THÈSE DE DOCTORAT
de l'École des Ponts ParisTech

Demand based optimization for airline industry and load balancing

École doctorale N° 532, Mathématiques et Sciences et Technologies de l'Information et de la Communication

Spécialité : Mathématiques

Thèse préparée au sein du CENTRE D'ENSEIGNEMENT ET DE RECHERCHE EN MATHEMATIQUES ET CALCUL SCIENTIFIQUE

Thèse soutenue le 7 décembre 2021, par
Sébastien DESCHAMPS

Composition du jury :

François CLAUTIAUX Professeur, Institut de Mathématiques de Bordeaux	*Rapporteur*
Dominique FEILLET Professeur, Mines de Saint-Étienne	*Rapporteur*
Antoine DÉSIR Maître de conférence, INSEAD	*Examinateur*
Ivana LJUBIC Professeure, ESSEC Business School	*Examinatrice*
Frédéric MEUNIER Professeur, École des Ponts ParisTech	*Directeur de thèse*
Axel PARMENTIER Maître de conférence, École des Ponts ParisTech	*Encadrant de thèse*

Abstract

This thesis develops optimization methods applied in the airline industry, and has been conducted in collaboration with Air France. This partnership falls within a scientific chair between Air France and École des Ponts ParisTech. In this thesis we explore several research topics, namely aircraft routing, revenue estimation and optimization, and load balancing problems (