recovery costs for the XA instances

Team	XA01	XA02	XA03	XA04
Best known solution	116195.2	1475322.10	285287.05	4112262.60
LNS 20 min.	103140.90	1459642.75	262945.25	3641047.75
% wrt best	11.23%	1.06%	7.83%	11.46%
LNS 30 min.	103140.90	1459642.75	262945.25	3620952.75
% wrt best	11.23%	1.06%	7.83%	11.95%
LNS 60 min.	99009.20	1457637.75	262945.25	3620952.75
% wrt best	14.79%	1.20%	7.83%	11.95%

TABLE 3.15: Recovery costs for the XB instances

Team	XB01	XB02	XB03	XB04
Best known solution	1010114.00	11297822.20	3878297.00	34331225.80
LNS 20 min.	999316.70	7810804.35	4199921.65	33227253.90
% wrt best	1.07%	30.86%	−7.66%	3.21%
LNS 30 min.	999316.70	7495019.65	4036663.10	32977316.70
% wrt best	1.07%	33.66%	−3.92%	3.94%
LNS 60 min.	999316.70	7147781.6	3910619.25	32510131.20
% wrt best	1.07%	36.73%	−0.83%	5.60%

TABLE 3.16: Recovery costs for the X instances

Team	X01	X02	X03	X04
Best known solution	248154.00	−138279.00	1748160.00	485904.75
LNS 20 min.	57583.00	−74180.25	1308414.75	144504.25
% wrt best	37.88%	−46.35%	25.15%	70.26%
LNS 30 min.	−127921.50	−98205.50	1253553.05	133929.00
% wrt best	151.55%	−28.98%	28.29%	72.44%
LNS 60 min.	−182409.75	−206073.25	1212619.30	103980.75
% wrt best	173.51%	32.90%	30.63%	78.60%

TABLE 3.17: Characteristics of the solution costs

	B		X		Total	
Teams	#best	avg. gap	#best	avg. gap	#best	avg. gap
LNS 20 min.	3		11		14	
Acuna-Agost	7	4.45%	1	21.31%	8	13.64%
LNS 30 min.	5		11		16	
Acuna-Agost	5	6.17%	1	30.02%	6	19.18%
LNS 60 min	6		11		17	
Acuna-Agost	4	7.22%	1	39.88%	5	25.04%

3.5 Conclusions

We have presented several improvements to a large neighbourhood search heuristic for the integrated aircraft and passenger recovery problem. We have shown that the modifications significantly improve the solution costs for all instances considered in the 2009 ROADEF Challenge. On the 22 instances, our algorithm yielded very good solutions, finding 17 best solution costs within a 5-minute computing time limit and 21 best solution costs within a 10-minute computing time limit. We also showed that for instances with a high number of cancelled itineraries, it can be profitable to run the algorithm for longer than 10 minutes. Future research will consider additional constraints to better represent the reality of the problem faced by airlines, as well as an exact method to compute lower bounds. Also, the instances we used to test the heuristic consider cancellation costs significantly greater than the delay costs, which leads to the cancellation costs shadowing the costs associated with the disutility of passengers in regards to delays. A better understanding of the disutility of the passengers and its related costs could lead to a more accurate objective function.

Acknowledgments

This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 227837-09 and 39682-10. This support is gratefully acknowledged. Thanks are also due to Serge Bisailon for his help with the computer implementation of the algorithm, and to three anonymous reviewers for their valuable comments.

Chapitre 4

A Column Generation Post-Optimization Heuristic for the Integrated Aircraft and Passenger Recovery Problem

Information sur le chapitre

Un article basé sur ce chapitre a été publié dans la revue *Computers & Operations Research* : K. Sinclair, J.-F. Cordeau, G. Laporte. A Column Generation Post-Optimization Heuristic for the Integrated Aircraft and Passenger Recovery Problem. *Computers & Operations Research* 65 : 42-52, 2016.

Abstract

The use of hub-and-spoke networks by most major commercial airlines means that small disruptions can have a significant impact on their operational costs. These disruptions, such as delayed or cancelled flights, reduction in arrival and departure capacity, and unavailable crew or aircraft, occur frequently and when they do, airlines must recover their operations as quickly as possible. In this paper we model the joint aircraft and passenger recovery problem as a mixed integer program and we present a column generation post-optimisation heuristic to solve it. We also show how the model and the heuristic can be modified to consider passenger recovery only. The resulting heuristic improves the best known solutions for all instances of the 2009 ROADEF Challenge, within reasonable computing times.

Keywords : airline recovery, fleet assignment, aircraft routing, passenger itineraries, column generation.

4.1 Introduction

Hub-and-spoke networks allow airlines to serve large markets with a limited number of flight legs. Therefore, most commercial airlines use such networks which ensure a very efficient use of critical resources. However, this implies that small disruptions can have a significant network-wide impact on planned operations. These disruptions can be caused by cancelled or delayed flights, unavailable crews or aircraft due to unplanned maintenance, or adverse weather conditions, which can force airport closures or limit the number of arrivals and departures. These disruptions can also have significant impact on the airlines' operational costs. Ball et al. [11] estimated the total cost of US air transportation delays at \$32.9 billion in 2007. When disruptions occur, the airlines must re-establish the planned schedule as quickly as possible, usually by the following day. The recovery period defines the time by which normal operations must resume. During this period, the airlines must plan the recovery operations for the aircraft, the crews and the passengers, and must also ensure that the aircraft and crews are positioned at the correct locations by the end of the recovery period in order to allow the planned schedule to resume.

As for several other tactical planning problems, the size and the complexity of recovery problems imply that they are usually solved in a sequential manner. Since they need to be solved very quickly, usually within a few minutes, exact optimization is impractical. It is therefore common to apply decomposition heuristics in such contexts. The aircraft recovery problem is usually solved first and the crew recovery problem is handled in a second stage. Aircraft recovery operations can include cancelling flight legs, delaying flight legs, aircraft swapping and modifying aircraft rotations (Ball et al. [13]). The objective of the aircraft recovery problem is to determine new aircraft rotations, while minimizing cancellation and delay costs and satisfying the maintenance constraints, the arrival and departure constraints, and the flow and locations constraints.

There exists a rich literature on the aircraft recovery problem. Teodorović et al. [62] developed a network model that minimizes the total delay of passengers and solved the problem to optimality using a branch-and-bound heuristic. However no realistic instances could be solved through this approach. Jarrah et al. [39] used minimum-cost network models, one delay model and one cancellation model, and implemented an algorithm which solves the shortest path problem repeatedly in order to determine the necessary flows. A greedy randomized adaptive search procedure (GRASP) was also developed by Arguello et al. [10]. The

algorithm is composed of a construction phase which arbitrarily selects a solution from a candidate list, examines neighbouring solutions and inserts the best one in the candidate list, followed by a local search phase. All the above authors solve the aircraft recovery problem for a homogeneous fleet. The heterogeneous fleet recovery problem was modeled by Cao et al. [22, 23] as a quadratic programming program which considers delaying and cancelling flight legs. These authors applied an approximate linear programming algorithm proposed by Coleman and Hulburt [27] to solve the problem. Rosenberg et al. [52] modeled the aircraft recovery problem as a set packing problem and used an aircraft selection heuristic to determine a subset of aircraft in order to reduce the size of the integer program. Eggenberg et al. [30] presented a constraint specific recovery network model which they solved by column generation. Dozić et al. [29] developed a heuristic that interchanges parts of rotations and returns a list of good solutions, while Xiuli et al. [64] presented a hybrid heuristic combining GRASP and tabu search.

After solving the aircraft recovery problem, the crew recovery problem can be solved by reassigning a subset of crews, deadheading crew members or using reserve crews. The objective of the crew recovery problem is to create new crew schedules while minimizing costs and the total number of schedule changes. Stojković et al. [61] presented the crew recovery problem as a set partitioning problem which they solved by a column generation method embedded within a branch-and-bound search tree. Lettovsky et al. [41] and Medard et al. [45] both formulated the problem as a set covering problem. The first authors applied a primal-dual subproblem simplex algorithm, while the second authors used depth-first tree search, reduced cost column generation and shortest path algorithms. Abdelganhy et al. [4] presented a mixed integer programming model and developed a rolling horizon approach which solves a sequence of optimization assignment problems. Other algorithms have also been applied to this problem (see, e.g., Nissen et al. [47] and Yu et al. [67]).

Finally, the passenger recovery problem is solved by reassigning those passengers whose itineraries have been cancelled or modified by the disruptions. Zhang et al. [70] developed an integer linear program and discussed two schemes. In the first, flight legs are cancelled and passengers are transported by surface mode. In the second, alternative hubs are selected and ground transportation is used between the initial and the alternate hub. Bratu et al. [21] used network flow techniques to solve the passenger recovery problem.

Solving the recovery problem in a sequential way typically leads to suboptimal solutions.

Therefore considering the integrated recovery problem can yield substantial cost reductions for airlines. Petersen et al. [50] solved the integrated aircraft, crew and passenger recovery problem by means a Benders decomposition scheme, with the scheduling problem as a master problem, and the aircraft, crew and passengers recovery problems as the subproblems. Zhang et al. [69] modeled the integrated problem as a set partitioning problem which they solved by means of a rolling horizon based algorithm. Other methods have been developed to solve two integrated recovery problems. Thus, for the joint aircraft and crew recovery problem, Luo et al. [43] modeled the problem as an integer linear program and applied a heuristic based on a restricted version of the model to solve it. Stojković et al. [61] developed a linear program model for this joint problem, whereas Abdelghany et al. [5] developed a multi-phase heuristic which integrates a simulation model and a resource assignment optimization model. As for the joint aircraft and passenger recovery problem, Zergodi et al. [68] presented an ant colony optimization algorithm that takes into consideration passenger delay and cancellation costs in the objective function, while Jafari et al. [36, 37] presented a mixed integer programming model in which the variables represented aircraft rotations and passenger itineraries instead of flight legs. A detailed survey of the recovery problems can be found in Clausen et al. [26].

This paper presents a post-optimization heuristic for the joint aircraft and passenger recovery problem as defined by Palpant et al. [48] for the 2009 ROADEF Challenge. Nine teams took part in the final of this competition. The winning team, Bisailon et al. [19], made use of a large neighbourhood search heuristic. The algorithms proposed by the remaining teams can be found on the web site <http://challenge.roadef.org/2009>. Among these, only three teams were able to find the best solution for at least one instance. Mansi et al. [44], who came second, presented a two-stage method. In the first stage, they attempted to find a feasible solution using mixed integer programming (MIP). If no feasible solution was found, a repair heuristic was applied. The second stage improved the solution by using an oscillation strategy that alternates between a constructive and a destructive phase. Peekstok et al. [49] who ranked sixth, developed a simulated annealing algorithm. Their algorithm accepts aircraft, airport and passenger infeasibilities which are handled by introducing a second term in the objective function. The cost of infeasibility is increased in order to force the algorithm to find a feasible solution. Jozefowicz et al. [40], who finished in seventh position, developed a three-phase heuristic. In the first phase, the disruptions are integrated in the schedule. Flight legs are removed and itineraries are cancelled in order to return a feasible solution. The second phase attempts to reassign disrupted passengers to the existing flight legs and in the final phase, additional flight legs are added to the aircraft rotations in order to reassign

the remaining disrupted passengers.

After the challenge, Acuna Agost [6] presented a post-processing procedure combined with the three-phase heuristic of Jozefowiez et al. [40]. The problem was formulated as an integer programming model based on a minimum cost multi-commodity flow problem. Two algorithms were developed to reduce the number of variables and constraints by identifying incompatible or suboptimal network nodes for each commodity. The solution method was able to greatly improve the solutions obtained by Jozefowiez et al. [40]. Sinclair et al. [56] later presented a large neighbourhood search heuristic (LNS) based on that of Bisailon et al. [19]. Several refinements were introduced in each phase so as to diversify the search. The resulting heuristic, which will be described in Section 4, provided the best solution for 21 of the 22 instances.

The contribution of this paper is to present a column generation post-optimization heuristic which, when applied after the LNS heuristic of Sinclair et al. [56], leads to much improved solution costs within reasonable computing times. We show that the solution costs obtained within the 2009 ROADEF challenge time limit can be greatly improved by slightly increasing the allotted computing time. The problem is formulated as a mixed integer programming model but can be modified, along with the heuristic, so as to only consider passenger recovery.

The remainder of the article is organized as follows. The joint aircraft and passenger recovery problem is described in the following section while Section 3 presents the APRP model. Section 4 summarizes the LNS heuristic developed by Sinclair et al. [56], while Section 5 presents the post-optimization column generation heuristic. Computational results are reported in Section 6, and conclusions follow in Section 7.

4.2 Problem description

We now formally describe the aircraft and passenger recovery problem (APRP) considered in the 2009 ROADEF Challenge. Given a planned schedule which includes passenger itineraries and aircraft routes, and a set of disruptions, the objective of the joint aircraft and passenger recovery problem is to determine new aircraft routes and passenger itineraries in order to provide an alternate feasible plan and to allow the return to the planned schedule by the end of the recovery period. Before presenting the model we introduce some terminology.

4.2.1 Airports

The airports form a node set $N = \{1, \dots, n\}$ where each node represents an airport at a specific time. For each airport $i \in N$, a_{ip} and b_{ip} represent respectively the maximum number of arrivals and the maximum number of aircraft departures in the time interval p , a 60-minute period beginning on the hour.

4.2.2 Aircraft

The aircraft fleet operated by the airline forms a set F . Each aircraft $f \in F$ is defined by an identification number, a family, a model, a maximum number of flight hours left before maintenance, and a cabin configuration which gives the maximum seating capacity in each cabin class. Also, each aircraft is characterized by a minimum turnaround time, a transit time and a flight range. There are three cabin classes : economy, business and first, and cap_{mf} is the capacity of cabin class m on aircraft f . We denote by M the set of all cabin classes. Each aircraft f performs a rotation, which is a sequence of flight legs or connections between two flight legs at the same airport, represented by the arc set $A = \{(i, j) : i, j \in N, i \neq j\}$. We define t_{ij} as the flying time associated with each flight arc (i, j) , and $O_{j(p)}$ and $I_{i(p)}$ as the set of all arcs (i, j) arriving at node j during period p and as the set of all arcs (i, j) departing from node i during period p , respectively. The demand for aircraft f at node i is denoted by a binary coefficient d_{if} . In addition, B_l represents the set of all aircraft l in the set L of aircraft families, T_g represents the set of all aircraft of model g in the set G of aircraft models, and finally, V_h represents the set of all aircraft with configuration h in the set H of aircraft configurations. Figure 4.1 depicts how the set of aircraft F is partitioned into subsets of families B , of models T and of configurations V .

Aircraft rotations must also satisfy continuity constraints, turnaround time constraints, maximum flight hours constraints and maintenance constraints. The latter constraints ensure that the right aircraft is located at the planned maintenance airport at the required time, whereas turnaround time constraints ensure that the time between two consecutive flight legs in a rotation is at least as large as the turnaround time. Finally, the maximum flight hours constraints ensure that the total flying time of aircraft f does not exceed the maximum flight hours left, t_f , before the next required maintenance.

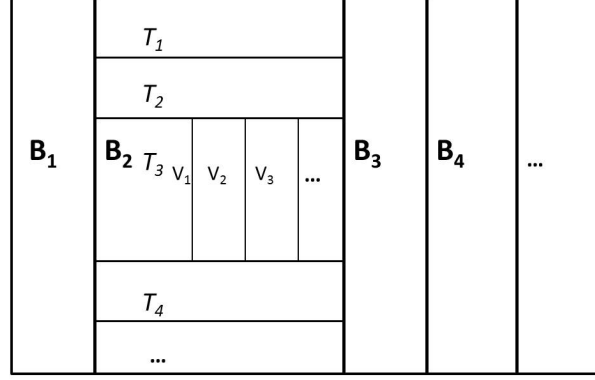


FIGURE 4.1: Set of aircraft F partitioned into families B_l , models T_g and configurations V_h

4.2.3 Passengers

Passenger itineraries are characterized by an identification number, the type of itinerary (inbound or outbound), the average ticket price, the number of passengers and the sequence of flight legs, including the cabin class for each leg. Each itinerary k in the itinerary set K must satisfy continuity constraints, connection time constraints, maximum allowed lateness constraints and functional constraints. The connection time constraints ensure that the time between two flight legs in an itinerary is at least as large as the 30-minute minimum allowed connection time, whereas the maximum allowed lateness constraints ensure that the delays for passengers who have not started their itinerary will not exceed 18 hours for domestic and continental itineraries, and 36 hours for intercontinental itineraries. There are no time limits for passengers who have already begun their itinerary. The functional constraints ensure that the destination of the modified itinerary is the same as the original itinerary's destination and that the departure time of the modified itinerary is at least as late as the time of the first flight of the original itinerary. In addition, the linking constraints ensure that the number of passengers on a given flight leg does not exceed the cabin capacity of the aircraft operating the flight leg. The passenger demand of itinerary k is d_k , and D_k and A_k are, respectively, the set of all arcs (i, j) , where node i is the departure node of itinerary k and the set of all arcs (i, j) , where node j is the arrival node of itinerary k . Finally, W_k is the set of all dummy arcs (i, j) for itinerary k , which are the cancellation arcs between the departure nodes and the arrival nodes.

4.2.4 Recovery period

A recovery period is defined for each instance. Modifications to passenger itineraries and to aircraft rotations are only possible during the recovery period. At the end of the recovery period, all aircraft need to be located at the designated airport, otherwise a penalty for non-compliant location of the aircraft is incurred.

4.2.5 Disruptions

The possible disruptions can be grouped into three categories : flight disruptions, aircraft disruptions and airport disruptions. Flight disruptions consist of flight delays and flight cancellations. Aircraft disruptions correspond to the unavailability of an aircraft and are defined by the aircraft number as well as the beginning and the end of the unavailability. Finally, airport disruptions consist of a reduced maximum number of arrivals or departures at a given airport at a given time.

The problem can be presented on a time-space network $G = (N, A)$, where $N = \{1, \dots, n\}$ is the set of airport nodes at a specific time and $A = \{(i, j) : i, j \in N, i \neq j\}$ is the set of flight legs or connection arcs between two flight legs. The objective function minimizes the weighted sum of passenger delay cost, passenger cancellation cost, passenger disutility cost, downgrading cost, operating cost and the cost for non-compliant location of aircraft. The passenger delay cost c_{ijmk}^{del} includes the disbursements incurred by the airline such as the costs of meals, drinks and lodging. These disbursements depend on the length of the delay and the initial flying time of the itinerary. The cancellations cost, c_k^{can} , includes the reimbursement of the ticket price and compensation. It also depends on the initial flying time of the itinerary. The passenger disutility cost refers to the inconvenience cost perceived by the passengers and depends on the itinerary type (domestic, continental or intercontinental) and on the cabin class. The latter cost is included in c_{ijmk} . A downgrading cost c_{km}^{down} is incurred whenever a passenger's itinerary is modified and the flight legs of the itinerary are in a lower cabin class than the original itinerary's reference cabin class. This cost depends on the level of downgrading and on the type of itinerary. The operating cost c_{ijf}^{op} includes fuel, maintenance and crew costs. It is based on the length of the flight leg and the aircraft type and does not depend on the number of passengers on board. Finally, a cost related to non-compliant location of aircraft is incurred whenever an aircraft is not positionned at the appropriate airport at the end of the recovery period and depends on the level of non-compliance.

The parameters n_j^h , n_j^g and n_j^l are, respectively, the number of aircraft of configuration h , the number of aircraft of model g and the number of aircraft of family l requested at node j , which represents an airport at the end of the recovery period. When the number of aircraft of a given family type at a given airport is smaller than the number required, a penalty cost c^{fam} is incurred. The costs c^{mod} and c^{conf} are, respectively, the penalty costs incurred when the number of aircraft of a given model and the number of aircraft of a given configuration are not matched at a given airport. Only those aircraft that have landed before the end of the recovery period are considered. To take the last three costs into consideration in our objective function, we have created for each airport three duplicates of the final node j as can be seen in Figure 4.2. The first node represents the family demand node D^{fam} , the second is the model demand node D^{mod} , and the third is the configuration demand node D^{conf} . We have also created a dummy node and arcs between these three demand nodes and the dummy node. The penalty costs for non-compliant location of aircraft are associated with these arcs. Arcs are also created between the dummy node and the sink node j . For example, in Figure 4.2 we have the following family flow conservation constraint : $x_{12} - x_{26} \leq n_j^l$. Therefore, if the value of x_{12} is greater than the number of aircraft of family l requested at node j , a non-compliant family location cost will be incurred. Finally, R^l , R^g and R^h are, respectively, the set of arcs between the family demand node and the dummy node for aircraft family l , the set of arcs between the model demand node and the dummy node for aircraft model g , and the set of arcs between the configuration demand node and the dummy node for aircraft configuration h .

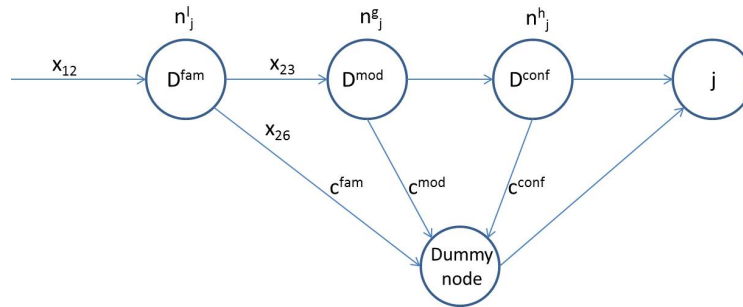


FIGURE 4.2: Demand nodes of the network

4.2.6 Mixed integer linear programming model

In order to model the aircraft and passenger recovery problem (APRP), we have developed the following mixed integer programming model, where $x_{ijf} = 1$ if and only if arc (i, j) is operated by aircraft f , and y_{ijmk} is the number of passengers from itinerary k assigned to arc (i, j) in class m . The problem is then to

(APRP)

$$\text{minimize } \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{km}^{down} y_{ijmk} \quad (4.1)$$

$$+ \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{ijmk}^{del} y_{ijmk} \quad (4.2)$$

$$+ \sum_{(i,j) \in W_k} \sum_{m \in M} \sum_{k \in K} c_k^{can} y_{ijmk} \quad (4.3)$$

$$+ \sum_{(i,j) \in A} \sum_{f \in F} c_{ijf}^{op} x_{ijf} \quad (4.4)$$

$$+ \sum_{f \in B^l} \sum_{(i,j) \in R^l} c^{fam} x_{ijf} \quad (4.5)$$

$$+ \sum_{f \in T^g} \sum_{(i,j) \in R^g} c^{mod} x_{ijf} \quad (4.6)$$

$$+ \sum_{f \in V^h} \sum_{(i,j) \in R^h} c^{conf} x_{ijf} \quad (4.7)$$

subject to

$$\sum_{i \in N} x_{ijf} - \sum_{i \in N} x_{jif} = 0 \quad f \in F, j \in N \text{ and } d_{jf} = 0 \quad (4.8)$$

$$\sum_{j \in N} x_{ijf} \leq 1 \quad f \in F, i \in N \text{ and } d_{if} = 1 \quad (4.9)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^l} x_{jif} \leq n_j^l \quad f \in F, j \in N, l \in L \text{ and } j = D^{fam} \quad (4.10)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^g} x_{jif} \leq n_j^g \quad f \in F, j \in N, g \in G \text{ and } j = D^{mod} \quad (4.11)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^h} x_{jif} \leq n_j^h \quad f \in F, j \in N, h \in H \text{ and } j = D^{conf} \quad (4.12)$$

$$\sum_{(i,j) \in I_j(p)} \sum_{f \in F} x_{ijf} \leq a_{jp} \quad p \in P, j \in N \quad (4.13)$$

$$\sum_{(i,j) \in O_i(p)} \sum_{f \in F} x_{ijf} \leq b_{ip} \quad p \in P, i \in N \quad (4.14)$$

$$\sum_{f \in F} x_{ijf} \leq 1 \quad (i, j) \in A \quad (4.15)$$

$$\sum_{(i,j) \notin D_k A_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k A_k} \sum_{m \in M} y_{jimk} = 0 \quad k \in K, j \in N \quad (4.16)$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \quad k \in K \quad (4.17)$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{fm} x_{ijf} \quad (i, j) \in A, m \in M \quad (4.18)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ijf} \leq t_f \quad f \in F \quad (4.19)$$

$$y_{ijmk} \geq 0 \quad (i, j) \in A, m \in M, k \in K \quad (4.20)$$

$$x_{ijf} = 0 \text{ or } 1 \quad (i, j) \in A, f \in F. \quad (4.21)$$

In this model, the first term of the objective function is the downgrading cost while term (4.2) is the passenger delay cost. Term (4.3) is the passenger cancellation cost and term (4.4) is the aircraft operating cost. Terms (4.5)–(4.7) are, respectively, the costs for improper positioning of family, model, and configuration. Constraints (4.8) and (4.9) are the aircraft flow conservation constraints, while constraints (4.10)–(4.12) are, respectively, the family, the model and the configuration flow conservation constraints. Constraints (4.13) and (4.14) impose the airport capacity limit. Constraints (4.15) assign each flight arc to at most one aircraft. Constraints (4.16) and (4.17) are the passenger flow conservation constraints. Constraints (4.18) define the aircraft seating capacity limits, and constraints (4.19) define the maximum flight hours of each aircraft.

All other constraints (i.e., turnaround time, transit time, flight range, connection time, maximum delay and maintenance constraints) are implicitly considered during the arc generation phase. To this end, we include the turnaround time and the transit time in the flight arcs and we include the passenger connection time in the passenger arcs. For example, a passenger arc from airport CDG to airport BOD with a flying time of 80 minutes and a connection time of 30 minutes will be set to 110 minutes. The flight range constraint, which limits the distance of the flight legs for each aircraft, is considered by creating only those flight arcs whose duration does not exceed the range limit, while the maintenance constraints are considered by creating a demand for an aircraft at the node corresponding to the beginning and the airport of the required maintenance, and setting all the corresponding flight arcs during the maintenance period to 1. Finally the maximum delay constraints are considered when creating the arrival nodes.

4.3 Solution methodology

Because of the complexity of the problem and the size of the instances, solving the APRP will either be infeasible for the larger instances or very time consuming for the smaller instances. Therefore, the following column generation post-optimization heuristic was developed. This heuristic is executed after the LNS heuristic developed by Sinclair et al. [56]. In the first part of the heuristic, we solve the aircraft and passenger recovery problem by applying the LNS heuristic. Flight delays, flight cancellations, aircraft swaps and passenger reaccommodations are considered during this phase. In the second part, the aircraft and passenger arcs selected in the LNS solution are included in the initial restricted master

problem of the column generation model. Only flight cancellations, aircraft swaps and passenger reaccommodations are considered in this phase. In this section we will first summarize the LNS heuristic of Sinclair et al. [56], and we will then describe the column generation post-optimization heuristic.

4.3.1 Large neighbourhood search heuristic

The LNS heuristic proposed by Sinclair et al. [56], which is an improvement of the LNS heuristic developed by Bisaillon et al. [19] for the 2009 ROADEF Challenge, alternates between three phases : construction, repair and improvement. An overview of the LNS solution method (taken from Bisaillon et al. [19]) is presented in Figure 4.3. We refer the readers to Sinclair et al. [56] for a detailed description of this heuristic.

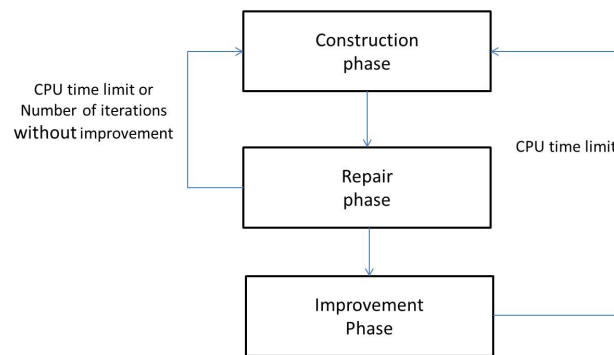


FIGURE 4.3: Overview of the LNS heuristic

Construction phase

During the construction phase, the disruptions are integrated into the planned schedule and the heuristic attempts to construct a feasible solution using five steps. First, to reach aircraft rotation feasibility, the disrupted flight legs are delayed by increments of 60 minutes. In Step 2, the heuristic tries to recreate cancelled flight legs, while Step 3 attempts to reach maintenance feasibility by removing a loop (i.e., a sequence of flight arcs beginning and ending at the same airport) scheduled before the required maintenance. In Step 4, the flights that have become infeasible due to airport capacity reduction are delayed by increments of 60 minutes. The flights that can cause the highest penalty if cancelled are considered first. Finally, Step 5 considers flights that have been cancelled because of aircraft disruptions. The

heuristic attempts to recreate these flights using available aircraft by means of a longest path algorithm.

Repair phase

The first step of the repair phase considers flights that still violate the airport capacity constraints and tries to delay them to a less congested time period. If feasibility is not reached, either a loop or the remainder of the rotation is removed. The second step attempts to reinsert cancelled flight sequences into the schedule by using available aircraft, and Step 3 attempts to accommodate on existing flights passengers on cancelled itineraries by repeatedly solving a shortest path problem. Finally, Step 4 tries to create new flight legs to accommodate passengers who still belong to cancelled itineraries by either solving a multi-commodity flow problem or a shortest path problem.

Improvement phase

The improvement phase consists of five steps. First, the algorithm attempts to improve the solution by delaying flights one at a time in order to accommodate additional passengers. Passengers are reassigned by solving a shortest path problem. In order to diversify the search, in Step 2 the algorithm attempts to improve the solution by destroying either part of an aircraft rotation, or the complete rotation. The algorithm then tries to create new flight legs using the available aircraft to reaccommodate the passengers on cancelled itineraries by means of a shortest path algorithm. Step 3 considers passengers who are positioned at a disrupted airport (i.e., at an airport with a reduced arrival or departure capacity), and attempts to redirect them to an alternate airport using available seats on existing flights. Once the passengers have been redirected to an alternate airport, the heuristic attempts to create new flights from that airport to the destination airport. The opposite procedure is also applied for passengers whose final destination is a disrupted airport. Step 4 consists in delaying subsequent passenger departures in order to accommodate additional passengers. Because of the maximum delay constraints, it is sometimes impossible to delay passengers to a less congested time period. Therefore, the algorithm attempts to delay passengers with the same origin-destination pair, but with a later departure time that would have violated the maximum delay constraints for the first group of passengers. Finally, Step 5 consists of repairing aircraft positions, by transferring, when possible, all aircraft to the required airport.

4.3.2 Column generation

Initially, the restricted master problem (RMP) includes all aircraft arcs from the LNS solution as well as all the passenger arcs found in the LNS solution. All other passenger variables are left out of the RMP because it would contain too many variables to be solvable efficiently. We start by solving the linear programming (LP) relaxation of the RMP. We define dual variables π^{ik} , λ^k and $\mu^{ijm} \leq 0$ associated with constraints (4.16)–(4.18), respectively. We also define the binary parameters d_{ijk} taking value 1 if and only if arc $(i, j) \in D_k$. The LP relaxation of the RMR is solved and the reduced costs $\bar{c}_{ijmk}$ of the passenger variables y_{ijmk} are calculated as

$$\bar{c}_{ijmk} = c_{ijmk} - (\pi^{jk} - \pi^{ik} + \lambda^k d_{ijk} + \mu^{ijm}). \quad (4.22)$$

Either all or some of the variables with negative reduced costs are included in the LP model which is solved again. The LP relaxation phase of the heuristic stops when the number of iterations reaches a certain threshold. Finally, the APRP with a limited number of variables problem is solved. Solving the instances to optimality requires high computing times, therefore they are solved with an optimality gap of 0.01%.

4.3.3 Model reduction

In order to efficiently solve the problem, it was necessary to decrease the size of the model by reducing the number of itinerary variables as well as the number of aircraft variables. We therefore grouped all passengers with the same origin-destination, departure time and cabin class. Not considering the cabin class of the passengers would greatly reduce the number of itinerary variables, but for some instances the downgrading costs would become too high. We also aggregated all aircraft having the same characteristics. Furthermore, only the flight arcs found in the solution after running the LNS for 10 minutes were included in the model. Finally, it is possible to consider only a subset of itinerary groups, so as to reduce the size of the model when necessary. We chose to exclude the itinerary groups with the latest departure times, given that these itineraries tend to have the least number of possible feasible routes and therefore could probably improve the solution by the least amount. More specifically, we exclude those itineraries whose departure time exceeds a certain time limit, that is the end of the recovery period from which we subtract $(\Delta + \Delta_1)$, where Δ and Δ_1 are time periods.

4.3.4 Multi-commodity flow problem

Solving the APRP using the column generation heuristic proved feasible only for the smaller instances. We therefore used the column generation heuristic to solve a multi-commodity flow problem for the passengers (MC-APRP) obtained by removing from the APRP model the airport capacity constraints (4.13) and (4.14) as well as the aircraft flow conservation constraints (4.8)–(4.12) and the maximum flight hours constraints (4.19). The aircraft arcs found in the LNS solution form a set $X^* = \{(i, j, f) : (i, j) \in A, f \in F \text{ and } x_{ijf} = 1\}$. Therefore, the assignment constraints (4.15) can be replaced with

$$x_{ijf} = 1 \quad (i, j, f) \in X^*. \quad (4.23)$$

Finally, the aircraft operating costs (4.4) and the aircraft improper positioning costs (4.5)–(4.7) are removed. The resulting multi-commodity flow problem is as follows :

(MCFP)

$$\text{minimize } \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{km}^{down} y_{ijmk} \quad (4.24)$$

$$+ \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{ijmk}^{del} y_{ijmk} \quad (4.25)$$

$$+ \sum_{(i,j) \in W_k} \sum_{m \in M} \sum_{k \in K} c_k^{can} y_{ijmk} \quad (4.26)$$

subject to

$$x_{ijf} = 1 \quad (i, j, f) \in X^* \quad (4.27)$$

$$\sum_{(i,j) \notin D_k A_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k A_k} \sum_{m \in M} y_{jimk} = 0 \quad k \in K, j \in N \quad (4.28)$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \quad k \in K \quad (4.29)$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{fm} x_{ijf} \quad (i, j) \in A, m \in M \quad (4.30)$$

$$y_{ijmk} \geq 0 \quad (i, j) \in A, m \in M, k \in K \quad (4.31)$$

$$x_{ijf} = 0 \text{ or } 1 \quad (i, j) \in A, f \in F. \quad (4.32)$$

The MC-APRP is solved by using the column generation heuristic described in Section 4.2.

4.4 Computational results

The column generation heuristic was implemented and tested on a computer with two Intel Westmere EP X5650 six-core processors running at 2.667 GHz and 96GB of memory. Our algorithm was tested on the instances of the 2009 ROADEF Challenge. Tables 4.1 and 4.2 provide the characteristics of these instances. Table 4.3 presents the number of aircraft and cancelled passengers as well as the number of cancelled passenger groups and aircraft groups for each instance resulting from the model reduction described in Section 4.4.2.

TABLE 4.1: Characteristics of the B instances

	Recovery	Nb. of	Nb. of	Nb. of	Nb. of	Nb. flight	Nb. aircraft	Nb. airport
	period	aircraft	airports	flights	itineraries	disruptions	disruptions	disruptions
B01	36	255	45	1,423	11,214	230	0	0
B02	36	256	45	1,423	11,214	255	0	0
B03	36	256	45	1,423	11,214	229	1	0
B04	36	256	45	1,423	11,214	230	0	1
B05	36	256	45	1,423	11,214	0	0	34
B06	36	256	45	1,423	11,214	230	0	0
B07	36	256	45	1,423	11,214	255	0	0
B08	36	256	45	1,423	11,214	229	1	0
B09	36	256	45	1,423	11,214	230	0	1
B10	36	256	45	1,423	11,214	77	0	34

We have run the LNS for 10 minutes before executing the column generation heuristic. Solving the MIP of Section 3 exactly is very time consuming and proved feasible for only the two smaller instances XA01 and XA03. Table 4.4 presents the solution cost obtained after 10 iterations and including all variables with negative reduced costs, as well as all itinerary groups. It also reports the computing time, the best known solution, the solution cost after

running the LNS for 10 minutes, and the gaps with respect to the best known solution and the LNS solution.

TABLE 4.2: Characteristics of the XA, XB and X instances

	Recovery	Nb. of	Nb. of	Nb. of	Nb. of	Nb. flight	Nb. aircraft	Nb. airport
	period	aircraft	airports	flights	itineraries	disruptions	disruptions	disruptions
XA01	14	85	35	608	1,943	83	3	0
XA02	52	85	35	608	3,959	0	3	407
XA03	14	85	35	608	1,872	83	3	0
XA04	52	85	35	608	3,773	0	3	407
XB01	36	256	45	1,423	11,214	229	3	0
XB02	52	256	44	1,423	11,214	0	1	34
XB03	36	256	45	1,423	11,214	228	4	0
XB04	52	256	44	1,423	11,214	0	4	34
X01	78	618	168	2,178	28,308	0	1	1
X02	78	618	168	2,178	28,308	0	1	0
X01	78	618	168	2,178	29,151	0	1	1
X02	78	618	168	2,178	29,151	0	1	0

Table 4.5 presents the variations in the different costs for the LNS heuristic and the column generation heuristic. We can see from this table that the decrease in the operating cost has a significant impact on the total cost for both of these instances. For instance XA01, only four additional passengers were reaccommodated with the column generation heuristic, and five additional passengers were reaccommodated for the XA03 instance, therefore limiting the impact of the cancellation costs on the total cost variation. Finally, the downgrading costs have the lowest impact on the total cost variation. Table 4.6 presents the same information as Table 4.4 for all instances that were solved using the MC-APRP, for five iterations, and including all variables with negative reduced costs. Because of the size of some instances, feasible solutions were found for only 12 out of 20 instances. The best known solutions include all solutions obtained by the finalist teams of the 2009 ROADEF Challenge, found within 10 minutes of computing time, as well as the solutions found by Acuna-Agost [6] with a computing time varying between 602.73 and 1073.80 seconds, and the solutions found by Sinclair et al. [56] with a computing time limit of 60 minutes. We also present the solutions found after running the LNS heuristic for 10 minutes, since the computer used is different from that of Sinclair et al. [56] and the results vary for some instances (see instance XA01 in Table 4.4). Finally, Table 4.6 presents the solution cost when running the LNS heuristic for the same time as the time used to solve the MC-APRP (i.e., 10 minutes for the LNS, plus the computing time reported in column 10) and the associated gaps. The solution cost

TABLE 4.3: Characteristics of the aggregated passengers and aircraft

Instance	Nb. of aircraft	Nb. cancelled itineraries	Nb. aircraft groups	Nb. cancelled itinerary groups
B01	255	8,209	60	4,362
B02	256	8,236	60	4,358
B03	256	8,208	60	4,364
B04	256	8,215	60	4,367
B05	256	12,481	67	6,418
B06	256	8,775	60	4,472
B07	256	8,844	60	4,462
B08	256	8,728	60	4,469
B09	256	8,818	60	4,472
B10	256	13,254	67	6,544
XA01	85	1,943	15	416
XA02	85	4,530	15	1,931
XA03	85	1,872	15	461
XA04	85	4,337	15	2,009
XB01	256	8,243	60	4,366
XB02	256	12,416	67	6,412
XB03	256	8,816	60	4,472
XB04	256	13,133	68	6,534
X01	618	28,622	152	18,390
X02	618	28,389	152	18,085
X03	618	29,624	153	18,750
X04	618	29,335	153	18,407

when running the LNS heuristic for the same time as the time used to solve the MC-APRP is not presented for the instances for which no feasible solution was found with the column generation post-optimization heuristic.

TABLE 4.4: Recovery costs for the APRP

	Best	LNS 10 min	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
XA01	99009.20	96391.70	51854.85	47.58%	46.20%	83.83
XA03	262945.25	262945.25	186273.60	29.16%	29.16%	83.16

Including all itineraries is either very time consuming for the smaller instances or infeasible for the larger ones. Therefore, in order to compute feasible solutions within a reasonable computing time, we have excluded the itineraries whose departure time exceeded a certain time limit, that is the end of the recovery period minus $(\Delta + \Delta_1)$. We have tested our algorithm for the following values of Δ : 840, 780, 720, 660 600 and 540, setting $\Delta_1 = 300$ for instances with a recovery period of 52 hours, and $\Delta_1 = 0$ for all other the instances. We also included all variables with negative reduced cost, except for the instances that return an infeasible solution in Table 4.3, where we limited the number of negative reduced cost variables to 250,000. The number of iterations remained equal to five.

TABLE 4.5: Variation in the different costs

	XA01 LNS	XA01 CG	Variation	% tot. cost decr.	XA03 LNS	XA03 CG	Variation	% tot. cost decr.
Delay costs	65466.70	71454.85	-9.14%	-13.45%	110445.25	84098.60	23.85%	34.36%
Cancellation costs	5625.00	0	100%	17.81%	25450	14675.00	42.34%	14.01%
Downgrading costs	7500.00	3750.00	50.00%	11.87%	9000.00	5850.00	35.00%	4.11%
Operating costs	-13200.00	-43350	69.55%	97.33%	58050.00	41650.00	28.25%	21.39%
Location costs	31000.00	20000.00	35.48%	24.70%	60000.00	40000.00	33.33%	26.09%

TABLE 4.6: Recovery cost for the MC-APRP

	Best	LNS 10 min	LNS full time			Column generation post-optimization			
			Cost	Improvement	Improvement	Cost	Improvement	Improvement	seconds
				wrt best	wrt LNS		wrt best	wrt LNS	
B01	797903	843084	843084	-5.66%	0%	502823	36.98%	40.36%	709.15
B02	1020906	1219886	1219886	-19.49%	0%	678375	33.54%	44.38%	739.26
B03	831642	866408	866408	-4.18%	0%	542537	34.76%	37.38%	818.05
B04	907752	970523	968137	-6.65%	0.25%	602920	33.58%	37.72%	794.06
B05	7411929	8836265	no feasible solution found						
B06	2586412	3000770	2782830	-7.59%	7.26%	1550188	40.06%	44.29%	1307.75
B07	4184662	4516850	4436946	-6.03 %	1.77%	2846093	31.99%	35.85%	1502.81
B08	2845990	3155738	2909880	-2.24%	7.79%	1629301	42.75%	44.01%	1262.30
B09	2564760	3105536	2797217	-9.06	9.93%	1528599	40.40%	45.35%	1203.05
B10	30876122	32979391	no feasible solution found						
XA02	1465060	1596660	1596301	-8.96%	0.02%	951386	35.06%	38.88%	209.46
XA04	3866093	4268885	4246729	-9.85%	0.52%	2697196	30.65%	37.19%	302.61
XB01	1002908	1059488	1059488	-5.64%	0%	641028	36.08%	39.50%	846.94
XB02	8080074	9468598	no feasible solution found						
XB03	3878297	4536106	4249711	-9.06%	6.31%	2465650	36.42%	41.98%	1346.05
XB04	32707740	34332059	no feasible solution found						
X01	-182410	142443	no feasible solution found						
X02	-206073	-23433	no feasible solution found						
X03	1212619	1353316	no feasible solution found						
X04	103980	175752	no feasible solution found						

Table 4.7 presents the average gap with respect to the best known solution value, the average gap with respect to the LNS solution value, the average computing time and the maximum time for parameter sets P2 $\Delta = 840$, P3 $\Delta = 780$, P4 $\Delta = 720$, P5 $\Delta = 660$, P6 $\Delta = 600$ and $\Delta = 540$. From Table 4.7, we can see that using a smaller Δ (i.e., a larger number of itinerary variables) leads to better solutions, but also larger computing times. Table 4.8 presents the solution costs, the gap with respect to the best known solution value, the gap with respect to the LNS heuristic solution value and the time for parameter set P2 with $\Delta = 840$. The same information for parameter sets P3–P7 is reported in Sinclair et al. [55] and can be found in Appendix 1. Excluded from Table 4.7 and 4.8 are the XA02 and XA04 instances for which we were able to obtain feasible solutions within a reasonable computing time without limiting the number of itineraries included. The X01–X04 instances are also excluded. These instances have a larger recovery period and they will be treated separately.

TABLE 4.7: Characteristics of the solution costs

	P2	P3	P4	P5	P6	P7
Average % wrt best	26.03%	28.36%	30.22%	32.67%	34.36%	36.04%
Average % wrt LNS	33.49%	35.52%	37.18%	39.36%	40.90 %	42.40%
Average time	919.96	969.60	1004.87	1130.64	1188.69	1219.20
Maximum time	1738.49	1940.20	1929.25	1911.16	2062.03	2062.03

TABLE 4.8: Recovery costs for the multi-commodity flow problem for parameter set P2

	Best	LNS 10 min	Cost P2	Improvement wrt best	Improvement wrt LNS	Seconds
B01	797903.00	843084.35	587860.10	26.32%	30.27%	580.10
B02	1020906.00	1219885.75	849390.15	16.80%	30.37%	588.21
B03	831642.00	866408.20	633662.90	23.81%	26.86%	588.58
B04	907752.00	970522.80	676278.55	25.50%	30.32%	582.17
B05	7411929.45	8836264.50	4696636.55	36.63%	46.85%	1188.03
B06	2586412.25	3000770.75	1911750.60	26.08%	36.29%	802.64
B07	4184662.00	4516850.20	3526297.60	15.73%	21.93%	868.45
B08	2845990.30	3155738.00	2052855.90	27.87%	34.95%	854.71
B09	2564759.60	3105536.35	1824978.80	28.84%	41.23%	842.06
B10	30876122.15	32979391.35	23942711.60	22.46%	27.40%	1738.49
XB01	1002908.25	1059488.50	743298.15	25.89%	29.84%	587.09
XB02	8080073.65	9468598.10	4719373.30	41.59%	50.16%	1188.78
XB03	3878297.00	4536105.5	2989665.25	22.91%	34.09%	853.97
XB04	32707740.00	34332058.95	24633162.40	24.69%	28.25%	1616.23
			average	26.08%	33.49%	919.96

The number of iterations and the number of variables with negative reduced costs included in the LP relaxation can have a significant impact both on the solution cost and on the computing time. We have therefore tested different combinations of number iterations and number of variables. Figure 4.4 presents the average solution cost and the average computing time when 100%, 80%, 60%, 40% and 20% of the variables with negative reduced costs are included in the LP relaxation for a number of iterations varying between two and nine, and for $\Delta = 720$. The detailed solutions and computing times for each instance are reported in Sinclair et al. [55] and can be found in Appendix 1.

Since the quality of the heuristic not only depends on the solution cost, but also on the computing time, opting to include 80% of the variables with negative reduced cost and running it for four iterations seems to be a good compromise. Table 4.9 presents the solutions cost, the gap with respect to the best known solution value, the gap with respect to the LNS solution value, and the computing time for parameter set P2, with 80% of the negative

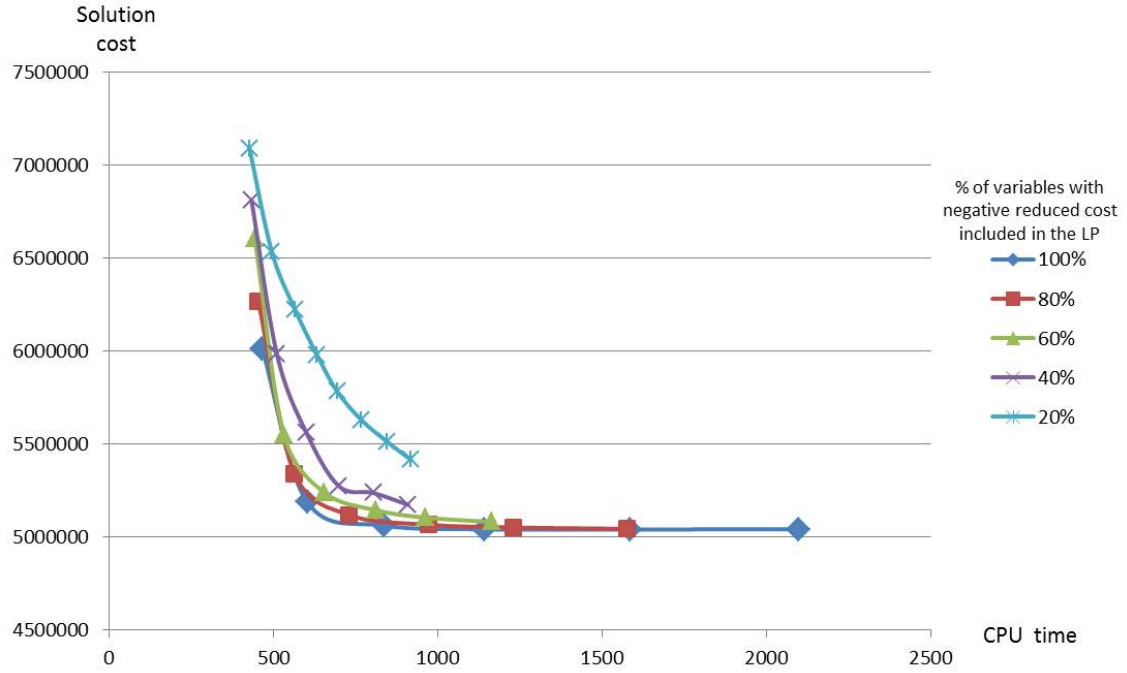


FIGURE 4.4: Average solution cost vs computing time

reduced cost variables included and a limit of four iterations. The same information for parameter sets P3–P7 is reported in Sinclair et al. [55] and in Appendix 1. Table 4.10 presents the average gap with respect to the best known solution, the average gap with respect to the LNS solution, the average computing time and the maximum time for parameter sets P2–P7, also with 80% of the negative reduced cost variables included and a limit of four iterations. Comparing Table 4.7 and 4.10, we see that the difference in the solution costs when using 80% of the negative reduced costs and four iterations is small, while the computing times are reduced significantly. The maximum computing time is also much smaller.

Tables 4.11 and 4.12 present the variation in the number of cancelled passengers, the total costs, the delay costs, the cancellation costs and the downgrading costs for the LNS and the column generation algorithm, when using parameter set P5, 80% of the negative reduced cost variables and four iterations. Since we are solving the multi-commodity flow network, the operating cost and the non-compliant location costs do not differ between the LNS and the column generation algorithm. Tables 4.11 and 4.12 also show the impact of each cost variation on the total cost decrease.

TABLE 4.9: Recovery costs for parameter sets P2

	Cost P2	Improvement wrt best	Improvement wrt LNS	Seconds
B01	575503.35	27.87%	31.74%	431.04
B02	851107.35	16.63%	30.23%	430.47
B03	622202.90	25.18%	28.19%	429.65
B04	677607.50	25.35%	30.18%	417.70
B05	4502096.55	39.26%	49.05%	1014.33
B06	1918930.30	25.81%	36.05%	557.91
B07	3531149.10	15.62%	21.82%	548.22
B08	2073673.25	27.14%	34.29%	497.97
B09	1834045.75	28.49%	40.94%	482.18
B10	24215805.25	21.57%	26.57%	1384.82
XB01	728317.55	27.38%	31.26%	446.23
XB02	4798354.25	40.61%	49.32%	994.08
XB03	2995005.30	22.78%	33.97%	527.66
XB04	24955836.70	23.70%	27.31%	1315.98
average		26.24%	33.64%	677.02

TABLE 4.10: Characteristics of the solution costs

	P2	P3	P4	P5	P6	P7
Average % wrt best	26.24%	27.36%	29.67%	32.17%	33.15%	34.99%
Average % wrt LNS	33.64%	34.66%	36.68%	38.90%	39.85 %	41.49%
Average time	677.02	692.21	730.88	790.91	839.95	864.40
Maximum time	1384.82	1454.48	1431.68	1550.66	1605.45	1763.08

Table 4.13 presents the solution costs for the X01–X04 instances, as well as the best known solution values, the LNS solution value with an execution time of 10 minutes, the gaps with to the best known solution, the gaps with respect to the LNS solution, and the computing time. Table 4.13 also presents the solution cost when running the LNS heuristic for the same time as for the MC-APRP. Because of the size of the instances, only a small group of itineraries are considered, which leads to smaller improvements in the solution costs. As can be seen, the column generation heuristic improves the LNS solution value, but is quite far from the best known solution value. This can be explained by the fact that the best known solutions for these four instances were obtained by Sinclair et al. [56] by running the LNS for 60 minutes. Also, because of the size of these instances, the LNS heuristic only performs one iteration within the 60 minute time limit. Consequently the time needed to find a good feasible solution is much longer. Therefore, we increased the execution time of the LNS heuristic to 60 minutes before starting the column generation heuristic. Table 4.14

TABLE 4.11: Cost variations and % of total cost decrease for the B instances

	Total cost	Delay cost	Cancellation cost	Downgrading cost	Nb. cancelled passengers
B01 LNS	843084	1221164	38521	315250	18
B01 CG	545788	1073438	6750	197450	3
% tot. cost decr.		49.69%	10.69%	39.62%	
B02 LNS	1219886	1298970	98366	647550	63
B02 CG	735904	1167634	27720	365550	23
% tot. cost decr.		27.14%	14.60%	58.27%	
B03 LNS	866408	1222002	58256	350300	48
B03 CG	583735	1090247	40388	217250	39
% tot. cost decr.		46.61%	6.32%	47.07%	
B04 LNS	970523	130215	61808	370450	36
B04 CG	618170	1160170	18350	203500	15
% tot. cost decr.		40.28%	12.33%	47.38%	
B05 LNS	8845301	4602247	3911204	1676200	4132
B05 CG	4361107	4085954	808353	811150	419
% tot. cost decr.		11.51%	69.20%	19.29%	
B06 LNS	3000771	1577289	1259032	371000	1088
B06 CG	1709663	1297081	398332	220800	249
% tot. cost decr.		21.70%	66.66%	11.63%	
B07 LNS	4516850	1824757	2360143	596950	1357
B07 CG	3265724	1437328	1692946	400450	761
% tot. cost decr.		30.97%	53.33%	15.71%	
B08 LNS	3155783	1546442	1500341	345450	1189
B08 CG	1804324	1295653	504722	240400	274
% tot. cost decr.		18.56%	73.67%	7.77%	
B09 LNS	3105536	1722312	1184724	390350	712
B09 CG	1601768	1306858	257710	229050	150
% tot. cost decr.		27.63%	61.65%	10.73%	
B10 LNS	34610231	4678224	29587106	1168500	28274
B10 CG	23142418	6521812	17018506	1168500	15078
% tot. cost decr.		-16.08%	109.60%	6.48%	

presents the solution costs, the best known solution value, the 60 minute LNS solution costs, the gap with respect to the best known solution solution, the gap with respect to the 60 minute LNS solution value, and the computing time.

Table 4.15 presents the best known solution and the best solution obtained with the column generation heuristic using the optimal parameter set for each instance. The table also presents the details of the cost, i.e., the downgrading cost, the delay cost, the cancellation cost and the operating cost as well as the number of cancelled passengers.

TABLE 4.12: Cost variations and % of total cost decrease for the XB instances

	Total cost	Delay cost	Cancellation cost	Downgrading cost	Nb. cancelled passengers
XB01 LNS	1059489	1310811	101577	343850	67
XB01 CG	687617	1138457	45310	200600	38
% tot. cost decr.		46.35%	15.13%	38.52%	
XB02 LNS	9468598	4525173	4605325	1678300	5037
XB02 CG	4694816	4234184	933833	867000	485
% tot. cost decr.		6.10%	76.91%	16.99%	
XB03 LNS	4536106	813870	2326235	571400	1860
XB03 CG	2625381	1610569	856862	333350	518
% tot. cost decr.		10.64%	76.90%	12.46%	
XB04 LNS	34332059	4531683	29385176	1947000	28728
XB04 CG	23786339	6366913	17790776	1160450	15868
% tot. cost decr.		-17.40%	109.94%	7.46%	

TABLE 4.13: Recovery costs for the MC-APRP instances X01-X04

	Best	LNS 10 min	LNS full time			Column generation post-optimization			
			Cost	Improvement wrt best	Improvement wrt LNS	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
X01	-182410	142443	68172	-137.37%	52.14%	125019	-167.87%	13.08%	1456.97
X02	-206073	-23433	-94257	-54.26%	302.24%	-36858	-82.11%	57.29%	1179.39
X03	1212619	1353316	959046	20.91%	29.13%	1218000	-0.44%	10.01%	1299.26
X04	103981	175752	152337	-46.50%	13.32%	140356	-34.98%	20.14%	1288.48

TABLE 4.14: Recovery costs for the MC-APRP instances X01-X04

	Best	LNS 60 mins	Cost	Improvement wrt best	Improvement wrt LNS	Seconds
X01	-182410	-193133	-219167	20.15%	13.48 %	1314.35
X02	-206073	-345752	-361411	75.38 %	4.53%	1300.84
X03	1212619	913091	878027	27.59 %	3.84%	1341.52
X04	103981	108565	103821	0.15%	4.37%	1330.13

4.5 Conclusions

We have presented a post-optimization column generation heuristic which, when executed after the LNS heuristic, yields the best known solutions for all of the instances of the 2009 ROADEF challenge within a reasonable computing time. We have also shown that this algorithm can be modified to solve large instances by only considering the passenger variables. Given that solving the MIP for the smaller instances considerably reduces the aircraft operating costs, being able to solve the MIP for all of the instances should yield sub-

TABLE 4.15: Best solutions for B and X instances

	Previously best known solution	Best solution CG	Improvement	Delay cost	Cancellation cost	Downgrading cost	Operating cost	Nb. cancelled passengers
B01	797903	502823	36.98%	1049623	6750	178300	-731850	3
B02	1020906	678466	33.54%	1147445	27720	328300	-825000	23
B03	831642	542537	34.76%	1073350	25388	197950	-764150	31
B04	907752	602920	33.58%	1147070	29350	201350	-763850	15
B05	7411929	4165084	43.81%	3941977	785906	787550	-1350350	405
B06	2586412	1550188	40.06%	1210163	362225	184350	-206550	230
B07	4184662	2846093	31.99%	1312799	818195	351000	-265000	595
B08	2845990	1629301	42.75%	1207453	443048	215250	-236450	255
B09	2564760	1528599	40.40%	1241109	259140	220200	-191850	143
B10	30876122	22335381	27.66%	6433574	16332757	1135450	-1566400	14439
XA01	99009	51855	47.63%	71455	0	3750	-43350	0
XA02	1465060	930866	36.46%	1170843	138103	60750	-479830	157
XA03	262945	186274	29.15%	84099	14675	5850	41650	19
XA04	3866093	2687460	30.49%	2146040	519030	59850	-58460	599
XB01	1002908	641028	36.08%	1114703	41625	181450	-736750	34
XB02	8080074	4469525	44.68%	4082104	901222	826400	-1340200	468
XB03	3878297	2465650	36.42%	1508790	834510	297750	-235400	498
XB04	32707740	22956737	29.81%	6337035	17013802	1137700	-1571800	15174
X01	-182410	-219167	20.15%	919383	0	200300	-1379850	0
X02	-206073	-361411	75.38%	807789	0	158850	-1351050	0
X03	1212619	878027	27.59%	848534	33692	167650	-191580	17
X04	103980	103821	0.15%	116621	13150	56900	-182850	5

stantial improvements. Future research should focus on developing solution methods capable of solving the MIP for larger instances, such as embedding the column generation algorithm within a rolling-time horizon framework. One should also attempt to solve more realistic instances, such as those with a higher number of disruptions. Future research should also focus on integrating the crew recovery component in the solution methodology instead of solving the problem in a sequential manner.

Acknowledgments This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 227837-09 and 39682-10. This support is gratefully acknowledged. Thanks are due to a referee whose comments helped improve the paper.

4.6 Appendix 1

TABLE 4.16: Recovery costs for parameter set P3

	Best	LNS 10mins	Cost P3	% wrt best	% wrt LNS	Sec.
B01	797903.00	843084.35	578996.85	27.44%	31.32%	616.00
B02	1020906.00	1219885.75	801172.35	21.52%	34.32%	628.08
B03	831642.00	866408.20	619228.65	25.54%	28.53%	620.24
B04	907752.00	970522.80	666571.55	26.57%	31.32%	610.91
B05	7411929.45	8836264.50	4357125.45	41.21%	50.69%	1320.66
B06	2586412.25	3000770.75	1841335.15	28.81%	38.64%	899.44
B07	4184662.00	4516850.20	3294329.9	21.28%	27.07%	903.03
B08	2845990.30	3155738.00	1948041.15	31.55%	38.27%	907.84
B09	2564759.60	3105536.35	1790824.85	30.18%	42.33%	857.46
B10	30876122.15	32979391.35	23475640.00	23.97%	28.82%	1940.20
XB01	1002908.25	1059488.50	729401.90	27.27%	31.16%	630.28
XB02	8080073.65	9468598.10	4864923.40	39.79%	48.62 %	1213.81
XB03	3878297.00	4536105.50	2893335.60	25.40%	36.22 %	941.38
XB04	32707740.00	34332058.95	24039320.60	26.50%	29.98%	1549.04
			average	28.36%	35.52%	974.17

TABLE 4.17: Recovery costs for parameter set P4

	Best	LNS 10mins	Cost P4	% wrt best	% wrt LNS	Sec.
B01	797903.00	843084.35	555470.10	30.38%	34.11%	662.73
B02	1020906.00	1219885.75	775469.65	24.04%	36.43%	674.24
B03	831642.00	866408.20	606598.70	27.06%	29.99%	661.24
B04	907752.00	970522.80	646558.30	28.77%	33.38%	653.40
B05	7411929.45	8836264.50	4316033.90	41.77%	51.16%	1248.94
B06	2586412.25	3000770.75	1775446.00	31.35%	40.83%	1006.57
B07	4184662.00	4516850.20	3254117.05	22.24%	27.96%	1046.52
B08	2845990.30	3155738.00	1888859.75	33.63%	40.15%	467.56
B09	2564759.60	3105536.35	1749523.45	31.79%	43.66%	922.89
B10	30876122.15	32979391.35	23254300.90	24.69%	29.49%	1929.25
XB01	1002908.25	1059488.50	713533.60	28.85%	32.65%	672.49
XB02	8080073.65	9468598.10	4602448.05	43.04%	51.39%	1260.89
XB03	3878297.00	4536105.50	2769747.60	28.58%	38.94%	996.88
XB04	32707740.00	34332058.95	23915783.10	26.88%	30.34%	1873.13
			average	30.22%	37.18%	1005.48

TABLE 4.18: Recovery costs for parameter set P5

	Best	LNS 10mins	Cost P5	% wrt best	% wrt LNS	Sec.
B01	797903.00	843084.35	540210.10	32.30%	35.92%	733.36
B02	1020906.00	1219885.75	733204.05	28.18%	39.90%	763.40
B03	831642.00	866408.20	578656.25	30.42%	33.21%	734.05
B04	907752.00	970522.80	615100.05	32.24%	36.62%	730.25
B05	7411929.45	8836264.50	4455491.80	39.89%	49.58%	1292.18
B06	2586412.25	3000770.75	1696005.30	34.43%	43.48 %	1105.16
B07	4184662.00	4516850.20	3254878.55	22.22%	27.94%	1153.97
B08	2845990.30	3155738.00	1789622.85	37.17%	43.29 %	1183.66
B09	2564759.60	3105536.35	1593777.50	37.86%	48.68%	1085.71
B10	30876122.15	32979391.35	22760940.80	26.28%	30.98%	1911.16
XB01	1002908.25	1059488.50	684761.80	31.72%	35.37%	777.72
XB02	8080073.65	9468598.10	4570117.40	43.44%	51.73%	1297.07
XB03	3878297.00	4536105.5	2598938.05	32.99%	42.71%	1131.81
XB04	32707740.00	34332058.95	23462950.5	28.26%	31.66 %	1821.47
			average	32.67%	39.36%	1122.93

TABLE 4.19: Recovery costs for parameter set P6

	Best	LNS 10mins	Cost P6	% wrt best	% wrt LNS	Sec.
B01	797903.00	843084.35	512706.85	35.74%	39.19%	805.11
B02	1020906.00	1219885.75	722092.15	29.27%	40.81%	768.24
B03	831642.00	866408.20	555054.95	33.26%	35.94%	799.48
B04	907752.00	970522.80	609125.30	32.90%	37.24%	801.64
B05	7411929.45	8836264.50	4357412.00	41.21%	50.69%	1351.99
B06	2586412.25	3000770.75	1655125.5	36.01%	44.84%	1143.71
B07	4184662.00	4516850.20	3210970.60	23.277%	28.91%	1235.35
B08	2845990.30	3155738.00	1734861.05	39.04%	45.03%	1156.57
B09	2564759.60	3105536.35	1549655.05	39.58%	50.10%	1157.36
B10	30876122.15	32979391.35	22304388.90	27.76%	32.37%	2062.03
XB01	1002908.25	1059488.50	657714.40	34.42%	37.92%	807.57
XB02	8080073.65	9468598.10	4702686.45	41.80%	50.33%	1318.87
XB03	3878297.00	4536105.5	3332676.20	14.07%	26.53%	1664.38
XB04	32707740.00	34332058.95	22973144.70	29.76%	33.09%	1892.55
			average	32.72%	39.50%	1211.78

TABLE 4.20: Recovery costs for parameter set P7

	Best	LNS 10mins	Cost P7	% wrt best	% wrt LNS	Sec.
B01	797903.00	843084.35	505382.10	36.66%	40.06%	788.08
B02	1020906.00	1219885.75	679362.40	33.45%	44.31%	782.16
B03	831642.00	866408.20	549928.45	33.87%	36.53%	824.20
B04	907752.00	970522.80	604307.05	33.43%	37.73%	790.83
B05	7411929.45	8836264.50	4357412.00	41.21%	50.69%	1351.99
B06	2586412.25	3000770.75	1608877.10	37.80%	46.38%	1168.39
B07	4184662.00	4516850.20	2848062.00	31.94%	36.95%	1417.39
B08	2845990.30	3155738.00	1668229.60	41.38%	47.14%	1260.84
B09	2564759.60	3105536.35	1512454.30	41.03%	51.30%	1202.05
B10	30876122.15	32979391.35	22304388.90	27.76%	32.37%	2062.03
XB01	1002908.25	1059488.50	647134.40	35.47%	38.92%	855.88
XB02	8080073.65	9468598.10	4641007.70	42.56%	50.99%	1317.26
XB03	3878297.00	4536105.5	2469046.20	36.34%	45.57%	1346.05
XB04	32707740.00	34332058.95	22973144.70	29.76%	33.09%	1892.55
			average	36.04%	42.40%	1219.20

4.7 Appendix 2

TABLE 4.21: Recovery costs for B01

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	343.47	632966.60	340.24	639226.60	342.18	645434.75	337.16	669818.50	332.28	696919.85
3	418.38	569063.80	407.94	571152.65	410.22	591240.40	384.50	627908.00	379.52	673195.35
4	505.03	556858.40	490.84	557536.10	547.35	565343.90	448.35	608357.80	452.54	671416.65
5	580.43	556272.85	546.25	555755.50	530.52	559140.10	549.01	588653.00	484.41	665846.85
6	626.48	555556.60	605.12	555704.40	576.36	556468.15	544.49	581336.70	524.55	652952.60
7	685.94	555615.35	662.61	554715.10	621.74	556165.45	588.27	565700.00	564.76	639630.05
8									605.96	612507.70
9									648.50	605947.35

TABLE 4.22: Recovery costs for B02

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	343.66	850333.25	341.40	855252.85	338.88	867695.75	337.45	881771.30	337.79	910847.70
3	427.52	787333.55	407.39	791124.15	395.17	809802.05	381.77	838606.60	377.22	885092.45
4	505.38	779023.35	487.48	776900.35	456.58	785674.95	464.85	812699.40	439.79	874537.75
5	630.47	776136.90	557.70	776719.30	532.68	777841.50	510.59	799355.10	482.75	867796.55
6	692.27	775378.65	647.66	775906.55	592.23	776434.40	588.28	787226.80	538.25	848677.65
7	705.79	776118.90	668.78	775821.80	661.03	776146.00	639.77	780152.30	569.90	840102.65
8									616.16	822147.45
9									667.24	811759.00

TABLE 4.23: Recovery costs for B03

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	348.27	682424.35	338.17	687423.6	340.27	699451.50	334.07	708906.10	331.02	726914.85
3	424.71	620341.10	416.30	619950.90	405.52	645736.60	384.15	667618.95	406.48	712147.05
4	515.71	608133.00	486.71	612682.95	453.27	621436.35	437.82	649302.95	456.69	704617.05
5	566.15	606898.95	543.81	608371.80	505.96	610485.95	484.35	635351.55	491.21	697721.10
6	622.64	606531.70	600.43	606757.55	590.49	608232.80	541.22	626130.95	529.06	681267.75
7	705.38	606545.15	662.29	606542.50	632.26	606450.85	590.48	615751.95	579.11	673469.90
8									619.42	660024.90
9									659.52	650037.20

TABLE 4.24: Recovery costs for B04

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	346.53	717731.25	340.77	724447.95	335.26	679386.95	333.06	748512.65	329.86	776334.85
3	425.29	659066.20	406.35	666106.00	396.18	679386.95	384.60	707032.25	371.54	750114.20
4	500.82	648986.05	482.24	648759.80	457.02	657845.75	428.40	688838.50	415.61	739305.85
5	578.67	646629.05	545.70	649762.60	513.40	650849.95	493.54	675910.00	466.08	734707.90
6	640.55	649191.30	599.12	647679.35	568.84	648712.70	540.27	667268.75	520.12	730828.45
7	703.81	648816.55	688.09	647646.85	626.34	648085.05	594.77	656851.90	572.22	724927.70
8									625.69	709627.45
9									652.71	697418.20

TABLE 4.25: Recovery costs for B05

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	699.14	5385601.95	682.39	5535441.15	655.39	6693350.00	625.11	5904561.55	622.63	6244537.15
3	879.83	4483060.55	836.02	4614819.75	766.89	4822710.45	745.32	5047205.40	724.93	5476481.50
4	1138.97	4337311.65	1025.02	4399960.30	953.88	4607764.70	884.21	4774330.45	817.73	5146162.70
5	1886.54	4274746.65	1426.92	4322348.05	1118.82	4443482.25	1017.98	4665472.60	912.40	4978035.95
6	2782.31	4264403.15	1805.23	4277724.80	1327.64	4348407.80	1133.00	4561167.25	1019.87	4866138.00
7	3831.70	4279919.50	2511.06	4267283.10	1693.83	4944828.30	1234.23	5188736.00	1139.13	4799115.85
8									1219.37	4761844.75
9									1367.91	4698596.25

TABLE 4.26: Recovery costs for B06

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	391.59	2062707.80	389.19	2086352.65	383.17	2163088.85	399.74	2227873.55	364.16	2334223.15
3	480.52	1851530.80	458.70	1866057.95	446.76	1889193.10	421.79	1963177.25	422.50	2087950.30
4	667.25	1780911.20	601.55	1789027.80	550.90	1804727.15	493.71	1867129.70	468.05	1979824.55
5	780.42	1775516.75	713.57	1779428.70	633.78	1785101.70	562.21	1822953.60	519.03	1948764.95
6	882.35	1772750.25	839.29	1776840.10	725.56	1786043.30	645.45	1810610.05	574.46	1929058.95
7	1090.18	1776648.75	965.59	1783162.00	810.58	1781039.65	727.49	1795332.30	638.37	1915077.70
8									699.13	1894870.10
9									739.10	1869774.30

TABLE 4.27: Recovery costs for B07

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	389.71	3519259.60	373.75	3553704.15	370.45	3587332.30	363.43	3643729.60	360.58	3771653.85
3	475.13	3302760.90	462.48	3325080.25	433.54	3366315.90	420.01	3442853.90	412.35	3545572.90
4	602.00	3256240.90	567.66	3266261.75	525.21	3290384.35	498.65	3368762.75	480.36	3488505.60
5	746.47	3254310.95	689.40	3256160.50	648.34	3269309.00	565.49	3318615.65	534.49	3458098.10
6	930.88	3253521.85	810.63	3252883.50	729.61	3256565.90	634.24	3286276.95	585.85	3429114.95
7	1081.63	3251191.9	953.86	3250029.25	832.06	3259860.55	700.69	3279996.70	622.77	3400103.95
8									695.56	3393522.70
9									747.62	3361858.60

TABLE 4.28: Recovery costs for B08

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	385.99	2267176.90	374.89	2301251.80	368.86	2337596.95	362.54	2408051.70	356.89	2472639.60
3	484.96	1976752.70	456.62	1997346.35	440.79	2027598.9	424.02	2158612.40	409.37	2266991.25
4	596.24	1893827.05	610.53	1896322.20	565.31	1918878.90	484.44	2041773.50	480.43	2132520.90
5	741.65	1885067.15	709.41	1890475.05	650.57	1899108.95	538.74	1948790.05	522.93	205615.95
6	884.65	1885046.65	849.54	1891981.75	715.71	1891230.15	624.72	1926090.55	567.03	5075696.50
7	1051.90	1887808.60	974.92	1888788.20	835.51	1890126.60	682.51	1912937.70	612.50	2054995.25
8									694.37	2030441.20
9									729.10	2008188.85

TABLE 4.29: Recovery costs for B09

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	390.66	2362421.30	380.31	2389923.95	363.25	2416870.35	358.78	2480061.85	357.17	2528233.50
3	487.54	1788349.70	475.54	1801124.15	437.88	1839690.00	416.89	1980097.90	404.57	2031048.35
4	618.40	1753481.35	564.22	1757797.35	535.50	1775049.55	492.33	1856202.40	454.81	1974452.80
5	741.88	1754129.55	689.55	1755794.75	618.92	1761333.05	556.90	1801434.05	529.17	1944266.00
6	917.37	1752061.20	784.41	1753527.30	699.13	1752925.20	622.12	1781019.95	562.30	1912182.25
7	1000.58	1749941.45	883.59	1751596.70	779.13	1752471.40	688.45	1769276.95	634.05	1895185.55
8									695.12	1863047.00
9									743.38	1842268.50

TABLE 4.30: Recovery costs for B10

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	680.32	27801666.90	698.92	29277617.40	662.26	30880115.20	632.24	32264912.40	626.16	33591877.90
3	1036.88	23853571.80	965.17	24754852.50	812.42	25980681.40	779.30	28250452.90	729.70	31322858.70
4	1926.09	23228899.30	1424.46	23568739.70	1217.90	24164956.00	951.07	25707479.10	837.79	29733758.00
5	3020.53	23204423.40	2320.99	23378170.40	1631.85	23768990.50	1274.82	24604568.40	1002.24	28287643.30
6	4454.31	23194123.50	3207.81	23286845.70	2235.45	23561479.60	1588.33	24075591.80	1110.82	26956915.90
7	6873.89	23193251.40	4243.06	23226305.20	2833.61	23440620.10	1864.85	23797297.80	1230.11	26039934.10
8									1446.79	25402087.10
9									1605.43	24914679.00

TABLE 4.31: Recovery costs for XA02

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	113.78	1171229.15	111.31	1334815.55	111.29	1216205.50	110.80	1301899.30	108.88	1340449.90
3	144.22	2761115.55	135.53	1000075.95	135.40	1047350.75	132.31	1197963.20	127.66	1235154.80
4	178.77	960167.85	157.79	961273.00	152.48	990437.70	147.33	1106952.95	139.76	1225910.65
5	209.46	951385.60	186.26	952280.35	172.91	965662.75	163.54	1049549.60	155.36	1183081.95
6	240.11	945920.00	218.26	948139.45	198.22	957672.05	180.88	998395.25	168.24	1171622.60
7	281.78	942012.45	243.95	930865.80	220.02	948825.90	200.45	983077.95	182.95	1165620.35
8	306.59	942741.95	287.89	943706.20	246.61	948305.25	219.35	974089.65	200.04	1148933.20
9	342.09	942637.15	308.11	944025.75	271.38	949776.15	236.99	962865.85	219.15	1101609.80

TABLE 4.32: Recovery costs for XA04

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	121.33	3238254.90	120.80	3658617.15	117.08	3510652.90	112.76	3594453.60	1108.58	3658617.15
3	161.59	2761115.55	148.72	2786746.85	139.50	2925149.75	131.91	3176432.30	127.19	3344986.65
4	218.27	2701940.70	188.21	2709331.40	162.70	2788719.05	153.18	2985001.65	149.60	3229381.60
5	302.61	2681124.20	237.07	2681525.10	203.73	2717816.05	177.91	2870482.35	162.61	3152530.25
6	398.37	2687766.05	313.66	2689470.85	232.45	2700807.35	201.23	2797044.65	176.15	3099723.30
7	471.45	2690931.70	392.31	2687459.60	280.86	2693575.90	229.36	2767968.85	192.49	3072607.80
8	606.54	2690810.70	487.73	2686595.35	339.22	2689325.65	261.09	2741704.50	214.47	3044983.50
9	777.80	2686589.30	635.89	2686052.60	415.12	2687610.65	305.04	2714653.90	229.84	3017888.10

TABLE 4.33: Recovery costs for XB01

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	348.58	847005.15	348.47	849787.90	342.47	862892.90	342.39	871319.30	337.95	901420.95
3	432.8	725421.65	415.97	728238.30	402.80	756087.85	388.81	782783.65	402.19	831195.45
4	511.03	713992.55	485.08	715207.00	471.99	726870.50	449.01	766864.20	447.50	823874.70
5	579.12	713134.20	562.92	714018.45	534.46	717444.00	491.63	745734.85	488.42	807497.15
6	638.27	713100.45	628.85	713304.20	595.97	714449.40	552.25	733297.90	538.98	800242.95
7	691.74	713175.20	669.90	712487.05	634.39	713976.00	604.70	723671.50	591.60	792193.35
8									621.95	775501.45
9									670.97	757276.10

TABLE 4.34: Recovery costs for XB02

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	707.64	5634358.85	681.90	5771611.25	648.66	5990324.35	627.01	6281565.55	621.21	6713134.85
3	872.77	4781959.65	805.84	4939840.65	798.60	5123820.25	774.19	5343344.75	733.69	5792090.95
4	1230.22	4619381.65	1014.81	4735998.80	933.86	4907514.8	898.16	5140637.05	833.25	5494822.70
5	1651.34	4549887.15	1429.38	4573631.00	1149.39	4737348.40	1026.84	4974376.10	944.49	5314832.50
6	3182.05	4544003.15	1955.57	4549630.60	1327.98	4623496.25	1157.29	4888443.00	1047.05	5231072.10
7	4248.98	4541336.80	3181.56	4548314.90	1664.82	4574093.45	1293.14	4781672.50	1154.61	5160560.65
8									1240.36	5114132.40
9									1375.33	5067528.20

TABLE 4.35: Recovery costs for XB03

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	412.68	3415939.05	386.89	3462535.95	383.22	3514060.35	359.32	3615089.30	354.95	3765400.85
3	473.69	2841367.55	466.46	2864382.05	435.44	2903273.85	423.57	3070419.75	407.05	3252203.15
4	648.60	2776712.40	567.94	2781459.90	524.69	2799015.20	498.91	2888605.00	467.65	3077695.15
5	860.62	2775763.85	738.71	2772648.15	636.98	2780220.75	585.74	2841479.30	514.72	2971355.80
6	936.81	2777111.95	872.34	2773727.85	784.52	2778027.30	625.92	2811082.42	563.79	2938162.65
7	1089.88	2773352.30	971.36	2771906.65	839.08	2773426.25	699.28	2795163.25	624.14	2909122.05
8									699.87	2890565.40
9									738.21	2866753.85

TABLE 4.36: Recovery costs for XB04

nb Iter.	100%		80%		60%		40%		20%	
	time	cost	time	cost	time	cost	time	cost	time	cost
2	714.44	28012002.95	663.52	29588755.50	655.12	31091479.20	636.56	32665328.00	629.90	33816892.80
3	1100.38	24395919.30	910.24	25210461.60	846.50	26227917.05	805.05	28865450.60	734.95	31844381.70
4	1733.72	23901805.00	1402.24	24143591.05	1033.40	24733086.85	956.07	26684050.10	849.38	30287567.70
5	2593.57	23804543.95	2128.99	23885407.40	1634.85	24256126.65	1172.68	24397881.00	947.65	28936143.10
6	4009.65	23819907.50	2991.37	23851537.45	1996.19	24148471.30	1447.26	24773325.45	1021.70	27893406.00
7	5589.66	23826171.55	4004.54	23823390.70	2714.57	24063688.20	1761.92	24494917.85	1214.41	26934096.60
8									1357.92	26231866.30
9									1502.76	25645492.00

Chapitre 5

A Rolling Horizon Heuristic for Aircraft and Passenger Recovery

Information sur le chapitre

Un rapport technique basé sur ce chapitre a été publié au CIRRELT, CIRRELT-2016-05. K. Sinclair, J.-F. Cordeau, G. Laporte. A Rolling Horizon Heuristic for Aircraft and Passenger Recovery.

Abstract

In the commercial airline industry, disruptions such as aircraft breakdowns, cancelled or delayed flights and airport capacity reductions can have a significant impact on the airline's planned operations and expenses. When disruptions occur, airlines must create new schedules so as to minimize the recovery cost and revert to the planned schedule as quickly as possible. To solve the joint aircraft and passenger recovery problem, we present a rolling horizon framework incorporating a column generation heuristic and show that the framework performs very well in solving the 2009 ROADEF Challenge instances, obtaining 12 of the best known solutions within 10 minutes of computing time. We also introduce new very large-scale instances and show that the rolling horizon framework is very effective in solving these instances within a reasonable computing time.

Keywords : airline recovery, fleet assignment, aircraft routing, passenger itineraries, large neighbourhood search.

5.1 Introduction

The commercial airline industry is characterized by increased revenues and low profit margins. Between 1995 and 2014, system passenger revenues in the United States have almost doubled, reaching 126 billion dollars in 2014, while the average annual net income between 2000 and 2014 was -2.47 billion [1]. To optimize the use of the necessary resources, operations research models and techniques have become widely used for schedule planning [14]. However, disruptions such as flight delays, flight cancellations, aircraft breakdowns, crew connection problems and unfavorable weather conditions can have a significant impact on the airline's planned operations and expenses. This is in great part due to the use of hub-and-spoke networks by most airlines, which causes small disruptions to have major repercussions on the network. When such disruptions occur, the airlines need to modify their planned schedule so as to minimize the total cost of recovery and return to the planned schedule by the end of the recovery period. Moreover, the recovery problem must be solved very quickly, usually within a few minutes, which increases the difficulty of solving large realistic instances. Because of the size and complexity of the problems considered, recovery problems are usually solved in a sequential manner. The first subproblem is the aircraft recovery problem which aims at creating new aircraft routes by cancelling or delaying flights, as well as modifying aircraft rotations. The crew recovery problem is then solved by reassigning crews, using either deadheading or reserve crews. Finally, the passenger recovery problem is solved by creating new passenger itineraries.

Teodorovic et al. [62] used a branch-and-bound heuristic to obtain optimal solutions to the homogenous fleet aircraft recovery problem, but no realistic instances were solved. The same problem was also solved by Jarrah et al. [39] and Arguello et al. [10]. The former developed minimum-cost network models which they solved by shortest path algorithms, while the latter developed a greedy randomized adaptive search procedure (GRASP) which selects candidate solutions and examines neighbouring solutions before proceeding with a local search phase. Eggenberg et al. [30] used a dynamic programming algorithm to generate the network model and applied a column generation heuristic to solve the heterogeneous fleet aircraft recovery problem. Cao et al. [22, 23] modeled the problem as a quadratic programming problem and applied the approximate linear programming algorithm proposed by Hulburt [27], while Rosenberg et al. [52] modeled it as a set packing problem and reduced the size of the integer program by applying an aircraft selection heuristic. Dozic et al. [29] developed a three-phase heuristic, which removes flight legs on disrupted aircraft rotations,

attempts to reinserts them in other rotations and interchanges parts of rotations between different aircraft. Finally, Xiuli et al. [64] developed a hybrid heuristic combining a tabu search heuristic and a GRASP.

Solving the aircraft recovery problem leads to additional disruptions in the crew scheduling and therefore the crew recovery problem is solved next. Stojković et al. [61] decomposed the problem as a set partitioning master problem, which they solved using a column generation method and a shortest path subproblem. Lettowsky et al. [41] modeled the problem as a set covering problem. The linear programming relaxation of the problem is solved first, and integer solutions are then found by branch-and-bound. Medard et al. [45] also modeled the problem as a set covering problem but used a greedy enumeration with a depth-first tree search approach and a column generation method. Other methods have been developed to solve the crew recovery problem; see, e.g., Abdelghany et al. [4], Nissen et al. [47] and Yu et al. [67].

Once the aircraft and crew recovery problems have been solved, the passenger recovery problem is usually addressed. To solve this problem, Bratu et al. [21] developed two models to determine the delayed flight legs and the cancelled flight legs, and applied a procedure to eliminate the dominated flight arc copies. Zhang et al. [70] developed a non-linear integer program that allows alternate transportation modes and developed an integrality constraint relaxation algorithm.

The decomposition of the recovery problem into subproblems can lead to suboptimal solutions. Therefore, recent research has focussed on integrating at least two of the three recovery problems. For the joint crew and aircraft recovery problem, Luo et al. [43] developed two models and used linear relaxation and a heuristic based on a restricted version of the model. Stojković et al. [61] presented the problem as a linear programming model and showed that the dual problem is a network flow problem that can be solved in linear time. Finally, Abdelghany et al. [5] developed a rolling horizon modeling framework that integrates a simulation model and a mixed integer programming model. The joint aircraft and passenger recovery problem has also been addressed. Zergodi et al. [68] developed a model for the aircraft recovery problem that considers the passenger delay and cancellation costs in the objective function and applied an ant colony algorithm, while Jafari et al. [36, 37] modeled the problem as a mixed integer program using aircraft rotations and passenger itineraries instead of flight legs. Finally, for the aircraft, crew and passenger recovery problem, Peterson

et al. [50] applied a Benders decomposition scheme with the flight scheduling problem as the master problem and the recovery problems as the subproblems.

This paper presents a rolling horizon framework for the joint aircraft and passenger recovery problem as defined in the context of the 2009 ROADEF Challenge (Palpant et al. [48]). This problem consists of creating new aircraft routes and passenger itineraries so as to minimize the total cost and return to the planned schedule by the end of the recovery period. The different methodologies developed during the Challenge can be found on the website <http://challenge.roadef.org/2009>. The following four methods produced the best solution for at least one instance. Bisailon et al. [19], who ranked first in the Challenge, developed a large neighbourhood search heuristic that alternates between a construction, a repair and an improvement phase. Mansi et al. [44] proposed a two-phase algorithm that initially attempts to find a feasible solution using mixed integer programming and a repair heuristic, and then used an oscillation strategy alternating between a construction and a destruction phase. Peekstock et al. [49] developed a simulated annealing algorithm that handles aircraft, airport and passenger infeasibilities by introducing a penalty term in the objective function. Finally, Jozefowiez et al. [40] proposed a three-phase algorithm. The first phase integrates the schedule disruptions by removing flight legs and cancelling itineraries, while the second phase attempts to accommodate disrupted itineraries by assigning them to existing flight legs, and the third phase attempts to add new flight legs to the aircraft rotations.

After the Challenge, Acuna-Agost [6] developed a post-optimization procedure combined with the algorithm proposed by Jozefowiez et al. [40]. It uses network pruning algorithms to reduce the number of variables and constraints in the minimum cost multi-commodity flow model and proves capable of significantly improving solutions within a reasonable computing time. Sinclair et al. [56] proposed a large neighbourhood search heuristic (LNS) based on the Bisailon et al. [19] algorithm, which, among other things, destroys and creates aircraft rotations so as to diversify the search. The proposed heuristic significantly improves the solution cost for 21 of the 22 ROADEF instances. Sinclair et al. [57] also developed a post-optimization algorithm which, when combined with the LNS heuristic, improves the solution cost for all instances of the ROADEF Challenge; it solves a mixed integer programming model and applies a column generation heuristic.

Whereas rolling horizon schemes have been applied to several fields of operations management (see Chand et al. [25] for an extensive review), their application to the passenger

transportation literature is limited. Rolling horizon planning considers only a portion of the complete planning horizon at each step. After solving the problem for a given portion, only part of the decisions are fixed, while the others are optimized at the next step. This method is highly appropriate when the available information is limited, as in passenger rail transportation, where the origin-destination pairs of the passengers are unknown. Nielsen et al. [46] addressed a disruption management problem in the railway rolling stock problem and solved a multi-commodity flow problem using a rolling horizon framework which reduces the computation time to a few seconds, even for large instances. Tornquist [63] applied a similar approach to the railway traffic disturbance problem.

When solving the airline recovery problem, all the necessary information is available at the beginning of the planning horizon. However, the use of a rolling horizon framework can still be appropriate in this context because of the complexity and the size of realistic instances, along with the limited computing time available. Zhang et al. [69] modeled the aircraft, crew and passenger recovery problem as a set partitioning problem and solved it by means of a rolling horizon based algorithm, but the size of the instances considered was limited. As mentioned previously, Abdelghany et al. [5] developed a rolling horizon framework to solve the joint aircraft and crew recovery problem. For the joint aircraft and passenger recovery problem, Jafari et al. [36] presented an assignment model that considers flight delays, aircraft swapping, ferrying, the use of reserve aircraft and passenger cancellation or reassignment. The model was embedded within a rolling horizon framework but the size of the instances considered was limited. Saddoune et al. [54] developed a rolling horizon approach based on column generation to solve the crew pairing problem. They showed that the rolling horizon framework outperforms the three-phase approach traditionally applied in the industry. Furini et al. [33] proposed a rolling horizon approach that integrates a tabu search heuristic for the aircraft sequencing problem. Two mixed integer models were presented and rules for partitioning the aircraft sequence were proposed.

The first contribution of this article is to develop a rolling horizon scheme that allows the solution of large instances within a reasonable computing time. There exists a rich literature on airline recovery problems, but most research has focused on either small instances with realistic disruptions, or on large instances with very few disruptions. The second contribution is the creation of new very large realistic instances that take into consideration major disruptions on airport arrival and departure capacity.

The aircraft and passenger recovery problem is described in Section 2, followed by the solution methodology in Section 3. Section 4 presents the computational results, followed by conclusions and paths for future research in Section 5.

5.2 Problem description

The aircraft and passenger recovery problem aims at creating alternative aircraft routes and passenger itineraries when the airline schedules become infeasible due to disruptions. Airlines must construct these new aircraft routes and passenger itineraries in a timely manner so as to return to the original schedule by the end of the recovery period, while satisfying numerous constraints. The problem parameters are now described.

5.2.1 Airports

The airports are represented by the set of nodes $N = \{1, \dots, n\}$, where each node corresponds to an airport at a given time. For each 60-minute time period beginning at hour p , each airport i has given arrival and departure capacities defined as a_{ip} and b_{ip} , respectively.

5.2.2 Aircraft

Each aircraft f is characterized by an identification number, a family, a model and a cabin configuration. The fleet of aircraft operated by the airline is denoted by F . Aircraft belonging to the same family, for example AirbusSmall or BoeingSmall, have interchangeable crews. For each aircraft family there is a number of different models, for example models B747 and B777 belong to the BoeingBig family, and all aircraft of the same model have the same turn-round time, transit time and range. The cabin configuration is considered fixed and defines the capacity, denoted as cap_{mf} , of each cabin category m (economy, business and first) of aircraft f , with M being the set of all cabin classes. All aircraft of the same family, model and cabin configuration belong to the same group and let $\beta(f)$ be the index of the group to which aircraft f belongs. The aircraft can also have a planned maintenance, where they will become unavailable for a predetermined period of time. In such a case, the aircraft group $\beta(f)$ contains only one aircraft and is also characterized by the maintenance airport, the maintenance period and the number of flight hours before the maintenance, denoted by t_f . The demand for aircraft group $\beta(f)$ at node i is denoted by $d_{i\beta(f)}$.

5.2.3 Flight legs

Each aircraft f is assigned a rotation consisting of a sequence of flight legs that must satisfy the rotation continuity constraints and the turn-round time constraints. Flight legs are represented by the set of arcs $A = \{(i, j); i, j \in N, i \neq j\}$, where each arc (i, j) is a flight leg between nodes i and j or a connection arc between two nodes at the same airport. Arc (i, j) can also represent a ground transportation arc between two different airports within the same urban region. The set $A_{\beta(f)} = \{(i, j); i, j \in N, i \neq j\}$, is the set of arcs (i, j) associated with aircraft group $\beta(f)$. Flight legs are characterized by a travel time t_{ij} from i to j , and by the type of flight (domestic, continental or intercontinental). The set of arcs (i, j) arriving at node j during period p is defined as $O_{j(p)}$ while the set of arcs (i, j) departing from node i during period p is defined as $I_{i(p)}$. The set of aircraft F is partitioned into families, models and configurations. We define B_l as the set of all aircraft of family l in the set L of aircraft families, T_g as the set of all aircraft model g in the set G of aircraft models and V_h as the set of all aircraft configuration h in the set H of aircraft configurations.

5.2.4 Passenger reservations

The set of passenger reservations is denoted by $S = \{1, \dots, s\}$, where each passenger reservation is characterized by the reservation number, the type of itinerary (inbound or outbound), the average price, the number of passengers and the itinerary, which consists of the sequence of flight legs that must satisfy the continuity constraints, the minimum connection time constraints and the cabin class constraint for each flight leg. The passenger demand for itinerary k is defined as d_k . For each itinerary k , we denote by D_k the set of all arcs (i, j) where i is the departure node of itinerary k , by U_k the set of all arcs (i, j) where i is the arrival node of itinerary k and by W_k the set of all dummy arcs (i, j) , which represent the cancellation arcs between the departure and the arrival nodes for itinerary k . Passenger itineraries must also satisfy additional functional constraints. The first are the delay constraints, which ensure that the maximum delay does not exceed 18 hours for domestic and continental itineraries and 36 hours for international itineraries. These constraints do not apply to return itineraries. The second are the origin-destination constraints which ensure that passengers on a reconstructed itinerary depart from the original departure airport and arrive at the original destination airport. Finally, passengers on a reconstructed itinerary cannot depart before the original departure time.

5.2.5 Disruptions

Four types of disruptions are considered : flight delays, flight cancellations, airport disruptions and aircraft disruptions. Airport disruptions limit the number of departures and arrivals at a given airport for a given time period, whereas aircraft disruptions lead to the unavailability of an aircraft for a period of time. In the latter case, no flight legs can be assigned to the aircraft during the disruption period.

5.2.6 Problem formulation

The problem can be represented on a time-space network $G = (N, A)$, where N is the set of nodes and A is the set of arcs previously defined. The objective is to minimize the total recovery cost, which consists of delay costs, cancellation costs, downgrading costs, flight operating costs and costs for non-compliant location of aircraft. There are two types of delay costs, direct costs and passenger disutility costs. The former consist of the disbursements the airline incurs in case of delays (lodging, drinks and meal expenses) and vary depending on the extent of the delay and the initial planned duration. The passenger disutility costs measure the passenger's perceived disturbance. It depends on the extent of the delay, the reservation type and the itinerary's reference cabin class. The delay costs are denoted by c_{ijmk}^{del} . The cancellation cost c_k^{can} of itinerary k is the reimbursement of the ticket price as well as a financial compensation that varies depending on the initially planned duration of the itinerary. Downgrading costs c_{km}^{down} are also a measure of the disutility of passengers and are incurred when the cabin class of one or more flight legs of the new itinerary is lower than the itinerary's reference cabin class. These costs depend on the type of flight leg and on the level of downgrading. The operating costs c_{ijf}^{op} depend on the travel time t_{ij} from node i to node j . They are considered to be independent of the number of passengers and of the possible delays. Finally, costs for non-compliant location of aircraft are incurred when the required number of aircraft is not available at the specified airports by the end of the recovery period.

There are three types of non-compliance penalties. A penalty cost c^{fam} is incurred whenever a required aircraft cannot be matched with an aircraft of the same family, whereas a penalty cost c^{mod} is incurred if a required aircraft cannot be matched with an aircraft of the same model. Finally, a penalty cost c^{conf} is incurred if a cabin configuration cannot be matched. Parameters n_j^l , n_j^g and n_j^h are the number of aircraft of family l , the number of aircraft of model g and the number of aircraft of configuration h requested at node j at the end of the recovery period. D^{fam} , D^{mod} and D^{conf} are duplicates of the final node at each

airport and represent, respectively, the family demand node, the model demand node and the configuration demand node, and the sets R_l , R_g and R_h are, respectively, the sets of all arcs between these nodes and a dummy node.

5.3 Solution methodology and mathematical model

In order to solve the joint aircraft and passenger recovery problem we have developped a rolling horizon variant of the column generation post-optimization heuristic described by Sinclair et al. [57]. Using the column generation post-optimization heuristic proves time consuming for large instances. Therefore, we have embedded the column generation heuristic within a rolling horizon framework, thus reducing the computing time and memory needed. We will first summarize the column generation heuristic developped by Sinclair et al. [57], and then describe the rolling horizon framework.

5.3.1 Column generation post-optimization heuristic

The column generation post-optimization heuristic is initialized with the solution obtained by applying the LNS heuristic developped by Sinclair et al. [56]. This solution is included in the initial restricted master problem of the column generation heuristic.

Large neighbourhood search heuristic

The LNS heuristic of Sinclair et al. [56] is an improvement to the LNS heuristic developped by Bessaillon et al. [19] for the ROADEF 2009 Challenge. The heuristic alternates between three phases : construction, repair and improvement.

Construction phase

In the first phase, the heuristic attempts to obtain an initial feasible solution by delaying or cancelling flights. If a rotation becomes infeasible because of delays on previous flights in the rotation, the first flight yielding an infeasibility is delayed by increments of 60 minutes. If no feasible rotation can be found, the flight is declared critical and will be treated in the repair phase. This phase also considers rotation infeasibility due to a cancelled flight in the rotation. The heuristic attempts to recreate the cancelled flight. If this proves impossible, the smallest sequence of flights that forms a loop, starting with the flight that yields the infeasibility, is removed. If no such loop exists, the flight yielding the infeasibility and all the subsequent flights in the rotation are cancelled. Rotation infeasibility due to maintenance is treated by

removing a loop before the maintenance that allows the rotation to respect the maintenance constraints. Flights that are cancelled because of airport capacity are rescheduled at a later, less congested period. If this is not possible, either the smallest loop is removed or all subsequent flights in the rotation are removed. Finally, when a rotation becomes infeasible due to aircraft breakdowns, the heuristic tries to recreate all the disrupted flights by means of a longest path algorithm.

Repair phase

The first step of the repair phase considers the critical flights obtained in the construction phase because of delays on previous flights, and tries to delay them to a less congested period. If this proves impossible, either the smallest loop containing the flight or the sequence starting from this flight to the end of the rotation is removed. Once aircraft rotation feasibility has been regained, the heuristic attempts to reinsert flight sequences that have been previously removed in the rotations of available aircraft. In the last two steps, the heuristic attempts to reaccommodate passengers whose itineraries have been cancelled, first by repeatedly solving a shortest path problem, then by creating new flights by either solving a multi-commodity flow problem or a shortest path problem.

Improvement phase

First, the improvement phase performs a local search to improve the solution by delaying some flights so as to accommodate additional passengers. The passengers are reassigned by repeatedly solving the shortest path problem. The heuristic then diversifies the search by destroying either part of a rotation or a complete rotation, and by then creating new flights using all of the available aircraft, again by repeatedly solving a shortest path problem. The next step attempts to reaccommodate passengers at disrupted airports (airports with reduced arrival and departure capacity). Passengers are assigned to an existing flight departing from the disrupted airports and the heuristic attempts to create new flights from the flight's arrival airport to the passengers's destination airport. In the opposite way, the heuristic also tries to accommodate passengers whose final destination is a disrupted airport. Because of the maximum delay constraints, it is sometimes impossible to accommodate passengers with a cancelled itinerary on a certain flight leg. The heuristic attempts to delay subsequent passengers who have the same origin-destination pair, thus accommodating additional passengers. Finally, new flights are created by means of a shortest path algorithm so as to redirect the aircraft to the required airports at the end of the recovery period.

Column generation heuristic

Sinclair et al. [57] developed the following mixed integer programming model for the aircraft and passenger recovery problem (APRP), where $x_{ijf} = 1$ if and only if arc (i, j) is operated by aircraft f , and y_{ijmk} is the number of passengers from itinerary k assigned to arc (i, j) in class m .

(APRP)

$$\text{minimize } \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{km}^{down} y_{ijmk} \quad (5.1)$$

$$+ \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{ijmk}^{del} y_{ijmk} \quad (5.2)$$

$$+ \sum_{(i,j) \in W_k} \sum_{m \in M} \sum_{k \in K} c_k^{can} y_{ijmk} \quad (5.3)$$

$$+ \sum_{(i,j) \in A} \sum_{f \in F} c_{ijf}^{op} x_{ijf} \quad (5.4)$$

$$+ \sum_{f \in B^l} \sum_{(i,j) \in R^l} c^{fam} x_{ijf} \quad (5.5)$$

$$+ \sum_{f \in T^g} \sum_{(i,j) \in R^g} c^{mod} x_{ijf} \quad (5.6)$$

$$+ \sum_{f \in V^h} \sum_{(i,j) \in R^h} c^{conf} x_{ijf} \quad (5.7)$$

subject to

$$\sum_{i \in N, (i,j) \in A_{\beta(f)}} x_{ijf} - \sum_{i \in N, (j,i) \in A_{\beta(f)}} x_{jif} = 0 \quad \beta(f) \in F, j \in N \text{ and } d_{j\beta(f)} = 0 \quad (5.8)$$

$$\sum_{j \in N, (i,j) \in A_{\beta(f)}} x_{ijf} \leq 1 \quad \beta(f) \in F, i \in N \text{ and } d_{i\beta(f)} = 1 \quad (5.9)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^l} x_{jif} \leq n_j^l \quad f \in F, j \in N, l \in L \text{ and } j = D^{fam} \quad (5.10)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^g} x_{jif} \leq n_j^g \quad f \in F, j \in N, g \in G \text{ and } j = D^{mod} \quad (5.11)$$

$$\sum_{i \in N} x_{ijf} - \sum_{i \in R^h} x_{jif} \leq n_j^h \quad f \in F, j \in N, h \in H \text{ and } j = D^{conf} \quad (5.12)$$

$$\sum_{(i,j) \in I_j(p)} \sum_{f \in \beta(f)} \sum_{\beta(f) \in F} x_{ijf} \leq a_{jp} \quad p \in P, j \in N \quad (5.13)$$

$$\sum_{(i,j) \in O_i(p)} \sum_{f \in \beta(f)} \sum_{\beta(f) \in F} x_{ijf} \leq b_{ip} \quad p \in P, i \in N \quad (5.14)$$

$$\sum_{f \in F} x_{ijf} \leq 1 \quad (i, j) \in A \quad (5.15)$$

$$\sum_{(i,j) \notin D_k U_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k U_k} \sum_{m \in M} y_{jimk} = 0 \quad k \in K, j \in N \quad (5.16)$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \quad k \in K \quad (5.17)$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{mf} x_{ijf} \quad (i, j) \in A, m \in M \quad (5.18)$$

$$\sum_{(i,j) \in A} t_{ij} x_{ijf} \leq t_f \quad f \in F \quad (5.19)$$

$$y_{ijmk} \geq 0 \quad (i, j) \in A, m \in M, k \in K \quad (5.20)$$

$$x_{ijf} = 0 \text{ or } 1 \quad (i, j) \in A, f \in F. \quad (5.21)$$

The terms (5.1)–(5.3) of the objective function define, respectively, the downgrading costs, the delay costs and the cancellation costs, while term (5.4) defines the aircraft operating costs. The terms (5.5)–(5.7) are, respectively, the non-compliance costs for the incorrect family, model and configuration positionning. Constraints (5.8) and (5.9) ensure that the aircraft flow conservation constraints are respected, while constraints (5.10)–(5.12) are the aircraft flow conservation constraints for the family, the model and the configuration, respectively. Constraints (5.13) and (5.14) enforce the airport capacity limit. Constraints (15) are the assignment constraints which ensure that each flight arc is assigned to at most one aircraft. Constraints (5.16) and (5.17) enforce passenger flow conservation. Finally, Constraints (5.18) ensure that the aircraft seating capacity limits are respected, and constraints (5.19) ensure that each flight does not exceed the maximum flight hours.

The initial restricted master problem (RMP) includes all aircraft and passenger arcs found in the LNS solution. To solve the problem by column generation, the linear programming (LP) relaxation of the problem is considered and the dual variables π^{ik} , λ^k and $\mu^{ijm} \leq 0$ are associated with constraints (5.16)–(5.18). At each iteration, a percentage of variables with negative reduced cost are included in the LP relaxation and this phase is repeated a predetermined number of iterations. Then the APRP problem with a limited number of variables is solved.

In order to solve the larger instances, the size of the model is reduced by considering all passengers with the same origin-destination pair, the same departure time and the same cabin class as a passenger group. Also, all aircraft with the same characteristics are aggregated except for aircraft with planned maintenance. Finally, for some instances, only a subset of itineraries are considered. Since solving the APRP problem for the large instances is impractical, Sinclair et al. [57] use the column generation heuristic just described to solve the following multi-commodity flow problem for passengers (MC-APRP), where $X^* = \{(i, j, f) : (i, j) \in A, f \in F \text{ and } x_{ijf} = 1\}$ is the set of the aircraft arcs found in the LNS solution.

(MC-APRP)

$$\text{minimize } \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{km}^{down} y_{ijmk} \quad (5.22)$$

$$+ \sum_{(i,j) \in A} \sum_{m \in M} \sum_{k \in K} c_{ijmk}^{del} y_{ijmk} \quad (5.23)$$

$$+ \sum_{(i,j) \in W_k} \sum_{m \in M} \sum_{k \in K} c_k^{can} y_{ijmk} \quad (5.24)$$

subject to

$$x_{ijf} = 1 \quad (i, j, f) \in X^* \quad (5.25)$$

$$\sum_{(i,j) \notin D_k U_k} \sum_{m \in M} y_{ijmk} - \sum_{(j,i) \notin D_k U_k} \sum_{m \in M} y_{jimk} = 0 \quad k \in K, j \in N \quad (5.26)$$

$$\sum_{(i,j) \in D_k} \sum_{m \in M} y_{ijmk} = d_k \quad k \in K \quad (5.27)$$

$$\sum_{k \in K} y_{ijmk} \leq \sum_{f \in F} cap_{fm} x_{ijf} \quad (i, j) \in A, m \in M \quad (5.28)$$

$$y_{ijmk} \geq 0 \quad (i, j) \in A, m \in M, k \in K \quad (5.29)$$

$$x_{ijf} = 0 \text{ or } 1 \quad (i, j) \in A, f \in F. \quad (5.30)$$

The terms (5.22)–(5.24) are, respectively the downgrading costs, the delay costs and the cancellation costs. Constraints (5.25) are the assignment constraints, which ensure that all aircraft arcs found in the LNS solution take value 1. Constraints (5.26) and (5.27) are the passenger flow conservation constraints, and constraints (5.28) enforce the aircraft seating capacity limit.

5.3.2 Rolling horizon

The column generation heuristic described in the previous section is embedded in a rolling horizon framework depicted in Figure 5.1. At iteration t , the recovery period is divided into rolling periods Δ_t which are in turn divided in a fixed and non-fixed part (Δ_t' and Δ_t'' , respectively). At the first iteration, all itineraries and all flights departing within the rolling period Δ_1 are considered. Since many passenger itineraries contain more than one flight, limiting the flights considered to those departing before the end of the rolling horizon period would limit the number of possibilities for the itineraries with departures close to the end of the rolling horizon period. Therefore, in order to accommodate additional passengers, flights departing within an additional time period Θ are also considered.

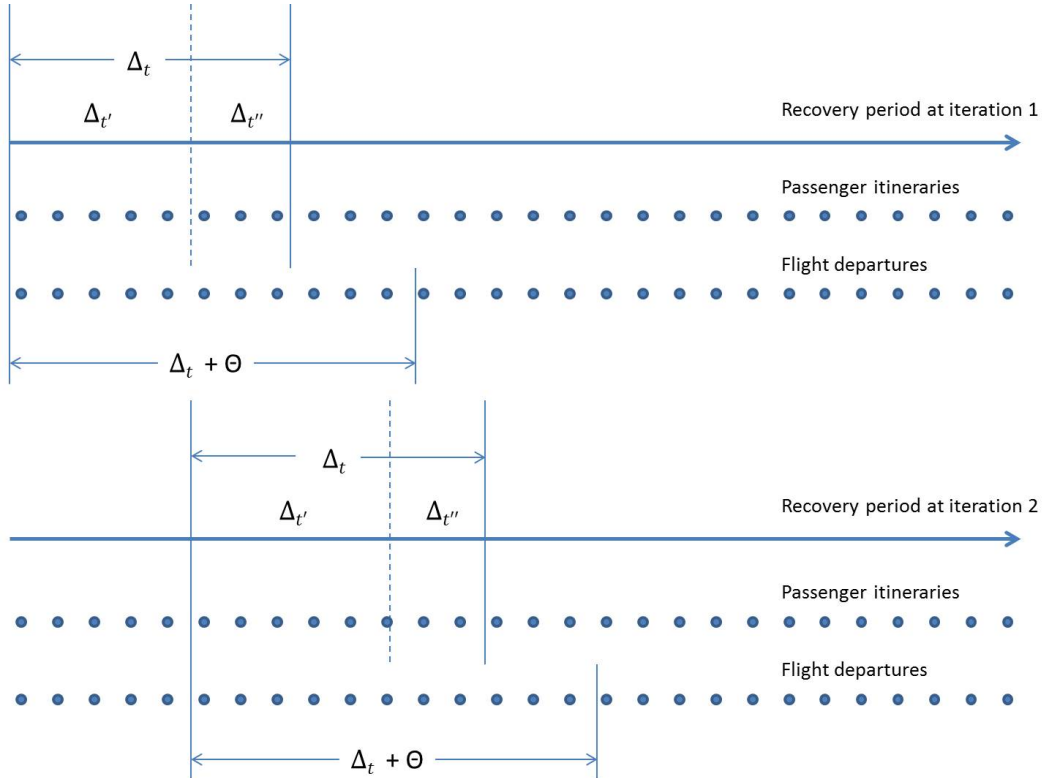


FIGURE 5.1: Rolling horizon framework

The column generation heuristic is applied in each rolling horizon period and the LP relaxation of the APRP problem is solved for a fixed number of iterations, denoted by α , before solving the APRP problem at the final iteration $\alpha + 1$. At each iteration, a percentage of the variables with negative reduced costs is included in the LP relaxation. The itineraries obtained in the column generation heuristic’s solution that begin within the fixed part of the rolling period Δ'_t , are fixed. At iteration, $t + 1$, all itineraries from the previous rolling period that remain cancelled are considered in the following rolling period Δ_{t+1} , which includes the non-fixed part of the previous rolling period Δ_t , and the column generation heuristic is applied again. The rolling horizon framework ends when the end of the rolling period is equal to or greater than the end of the recovery period.

5.4 Computational results

The rolling horizon heuristic was implemented and tested on a computer with two Intel Westmere EP X5650 six-core processors running at 2.667 GHz and with 96 GB of memory. Our algorithm was tested on the instances of the 2009 ROADEF Challenge, described in Tables 5.1–5.5, as well as on some new, very large instances derived from the 2009 ROADEF Challenge instances.

5.4.1 The 2009 ROADEF Challenge Instances

To test our methodology we used the instances provided in the 2009 ROADEF Challenge. Tables 5.1 and 5.2 present the disruption characteristics of the B instances. For these instances, the recovery period is 36 hours. There are 256 aircraft, 45 airports, 1,423 flights and 11,214 passenger itineraries.

TABLE 5.1: Characteristics the B01–B05 instances

	B01	B02	B03	B04	B05
Flight disruptions	230	255	229	230	0
Aircraft disruptions	0	0	1	0	0
Airport disruptions	0	0	0	1	34

TABLE 5.2: Characteristics the B06–B10 instances

	B06	B07	B08	B09	B10
Flight disruptions	230	255	229	230	77
Aircraft disruptions	0	0	1	0	0
Airport disruptions	0	0	0	1	34

Tables 5.3–5.5 present the characteristics of the XA, XB and X instances. The smaller instances, the XA instances, contain 85 aircraft, 35 airports and 608 flights. The XB instances have 256 aircraft, 44 or 45 airports and 1,423 flights, while the larger instances, the X instances, contain 618 aircraft, 168 airports and 2,178 flights.

TABLE 5.3: Characteristics of the XA instances

	XA01	XA02	XA03	XA04
Recovery period (h)	14	52	14	52
Itineraries	1,943	3,959	1,872	3,773
Flight disruptions	83	0	83	
Aircraft disruptions	3	3	3	3
Airport disruptions	0	407	0	407

TABLE 5.4: Characteristics of the XB instances

	XB01	XB02	XB03	XB04
Recovery period (h)	36	52	36	52
Itineraries	11,214	11,214	11,565	11,565
Flight disruptions	229	0	228	0
Aircraft disruptions	3	1	4	4
Airport disruptions	0	34	0	34

TABLE 5.5: Characteristics of the X instances

	X01	X02	X03	X04
Recovery period (h)	78	78	78	78
Itineraries	28,308	28,308	29,151	29,151
Flight disruptions	0	0	0	0
Aircraft disruptions	1	1	1	1
Airport disruptions	1	0	1	0

5.4.2 Creation of very large instances derived from the 2009 ROA-DEF Challenge instances

As can be seen in Tables 5.1–5.5, four instances from the Challenge are very large (i.e. instances X01, X02, X03 and X04), with a recovery period of 78 hours, 618 aircraft, 168 airports, 2,178 flights and a number of passenger itineraries varying between 28,308 and 29,151. Although the size of these instances is realistic, they contain very few disruptions (no flight disruptions, one aircraft disruption and either no or one airport disruption). We have therefore created new instances with numerous disruptions, based on these large instances. Since instance X02 is identical to X01 and X04 is identical to X03 except for the disruptions, only instances X01 and X03 were used to create the new instances.

We reduced the recovery period to 55 hours so as to be able to create flight disruptions during the first day of the schedule. The disruption scenarios created are similar to those in the Challenge in terms of flight delays, flight cancellations, aircraft disruptions and airport disruptions. We used four scenarios of flight disruptions (no disruption, 111 disruptions, 161 disruptions and 215 disruptions, noted -F, 111F, 161F and 215F respectively), four scenarios of airport disruptions (100% airport capacity, 75% airport capacity, 50% airport capacity and 30% airport capacity, noted -P, 75P, 50P and 30P respectively) and three scenarios of aircraft disruptions (one disruption, two disruptions and four disruptions, noted 1C, 2C and 4C respectively).

We created 48 instances from instance X01 and 48 instances from instance X03 using all combinations of the disruption scenarios described above. All instances can be found at <http://chairelogistique.hec.ca/data/airline/> and are identified as follows : X0175P2C111F is the instance X01 with the following disruption combination : 75% airport capacity, two aircraft disruptions and 111 flight disruptions. A description of all instances is presented in Appendix 1.

5.4.3 Parameter setting

To calibrate the different parameters we used a total of 24 instances : the 12 instances derived from the X01 instance with 30% airport arrival and departure capacity, referred to as the X0130P instances, and the 12 instances derived from the X03 instance with 75% airport arrival and departure capacity, referred to as the X0375P instances.

5.4.3.1 Number of iterations and percentage of variables with negative reduced cost For the parameter setting, the number of iterations performed within the column generation heuristic and the percentage of variables included in the LP relaxation are considered simultaneously. The number of iterations considered varies between three and six, and the percentage of variables varies between 40% and 100%. Table 5.6 presents the average improvement with respect to the best solution as well as the average computing time for all possible combinations of these parameters for the X0130P instances and the X0375P instances. The solutions obtained with 4 iterations and 100% of the variables with negative reduced cost included in the LP relaxation offer a good compromise between solution quality and computing time.

TABLE 5.6: Average solution improvement and computing times

	3 iterations		4 iterations		5 iterations		6 iterations	
	% improvement	time	% improvement	time	% improvement	time	% improvement	time
40%	16.84%	1965.63	16.72%	2135.50	4.251%	2298.71	2.06%	2481.58
60%	9.09%	1913.24	2.37%	2195.42	1.04%	2380.92	0.83%	2547.25
80%	6.94%	2028.79	1.31%	2262.67	0.54%	2436.50	0.623%	2598.04
100%	3.194%	2090.00	0.86%	2299.75	0.50%	2481.33	0.594%	2654.04

5.4.3.2 Fixed rolling horizon period

The rolling horizon period Δ defines the period for which passenger itineraries are considered in the column generation heuristic at each iteration. All itineraries departing before the end of the rolling horizon period are considered, but only part of the solution's itineraries are fixed. The rolling horizon period can therefore be considered as a fixed part, Δ' , and a non-fixed part, Δ'' . The average solution gap and the average computing time for the different percentages of the rolling horizon that are fixed are reported in Figures 5.2 and 5.3, respectively. The number of iterations within the column generation heuristic was set to four and the percentage of variables with negative reduced cost included in the LP relaxation was set at 100%. The best average computing time and the best average gap with respect to the best solution was obtained when parameter Δ' was set to 80% of Δ .

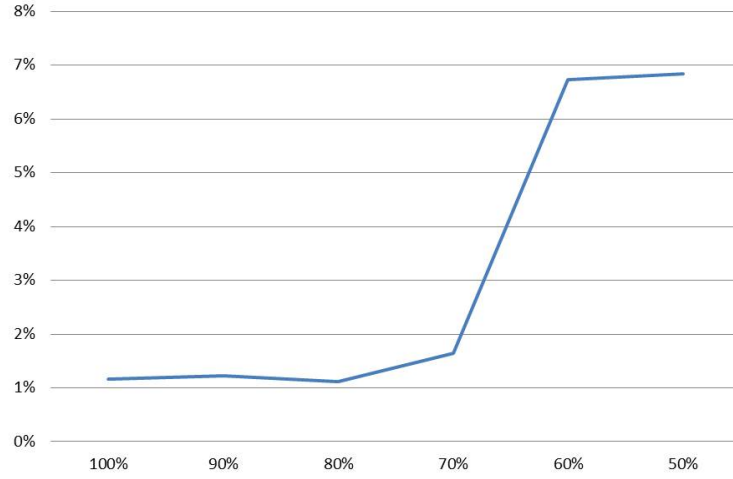


FIGURE 5.2: Improvement with respect to best solution value for parameter Δ'

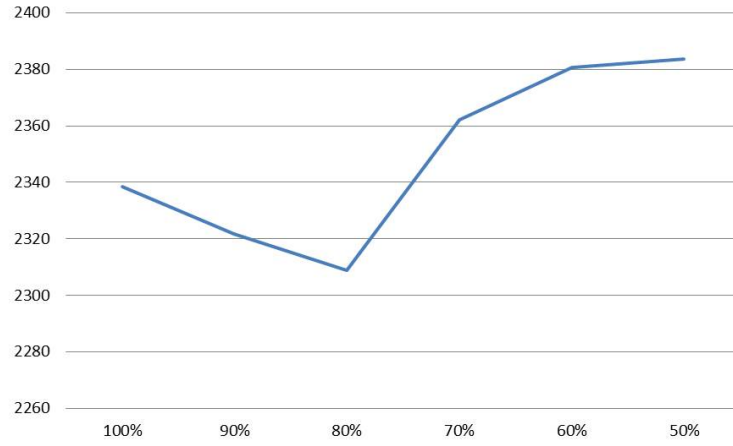


FIGURE 5.3: Computing time in seconds for parameter Δ'

5.4.3.3 Additional time period for flights Θ

The additional time period considered for the flights is defined as Θ . Figure 5.4 presents the average gap with respect to the best solution when the value of Θ varies between 0 and 1200 minutes, while Figure 5.5 presents the average computing time for the same values of Θ . The number of iterations within the column generation heuristic was set to four, the percentage of variables with negative reduced costs included in the LP relaxation was set to 100% and Δ' was set at 80% of Δ . As can be seen in Figure 4, solutions improve when we increase Θ from 0 minutes to 500 minutes. Then, solution quality slowly deteriorates. Figure 5 shows that the computing time decreases when Θ is increased. The value of Θ was therefore set to 500 minutes.

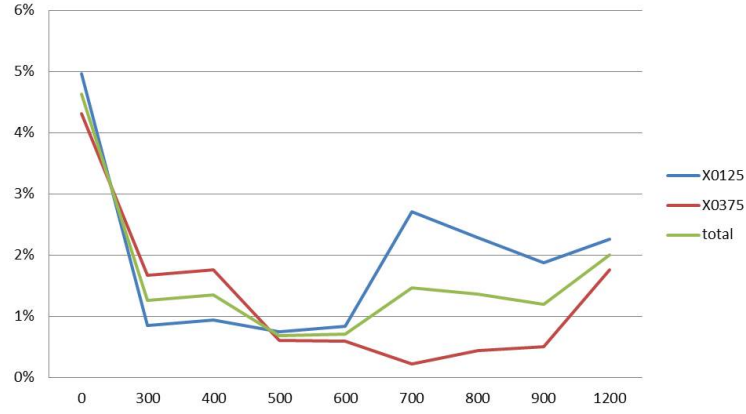


FIGURE 5.4: Improvement with respect to best solution value for parameter Θ

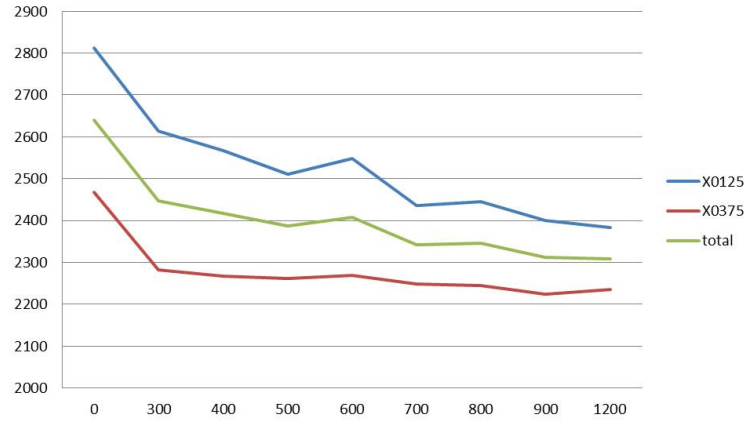


FIGURE 5.5: Computing time in seconds for parameter Θ

5.4.3.4 Length of the rolling horizon

The length of the rolling horizon period Δ has an impact on solution quality as well as on computing time. Tables 5.7–5.9 present the average gap with respect to the LNS solution value obtained after 10 minutes of computing time and 40 minutes of computing time for the X0130P instances and the X0375P instances as well as the average computing time for rolling horizon lengths varying between 900 and 1,500 minutes. Again, the number of iterations within the column generation heuristic was set to four, the percentage of variables with negative reduced costs included in the LP relaxation was set to 100% and Δ' was set at 80% of Δ . Finally, the additional time period for flights Θ was set to 500 minutes. The best solution values are obtained with a rolling horizon period of 1,200 minutes.

TABLE 5.7: Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	900 minutes		
	% improvement wrt LNS 10 min	% improvement wrt LNS 40 min	Seconds
X0130P	9.72%	-2.04%	2378.00
X0375P	9.49%	2.43%	2013.92

TABLE 5.8: Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	1,200 minutes		
	% improvement wrt LNS 10 min	% improvement wrt LNS 40 min	Seconds
X0130P	13.66%	2.43%	2508.75
X0375P	15.66%	9.94%	2271.58

TABLE 5.9: Average solution improvement wrt LNS 10 minutes and LNS 40 minutes

	1,500 minutes		
	% improvement wrt LNS 10 min	% improvement wrt LNS 40 min	Seconds
X0130P	11.51%	-0.01%	2637.33
X0375P	14.98%	9.59%	2393.25

5.4.4 Results

5.4.4.1 New instances

Since the rolling horizon heuristic is a post-optimization heuristic, we have run the LNS heuristic for 10 minutes before executing the rolling horizon heuristic. In order to compare the quality of the solutions obtained with the rolling horizon heuristic for the 96 new instances, we ran the LNS heuristic for the average computing time of the rolling horizon heuristic, which includes the 10 minutes for the LNS heuristic. To this end, we have separated the instances into eight groups : the X01 instances with 100% airport capacity, the X01 instances with 75% airport capacity, the X01 instances with 50% airport capacity and the X01 instances with 30% airport capacity, the X03 instances with 100% airport capacity, the X03 instances with 75% airport capacity, the X03 instances with 50% airport capacity and the X03 instances with 30% airport capacity (denoted by X01-P, X0175P, X0150P, X0130P, X03-P, X0375P, X0350P and X0330P respectively). The rolling horizon length Δ was set to 1200 minutes, Δ' was set to 80% of Δ , Θ was set to 500 minutes, the number of iterations within the column generation heuristic was set to four and 100% of the variables with negative reduced costs were included in the LP relaxation.

TABLE 5.10: Average solution improvement wrt LNS 10 minutes and LNS average time

	Average time	% improvement	% improvement
	seconds	wrt LNS 10 min	wrt LNS average time
X01-P	1913	1.21%	-0.49%
X0130P	2508	13.66%	1.57%
X0150P	2453	13.64%	9.77%
X0175P	2038	8.00%	4.10%
X03-P	1950	1.49%	0.63%
X0330P	3049	7.21%	4.63%
X0350P	2739	13.05%	5.94%
X0375P	2271	15.66%	18.02%

For each group, we ran the LNS heuristic for the average computing time of the rolling horizon heuristic for that group. Table 5.10 presents the average gap with respect to the solution values obtained by means of the LNS heuristic when run for 10 minutes and when ran for the average computing time of the rolling horizon heuristic for the respective group. The solution costs for all instances are reported in Appendix 2. As can be seen from Table 5.8, when there are no airport disruptions, as for the X01-P instances and the X03-P instances, the LNS heuristic performs well, outperforming the rolling horizon heuristic for the X01-P instances. However, the rolling horizon heuristic performs very well for all instances with airport capacity disruptions.

5.4.4.2 The 2009 ROADEF Challenge instances

Additional parameter setting tests were performed for the rolling horizon length parameter for the 2009 ROADEF Challenge instances. Tables 5.11 and 5.12 present the solution values and computing time for the rolling horizon heuristic, as well as solution values when running the LNS heuristic for 10 minutes, and the gap with respect to the LNS solution value.

Table 5.13 presents the best known solution values for the 2009 ROADEF Challenge instances obtained within 10 minutes of computing time as well as the rolling horizon heuristic solution value including the LNS heuristic obtained within 10 minutes of computing time (total time of 10 minutes for the LNS heuristic and rolling horizon heuristic). The rolling horizon length used is 1800 minutes. The heuristic was able to find solutions within 10 minutes for 14 of the 22 instances and obtained 12 best solution values. It is important to note

that the speed of the computer used is faster than the one used during the 2009 ROADEF Challenge. Reducing the computing time to take into consideration the speed of the computer would not yield good feasible solutions.

Table 5.14 presents the solution values obtained with the column generation heuristic of Sinclair et al. [57] as well as the computing time and the improvement with respect to the solution values obtained when running the LNS for 10 minutes. It is important to note that the LNS heuristic was run for 10 minutes before the column generation heuristic was applied. Table 5.14 also presents the solution values obtained with the rolling horizon heuristic and the computing time. The improvements with respect to the LNS solution values and the improvements with respect to the column generation heuristic are also reported in Table 5.14. Solutions found with the column generation outperform the rolling horizon heuristic, which was expected since the column generation heuristic is identical to the rolling horizon heuristic with a horizon length equal to the total recovery time. However the computing time necessary to obtain good solutions is much faster for the rolling horizon heuristic, except for instances X01–X04, which require a longer computing time but yield better solution costs.

5.5 Conclusions and future research

We have developed a rolling horizon framework for the post-optimization column generation heuristic to solve the joint aircraft and passenger recovery problem. The rolling horizon framework has proven very effective on the instances of the 2009 ROADEF Challenge with 10 minutes of computing time, yielding 12 out of 22 best solutions. We have also presented new, very large instances with many disruptions and showed that the rolling horizon framework performs very well on the instances with important airport disruptions, and outperforms the LSN heuristic of Sinclair et al. [56].

Future research should focus on developing useful tools for the airline industry that incorporate the rolling horizon framework by allowing an initial solution to be applied to the aircraft rotations and to the passenger itineraries with the earliest departure times, while solving the passenger recovery problem for the remainder of the passenger itineraries. It should eventually integrate the crew recovery problem to yield large realistic recovery problems.

Acknowledgments This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 2014-04959 and 2015-06189. This support

is gratefully acknowledged.

TABLE 5.11: Rolling horizon heuristic solution costs

LNS 10 min		900 min			1,200 min			1,500 min	
		Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS	Cost	Seconds
B01	870812	699172	305	19.71%	732792	294	15.85%	712561	305
B02	1056280	921709	274	12.74%	880891	294	16.60%	929974	294
B03	931289	846095	286	9.15%	738978	314	20.65%	776523	365
B04	962451	893814	287	7.13%	768812	282	20.12%	817409	315
B05	8377616	5288122	617	36.88%	5402406	702	35.51%	4675439	745
B06	2829589	2402166	309	26.11%	2179034	313	22.99%	2141244	345
B07	3157541	2603379	317	17.55%	2449561	312	22.42%	2495971	365
B08	3134454	2601498	300	17.00%	2394693	321	23.60%	2378236	365
B09	2676733	2464949	318	7.91%	2048043	354	23.49%	2012754	375
B10	32984637	29861715	831	9.47%	26795800	914	18.76%	22707143	935
X01	142443	79753	2883	44.01%	no feasible solution found			no feasible solution found	
X02	-20964	-45867	2834	118.79%	no feasible solution found			no feasible solution found	
X03	1353315	1032698	3041	23.69%	no feasible solution found			no feasible solution found	
X04	145534	82853	2949	43.07%	no feasible solution found			no feasible solution found	
XA01	79711	72280	51	9.32%	72280	51	9.32%	72280	51
XA02	1664971	1483525	106	10.90%	1488596	111	10.59%	1141325	111
XA03	262945	231755	72	11.86%	231755	73	11.86%	231755	73
XA04	4440451	4299784	124	3.17%	4190474	129	5.63%	3349620	135
XB01	1220781	1086093	281	11.03%	1060394	289	13.14%	1111985	315
XB02	8319761	5203970	618	37.45%	5144276	702	38.17%	4531979	725
XB03	4659228	3957083	319	15.07%	3490899	348	25.08%	3585616	375
XB04	33759687	31136330	845	7.77%	28714742	872	14.94%	24106526	945

TABLE 5.12: Rolling horizon heuristic solution costs

LNS 10 min		1,800 min			2,100 min		
		Cost	Seconds	Improvement wrt LNS	Cost	Seconds	Improvement wrt LNS
B01	870812	705949	338	18.93%	652546	333	25.06%
B02	1056280	849771	330	19.55%	849715	351	19.56%
B03	931289	714124	390	23.32%	714150	397	23.32%
B04	962451	734609	322	23.67%	734514	330	23.68%
B05	8377616	4490626	750	46.40%	4588308	792	45.23%
B06	2829589	1909322	388	32.52%	1909677	384	32.51%
B07	3157541	2164230	428	31.46%	2168023	432	31.34%
B08	3134454	2085396	382	33.47%	2085346	407	33.47%
B09	2676733	1827216	427	31.74%	1830889	437	31.60%
B10	32984637	23243194	1019	29.53%	25619963	919	22.33%
X01	142443	no feasible solution found			no feasible solution found		
X02	-20964	no feasible solution found			no feasible solution found		
X03	1353315	no feasible solution found			no feasible solution found		
X04	145534	no feasible solution found			no feasible solution found		
XA01	79711	72280	51	9.32%	72280	51	9.32%
XA02	1664971	1229003	123	26.18%	1123739	132	32.51%
XA03	262945	231755	73	11.86%	231755	73	11.86%
XA04	4440451	3292092	146	25.86%	3351022	146	24.53%
XB01	1220781	981685	340	19.95%	981685	635	19.59%
XB02	8319761	4466412	753	46.32%	4489754	784	46.04%
XB03	4659228	2977349	387	36.10%	2977566	433	36.09%
XB04	33759687	24793086	1052	26.56%	27245307	913	19.30%

TABLE 5.13: Improvement wrt the best known solution

	Best know solution value	Cost	% improvement wrt Best
B01	843983	710673	15.80%
B02	1220708	904519	25.90%
B03	859809	753967	12.31%
B04	968796	793647	18.08%
B05	8717434	no feasible solution found	
B06	2733781	1998689	26.89%
B07	4588857	2371871	48.31%
B08	3067543	2084658	32.04%
B09	2678741	1894053	29.29%
B10	32171455	no feasible solution found	
X01	222040	no feasible solution found	
X02	-18917	no feasible solution found	
X03	1316835	no feasible solution found	
X04	240656	no feasible solution found	
XA01	64799	72280	-11.54%
XA02	1462311	1148697	21.45%
XA03	186279	231755	-24.41%%
XA04	3972419	3560533	10.37%
XB01	1028223	981449	4.55%
XB02	9986653	no feasible solution found	
XB03	4204317	3276451	22.07%
XB04	33326556	no feasible solution found	

TABLE 5.14: Rolling horizon heuristic, column generation heuristic and LNS heuristic solution costs

LNS 10 min		Column generation			Rolling horizon			
		Cost	Seconds	Improvement	Cost	Seconds	Improvement	Computing time
				wrt LNS			wrt LNS	improvement wrt CG
B01	870812	505382	788	41.96%	652546	333	25.06%	57.74%
B02	1056280	679362	782	35.68%	849715	351	19.56%	55.12%
B03	931289	549928	824	40.95%	714124	390	23.32%	52.66%
B04	962451	604307	790	37.20%	734514	330	23.68%	58.23
B05	8377616	4357412	1351	47.99%	4490626	750	46.40%	44.49%
B06	2829589	1608877	1168	43.14%	1909322	388	32.52%	66.78%
B07	3157541	2848062	1417	9.80%	2164230	428	31.46%	69.90%
B08	3134454	1668229	1260	46.78%	2085396	382	33.47%	69.98%
B09	2676733	1512454	1202	43.50%	1827216	427	31.74%	64.48%
B10	32984637	22304388	2062	32.37%	22707143	938	31.16%	54.51%
X01	142443	125019	1456	12.23%	79753	2883	44.01%	-98.01%
X02	-20964	-36857	1179	75.81%	-45867	2834	118.79%	-140.04%
X03	1353315	1217999	1299	10.00%	1032698	3041	23.69%	-134.10%
X04	145534	140356	1288	3.56%	82853	2949	43.07%	-128.96%
XA01	79711	64799	108	18.71%	72280	51	9.32%	52.78%
XA02	1664971	951385	209	42.86%	1123739	132	32.51%	36.84%
XA03	262945	186279	91	29.16%	231755	73	11.86%	19.78%
XA04	4440451	2697195	302	39.26%	3292092	146	25.86%	51.66%
XB01	1220781	647134	855	46.99%	981685	340	19.95%	60.23%
XB02	8319761	4641007	1317	44.22%	4466412	753	46.32%	42.82%
XB03	4659228	2469046	1346	47.01%	2977349	387	36.10%	71.25%
XB04	33759687	22973144	1892	31.96%	24106526	944	28.95%	50.11%

5.6 Appendix 1

TABLE 5.15: Very large X01 instances

Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions	Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions
X01-P1C-P	100%	1	0	X0150P1C-P	50%	1	0
X01-P1C111P	100%	1	111	X0150P1C111P	50%	1	111
X01-P1C161P	100%	1	161	X0150P1C161P	50%	1	161
X01-P1C215P	100%	1	215	X0150P1C215P	50%	1	215
X01-P2C-P	100%	2	0	X0150P2C-P	50%	2	0
X01-P2C111P	100%	2	111	X0150P2C111P	50%	2	111
X01-P2C161P	100%	2	161	X0150P2C161P	50%	2	161
X01-P2C215P	100%	2	215	X0150P2C215P	50%	2	215
X01-P4C-P	100%	4	0	X0150P4C-P	50%	4	0
X01-P4C111P	100%	4	111	X0150P4C111P	50%	4	111
X01-P4C161P	100%	4	161	X0150P4C161P	50%	4	161
X01-P4C215P	100%	4	215	X0150P4C215P	50%	4	215
X0175P1C-P	75%	1	0	X0130P1C-P	30%	1	0
X0175P1C111P	75%	1	111	X0130P1C111P	30%	1	111
X0175P1C161P	75%	1	161	X0130P1C161P	30%	1	161
X0175P1C215P	75%	1	215	X0130P1C215P	30%	1	215
X0175P2C-P	75%	2	0	X0130P2C-P	30%	2	0
X0175P2C111P	75%	2	111	X0130P2C111P	30%	2	111
X0175P2C161P	75%	2	161	X0130P2C161P	30%	2	161
X0175P2C215P	75%	2	215	X0130P2C215P	30%	2	215
X0175P4C-P	75%	4	0	X0130P4C-P	30%	4	0
X0175P4C111P	75%	4	111	X0130P4C111P	30%	4	111
X0175P4C161P	75%	4	161	X0130P4C161P	30%	4	161
X0175P4C215P	75%	4	215	X0130P4C215P	30%	4	215

TABLE 5.16: Very large X03 instances

Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions	Instance	% airport capacity	Nb. aircraft disruptions	Nb. flight disruptions
X03-P1C-P	100%	1	0	X0350P1C-P	50%	1	0
X03-P1C111P	100%	1	111	X0350P1C111P	50%	1	111
X03-P1C161P	100%	1	161	X0350P1C161P	50%	1	161
X03-P1C215P	100%	1	215	X0350P1C215P	50%	1	215
X03-P2C-P	100%	2	0	X350P2C-P	50%	2	0
X03-P2C111P	100%	2	111	X0350P2C111P	50%	2	111
X03-P2C161P	100%	2	161	X0350P2C161P	50%	2	161
X03-P2C215P	100%	2	215	X0350P2C215P	50%	2	215
X03-P4C-P	100%	4	0	X0350P4C-P	50%	4	0
X03-P4C111P	100%	4	111	X0350P4C111P	50%	4	111
X03-P4C161P	100%	4	161	X0350P4C161P	50%	4	161
X03-P4C215P	100%	4	215	X0350P4C215P	50%	4	215
X0375P1C-P	75%	1	0	X0330P1C-P	30%	1	0
X0375P1C111P	75%	1	111	X0330P1C111P	30%	1	111
X0375P1C161P	75%	1	161	X0330P1C161P	30%	1	161
X0375P1C215P	75%	1	215	X0330P1C215P	30%	1	215
X0375P2C-P	75%	2	0	X0330P2C-P	30%	2	0
X0375P2C111P	75%	2	111	X0330P2C111P	30%	2	111
X0375P2C161P	75%	2	161	X0330P2C161P	30%	2	161
X0375P2C215P	75%	2	215	X0330P2C215P	30%	2	215
X0375P4C-P	75%	4	0	X0330P4C-P	30%	4	0
X0375P4C111P	75%	4	111	X0330P4C111P	30%	4	111
X0375P4C161P	75%	4	161	X0330P4C161P	30%	4	161
X0375P4C215P	75%	4	215	X0330P4C215P	30%	4	215

5.7 Appendix 2

TABLE 5.17: Rolling horizon heuristic solution costs for X01-P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X01-P1C-F	29202	29202	28877	1911	1,11%	1,13%
X01-P1C111F	563630	563630	563305	1899	0,06%	0,06%
X01-P1C161F	835229	835229	834904	1913	0.04%	0.04%
X01-P1C215F	933434	933434	933109	1913	0.03%	0.03%
X01-P2C-F	165120	165120	162645	1905	1.50%	1.50%
X01-P2C111F	699548	699548	697073	1904	0.35%	0.35%
X01-P2C161F	971147	971147	968672	1907	0.25%	0.25%
X01-P2C215F	1069353	1069353	1066878	1895	0.23%	0.23%
X01-P4C-F	912728	841019	812728	1919	10.96%	3.48%
X01-P4C111F	1449564	1378187	1449564	1923	0.00%	-4.92%
X01-P4C161F	1721160	1649784	1721160	1946	0.00%	-4.15%
X01-P4C215F	1819369	1747992	1819369	1923	0.00%	-3.92%
				1913	1.21%	-0.49%

TABLE 5.18: Rolling horizon heuristic solution costs for X0175P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0175P1C-F	3153374	3004508	2757677	2043	12.55%	8.95%
X0175P1C111F	3572524	3371889	3197068	2034	10.51%	5.47%
X0175P1C161F	3844123	3643488	3468915	2052	9.76%	5.03%
X0175P1C215F	3946908	3745664	3568266	2074	9.59%	4.97%
X0175P2C-F	3272176	3134469	2895885	2007	11.50%	8.24%
X0175P2C111F	3739109	3529792	3362360	2048	10.08%	4.98%
X0175P2C161F	4010708	3801250	3636468	2046	9.33%	4.53%
X0175P2C215F	4109453	3900437	3734436	2031	9.13%	4.45%
X0175P4C-F	4279559	4105679	4054053	2052	5.27%	1.27%
X0175P4C111F	4718534	5497474	4576832	1996	3.00%	0.45%
X0175P4C161F	5214131	5104734	5074184	2026	2.68%	0.60%
X0175P4C215F	5321330	5199999	5184522	2050	2.57%	0.30%
				2038	8.00%	4.10%

TABLE 5.19: Rolling horizon heuristic solution costs for X0150P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0150P1C-F	21948616	21363891	19406537	2503	11.58%	10.09%
X0150P1C111F	20671318	20878103	18151722	2546	12.19%	15.02%
X0150P1C161F	20564631	21414615	17513827	2525	14.84%	22.27%
X0150P1C215F	20666937	21519807	17626017	2432	14.71%	22.09%
X0150P2C-F	21992638	17078081	18842310	2391	14.32%	-9.36%
X0150P2C111F	20486160	19604211	16889389	2511	17.56%	16.07%
X0150P2C161F	21356782	21292165	18076663	2342	15.36%	17.79%
X0150P2C215F	20431422	21457855	17254349	2388	15.55%	24.36%
X0150P4C-F	21157806	18535435	18528213	2417	12.43%	0.04%
X0150P4C111F	22190477	19932935	18256991	2455	13.22%	3.51%
X0150P4C161F	22718027	20173250	20392381	2455	10.24%	-1.07%
X0150P4C215F	23171850	19736721	20459564	2475	11.71%	-3.53%
				2453	13.64%	9.77%

TABLE 5.20: Rolling horizon heuristic solution costs for X0130P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0130P1C-F	34675423	29410233	23230736	2498	15.70%	0.61%
X0130P1C111F	36074841	32182535	31414598	2517	12.92%	2.44%
X0130P1C161F	36513843	32464402	31895637	2508	12.65%	1.78%
X0130P1C215F	36298541	32825097	31719284	2507	12.62%	3.49%
X0130P2C-F	34408636	30235969	29519861	2431	14.21%	2.43%
X0130P2C111F	35976686	31986739	31019605	2591	13.78%	3.12%
X0130P2C161F	35944436	32020919	31260576	2542	13.03%	2.43%
X0130P2C215F	36680910	32435885	32295562	2531	11.96%	0.43%
X0130P4C-F	36507512	31032889	31326811	2491	14.19%	-0.94%
X0130P4C111F	36887223	32138180	31195609	2478	15.43%	3.02%
X0130P4C161F	37200872	32345717	31966348	2507	14.07%	1.19%
X0130P4C215F	37815066	32393931	32757815	2504	13.37%	-1.11%
				2508	13.66%	1.57%

TABLE 5.21: Rolling horizon heuristic solution costs for X03-P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X03-P1C-F	135214	130872	135214	1959	0.00%	-3.21%
X03-P1C111F	752125	474783	752125	1946	0.00%	-0.58%
X03-P1C161F	1073334	1068992	1073334	1941	0.00%	-0.40%
X03-P1C215F	1166308	1161966	1166308	1967	0.00%	-0.37%
X03-P2C-F	352510	340002	318013	1977	9.79%	6.91%
X03-P2C111F	974273	961909	941039	1971	3.41%	2.22%
X03-P2C161F	1295475	1283015	1261931	1966	2.59%	1.67%
X03-P2C215F	1386865	1374399	1357188	2019	2.14%	1.27%
X03-P4C-F	3516158	3516158	3516158	1937	0.00%	0.00%
X03-P4C111F	4133069	41133069	4133069	1906	0.00%	0.00%
X03-P4C161F	4454272	4454272	4454272	1909	0.00%	0.00%
X03-P4C215F	4547252	4547252	4547252	1909	0.00%	0.00%
				1951	1.49%	0.63%

TABLE 5.22: Rolling horizon heuristic solution costs for X0375P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0375P1C-F	8545854	9018845	7446975	2119	12.86%	21.11%
X0375P1C111F	9945090	8892441	7687516	2287	20.70%	15.67%
X0375P1C161F	10452293	9265976	8212186	2476	21.43%	12.83%
X0375P1C215F	10556691	9149021	8305761	2441	21.32%	10.15%
X0375P2C-F	8860841	10087113	7519444	2190	15.14%	34.15%
X0375P2C111F	9453074	10626522	7960659	2167	15.79%	33.49%
X0375P2C161F	10700057	9513342	8587105	2298	19.75%	10.79%
X0375P2C215F	10793242	9237398	8680608	2270	19.57%	6.41%
X0375P4C-F	14118152	13749637	12393596	2263	12.22%	10.94%
X0375P4C111F	12996004	14073368	11193765	2298	13.98%	25.74%
X0375P4C161F	12240808	14582552	11133030	2219	9.05%	30.98%
X0375P4C215F	14447469	14390462	13837715	2231	4.22%	3.99%
				2271	15.66%	18.02%

TABLE 5.23: Rolling horizon heuristic solution costs for X0350P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0350P1C-F	53059175	47022208	45155550	2709	14.90%	4.13%
X0350P1C111F	51079460	47594942	44666606	2790	12.55%	6.56%
X0350P1C161F	50778584	47584786	44955426	2853	11.47%	5.85%
X0350P1C215F	51650387	46312204	45294260	2828	12.11%	2.02%
X0350P2C-F	51451902	52048527	42151673	2649	18.08%	23.48%
X0350P2C111F	52597896	47124646	45416808	2712	13.65%	3.76%
X0350P2C161F	53489259	47168467	46274118	2722	13.49%	1.93%
X0350P2C215F	53779458	47541232	46359710	2657	13.80%	2.55%
X0350P4C-F	51442930	53119278	45013115	2780	12.50%	18.01%
X0350P4C111F	56579786	51511186	50461435	2719	11.34%	2.69%
X0350P4C161F	57376533	50069568	51606809	2714	10.06%	-2.89%
X0350P4C215F	58196178	52495569	50843306	2745	12.63%	3.25%
				2740	13.05%	5.94%

TABLE 5.24: Rolling horizon heuristic solution costs for X0330P instances

	Cost LNS 10 min.	Cost LNS Average time	Cost RH	Seconds	% improvement wrt LNS 10 min.	% improvement wrt LNS average time
X0330P1C-F	81339237	83705344	76190805	3074	6.33%	9.86%
X0330P1C111F	84435683	81597879	76959762	2969	8.85%	6.03%
X0330P1C161F	85665316	82611641	81329258	2987	5.06%	1.58%
X0330P1C215F	84758414	80621689	78722522	3257	7.12%	2.41%
X0330P2C-F	85079528	83260844	78743529	2965	7.45%	5.74%
X0330P2C111F	82112646	82148598	73756072	3026	10.18%	11.38%
X0330P2C161F	85409599	82935582	78539683	3081	8.04%	5.60%
X0330P2C215F	87438471	81817686	83111907	2959	4.95%	-1.56%
X0330P4C-F	89604072	85356624	82970773	3252	7.40%	2.88%
X0330P4C111F	89879808	84263755	84290957	2945	6.22%	-0.03%
X0330P4C161F	90616014	88196817	84545997	3014	6.70%	4.32%
X0330P4C215F	88925649	87678986	81632612	3060	8.20%	7.41%
				3049	7.21%	4.63%

Chapitre 6

Conclusion

Les perturbations dans le transport aérien sont fréquentes et peuvent avoir un impact significatif sur les opérations planifiées. L'utilisation des réseaux en étoile amplifie les répercussions des perturbations et ainsi, une petite perturbation, telle un vol retardé, peut causer un grand nombre de perturbations dans le reste du réseau. Les perturbations peuvent également être causées par des annulations de vol, l'indisponibilité d'un appareil ou d'un équipage ou des conditions météorologiques défavorables qui diminueront de façon significative la capacité aéroportuaire. Lorsque surviennent de tels événements, les compagnies aériennes doivent effectuer des opérations de récupération afin de rétablir les horaires planifiés.

La récupération des opérations comporte trois sous-problèmes, soit la récupération des appareils, des équipages et des passagers. Toutefois, l'approche séquentielle généralement utilisée pour résoudre ces problèmes peut entraîner une sous-optimisation du système. Il est donc important de considérer conjointement les différents sous-problèmes de récupération. Cette thèse porte sur le problème de récupération intégrée des appareils et des passagers tel que défini lors du Challenge ROADEF 2009 et présente trois méthodes de résolution.

6.1 Contributions

La première méthode proposée est une heuristique de recherche à grand voisinage utilisant, entre autres, la création de nouveaux segments de vol, la modification des paires origine-destination des passagers et une approche de destruction et de réparation des rotations d'appareils permettant une exploration approfondie de l'espace de solution. La contribution principale de ce chapitre est le développement d'une heuristique performante qui

permet d'obtenir de très bonnes solutions sur les instances du Challenge, obtenant 21 des 22 meilleures solutions moyennant un temps de calcul de 10 minutes. De plus, ce chapitre démontre qu'il peut être profitable, en terme de qualité de solution, de permettre un temps de calcul supérieur à 10 minutes, et ce, particulièrement pour les très grandes instances. Toutefois, bien qu'il soit possible de comparer les solutions obtenues avec celles obtenues par les autres méthodes proposées, il est impossible de mesurer la qualité des solutions puisqu'il n'existe jusqu'à présent aucune recherche portant sur la résolution optimale du problème proposé.

Le chapitre suivant propose un modèle de programmation mixte en nombres entiers et une heuristique de post-optimisation basée sur la génération de colonnes. L'heuristique est initialisée avec la solution obtenue en appliquant l'heuristique de recherche à grand voisinage décrite plus haut et différentes méthodes de réduction du modèle sont proposées. De plus, pour résoudre les grandes instances, il est possible de modifier le modèle afin de ne considérer que les passagers. La contribution majeure de ce chapitre est la présentation d'une heuristique très efficace qui permet d'améliorer de façon significative les solutions obtenues avec l'heuristique de recherche à grand voisinage et ainsi d'obtenir les meilleures solutions pour l'ensemble des instances du Challenge avec un temps de calcul raisonnable. La limite principale de ce chapitre est le temps de calcul nécessaire pour résoudre les instances de moyenne et de grande taille avec l'heuristique de post-optimisation intégrant les appareils et les passagers. Pour ces instances, il est nécessaire de modifier le modèle et de ne considérer que les passagers. Toutefois, ceci entraîne une sous-optimisation.

La dernière méthodologie proposée est une heuristique d'horizon roulant. Bien que toute l'information nécessaire soit disponible au début de l'horizon de planification, la complexité des problèmes de récupération et la taille des instances rendent l'utilisation d'horizon roulant appropriée. L'approche d'horizon roulant proposée intègre l'heuristique de post-optimisation présentée plus haut et permet de diminuer le temps de calcul nécessaire tout en permettant d'obtenir de bonnes solutions. La seconde contribution importante du chapitre est la création de très grandes instances comportant un nombre important de perturbations. L'heuristique est particulièrement efficace pour les instances comportant un grand nombre de perturbations d'aéroports et donc permet d'améliorer significativement les solutions lorsque les coûts de récupération sont très élevés.

6.2 Perspectives de recherche future

Afin d'évaluer la qualité des solutions obtenues avec les méthodologies présentées dans la thèse, il serait pertinent de développer des méthodes exactes permettant de calculer les bornes inférieures du problème de récupération intégrée des appareils et des passagers. Il serait également pertinent de considérer la récupération des équipages et ainsi résoudre le problème de récupération complet. De plus, l'approche d'horizon roulant pourrait être modifiée afin d'intégrer, à chaque période roulante, l'heuristique de recherche à grand voisinage et l'heuristique de post-optimisation. Les recherches futures devraient également porter sur le développement d'outils utiles pour l'industrie du transport aérien utilisant une approche d'horizon roulant qui permettrait de diviser le problème en sous-problèmes et d'appliquer les solutions des premiers sous-problèmes pendant la résolution des problèmes subséquents.

Bibliographie

- [1] databank.worldbank.org/data/.
- [2] www.iata.org.
- [3] J. Abara. Applying integer linear programming to the fleet assignment problem. *INTERFACES*, 19 :20–28, 1989.
- [4] A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, 127 :309–331, 2004.
- [5] K. F. Abdelghany, A. F. Abdelghany, and G. Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185 :825–848, 2008.
- [6] R. Acuna-Agost. Passenger improver (PI). A new approach for the passenger recovery problem. POSTDOC Report, AMADEUS and École nationale de l’aviation civile, Toulouse, 2010.
- [7] S. AhmadBeygi, A. Cohn, and M. Weir. An integer programming approach to generating airline crew pairings. *Computers & Operations Research*, 36 :1284–1298, 2009.
- [8] A. H. Ahmed and C. A. Poojari. An overview of the issues in the airline industry and the role of optimization models and algorithms. *Journal of the Operational Research Society*, 59 :267–277, 2008.
- [9] R. Anbil, R. Tanga, and E. L. Johnson. A global approach to crew-pairing optimization. *IBM Systems Journal*, 31 :71–78, 1992.
- [10] M.F. Arguello, J.F. Bard, and G. Yu. A grasp for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 5 :211–228, 1997.
- [11] M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, and A. Odoni. Total delay impact study : a comprehensive assessment of the costs and impacts of flight delay in the united states. *Washington, D.C. : FAA*, 2010.

- [12] M. Ball, C. Barnhart, M. Dresner, M. Hansen, K. Neels, A. Odoni, E. Peterson, L. Sherry, A. Trani, and B. Zou. Total delay impact study. Technical report, NEXTOR, 2010.
- [13] M. Ball, C. Barnhart, G. L. Nemhauser, and A. Odoni. Air transportation : Irregular operations and control. In *Handbooks in Operations Research and Management Science, Vol. 14*, pages 1–61. North-Holland, Amsterdam, 2007.
- [14] C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37 :368–391, 2003.
- [15] C. Barnhart, L. Hatay, and E. L. Johnson. Deadhead selection for the long-haul crew pairing problem. *Operations Research*, 43 :491–499, 1995.
- [16] C. Barnhart, T. S. Kniker, and M. Lohatepanont. Itinerary-based airline fleet assignment. *Transportation Science*, 36 :199–217, 2002.
- [17] C. Barnhart and R. G. Shenoi. An approximate model and solution approach for the long-haul crew pairing problem. *Transportation Science*, 32, 1998.
- [18] M. E. Berge and G. A. Hopperstad. Demand driven dispatch : A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research*, 41 :153–168, 1993.
- [19] S. Bisailon, J. F. Cordeau, G. Laporte, and F. Pasin. A large neighbourhood heuristic for the aircraft and passenger recovery problem. *4OR : A Quarterly Journal of Operations Research*, 9 :139–157, 2010.
- [20] N. Bélanger, G. Desaulniers, F. Soumis, and J. Desrosiers. Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues. *European Journal of Operational Research*, 175 :1754–1766, 2006.
- [21] S. Bratu and C. Barnhart. Flight operations recovery : New approaches considering passenger recovery. *Journal of Scheduling*, 9 :279–298, 2006.
- [22] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part i : Mathematical formulation. *Transportation Planning and Technology*, 20 :183–199, 1997.
- [23] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays part ii : Algorithm and computational experiments. *Transportation Planning and Technology*, 20 :201–217, 1997.
- [24] P. Cappanera and G. Gallo. A multicommodity flow approach to the crew rostering problem. *Operations Research*, 52 :583–596, 2004.

- [25] S. Chand, V. Hsu, and S. Sethi. Forecast, solution and rolling-horizons in operations management problems : a classified bibliography. *Manufacturing and Service Operations Management*, 4 :25–43, 2002.
- [26] J. Clausen, A. Larsen, J. Larsen, and N. Rezanova. Disruption management in the airline industry - concepts, models and methods. *Computers & Operations Research*, 37 :809–821, 2010.
- [27] T.F. Coleman and L.A. Hulburt. A direct activeset algorithm for large sparse quadratic programs with simple bounds. *Mathematical Programming*, 45 :373–406, 1989.
- [28] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. M. Solomon, and F. Soumis. Crew pairing at air france. *European Journal of Operational Research*, 97 :245–259, 1997.
- [29] S. Dozic, M. Kalic, and O. Babic. Heuristic approach to the airline schedule disturbances problem : single fleet case. *Procedia-Social and Behavioral Sciences*, 54 :1232–1241, 2012.
- [30] N. Eggenberg, M. Salani, and M. Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Computers & Operations Research*, 37 :1014–1026, 2010.
- [31] T. A. Feo and J. F. Bard. Flight scheduling and maintenance base planning. *Management Science*, 35 :1415–1432, 1989.
- [32] J.A. Filar, P. Manyem, and K. White. How airlines and airports recover from schedule perturbations : A survey. *Annals of Operations Research*, 108 :315–333, 2001.
- [33] F. Furini, M. P. Kidd, C. A. Persiani, and P. Toth. Improved rolling horizon approaches to the aircraft sequencing problem. *Journal of Scheduling*, pages 1–13, 2015.
- [34] R. Gopalan and K. T. Talluri. The aircraft maintenance routing problem. *Operations Research*, 46 :260–271, 1998.
- [35] C. Hane, C. Barnhart, E. L. Johnson, R. E. Marsten, G. L. Nemhauser, and G. Sigismondi. The fleet assignment problem : Solving a large-scale integer program. *Mathematical Programming*, 70 :211–232, 1995.
- [36] N. Jafari and S. H. Zegordi. The airline perturbation problem : considering disrupted passengers. *Transportation Planning and Technology*, 33 :203–220, 2010.
- [37] N. Jafari and S. H. Zegordi. Simultaneous recovery model for aircraft and passengers. *Journal of the Franklin Institute*, 348 :1638–1655, 2011.
- [38] A. I. Z. Jarrah and J. T. Diamond. The problem of generating crew bidlines. *Interfaces*, 27 :49–64, 1997.

- [39] A.I.Z. Jarrah and G. Yu. A decision support framework for airline cancellations and delays. *Transportation Science*, 27 :266–280, 1993.
- [40] N. Jozefowiez, C. Mancel, and F. Mora-Camino. A heuristic approach based on shortest path problems for integrated flight, aircraft and passenger rescheduling under disruptions. *LAAS technical report, Université de Toulouse*, 2010.
- [41] L. Lettovsky, E. L. Johnson, and G. L. Nemhauser. Airline crew recovery. *Transportation Science*, 34 :337–348, 2000.
- [42] M. Lohatepanont and C. Barnhart. Airline schedule planning : Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38 :19–32, 2004.
- [43] S. Luo and G. Yu. On the airline schedule perturbation problem caused by the ground delay program. *Transportation Science*, 31 :298–311, 1997.
- [44] R. Mansi, S. Hanafi, C. Wilbault, and F. Clautiaux. Disruptions in the airline industry : math-heuristic for re-assigning aircraft and passengers simultaneously. *European Journal of Industrial Engineering*, 6 :690–712, 2012.
- [45] C. P. Medard and N. Sawhney. Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183 :1013–1027, 2007.
- [46] L.K. Nielsen, L. Kroon, and G. Maroti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220 :465–509, 2012.
- [47] R. Nissen and K. Haase. Duty-period-based network model for crew rescheduling in european airlines. *Journal of Scheduling*, 9 :255–278, 2006.
- [48] M. Palpant, M. Boudia, C-A. Robelin, S. Gabteni, and F. Laburthe. Roadev 2009 challenge : Disruption management for commercial aviation. *Amadeus S.A.S, Operations Research Division, Sophia Antipolis, France, www.roadev.org*, 2009.
- [49] J. Peekstok and E. Kuipers. Roadev 2009 Challenge : Use of a simulated annealing-based algorithm in disruption management for commercial aviation. [http ://challenge.roadev.org/2009/files/PeekstokKuipers.pdf](http://challenge.roadev.org/2009/files/PeekstokKuipers.pdf).
- [50] J. D. Petersen, G. Solveling, E. L. Johnson, J.-P. Clarke, and S. Shebalov. An optimization approach to airline integrated recovery. *Transportation Science*, 46 :482–500, 2012.
- [51] B. Rexing, C. Barnhart, T. Kniker, A. Jarrah, and N. Krishnamurthy. Airline fleet assignment with time windows. *Transportation Science*, 34 :1–20, 2000.

- [52] J. M. Rosenberger, E. L. Johnson, and G. L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37 :408–421, 2003.
- [53] R. A. Rushmeier and S. A. Kontogiorgis. Advances in the optimization of airline fleet assignment. *Transportation Science*, 31 :159–169, 1997.
- [54] M. Saddoune, G. Desaulniers, and F. Soumis. Aircrew pairings with possible repetitions of the same flight number. *Computers & Operations Research*, 40 :805–814, 2013.
- [55] K. Sinclair, J.-F. Cordeau, and G. Laporte. A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. Technical report, CIRRELT, Montreal, 2014.
- [56] K. Sinclair, J.-F. Cordeau, and G. Laporte. Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem. *European Journal of Operational Research*, 233 :234–245, 2014.
- [57] K. Sinclair, J.-F. Cordeau, and G. Laporte. A column generation post-optimization heuristic for the integrated aircraft and passenger recovery problem. *Computers & Operations Research*, 65 :42–52, 2015.
- [58] PassMark software. <http://www.cpubenchmark.net/singleThread.html>.
- [59] C. Sriram and A. Haghani. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A*, 37 :29–48, 2003.
- [60] M. Stojkovic, F. Soumis, and J. Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32 :232–245, 1998.
- [61] M. Stojkovic, F. Soumis, J. Desrosiers, and M. M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research Part A*, 36 :779–788, 2002.
- [62] D. Teodorović and S. Guberinić. Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15 :178–182, 1984.
- [63] J. Törnquist. Railway traffic disturbance management - an experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research Part A*, 41 :249–266, 2007.
- [64] Z. Xiuli and G. Yanchi. An improved grasp for irregular flight recovery. *IEEE International Conference on System Science and Engineering (ICSSE)*, pages 465–469, 2012.

- [65] S. Yan, C. H. Tang, and T. C. Fu. An airline scheduling model and solution algorithms under stochastic demands. *European Journal of Operational Research*, 190 :22–39, 2008.
- [66] S. Yan and H. F. Yang. A decision support framework for multi-fleet routing and multi-stop flight scheduling. *Transportation Research Part A*, 30 :379–398, 1996.
- [67] G. Yu, M. Arguello, G. Song, S.M. McCowan, and A. White. A new era for crew recovery at continental airlines. *Interfaces*, 33 :5–22, 2003.
- [68] S. H. Zegordi and N. Jafari. Solving the airline recovery problem by using ant colony optimization. *Internation Journal of Industrial Engineering & Production Research*, 23 :121–128, 2010.
- [69] D. Zhang and Y.K. Lau. A rolling horizon based algorithm for solving the integrated airline schedule recovery problem. *Journal of Automation and Control Engineering*, 2 :232–237, 2014.
- [70] Y. Zhang and M. Hansen. Real-time inter-modal substitution (rtims) as an airport congestion management strategy. *Transportation Research Records*, 2052 :90–99, 2008.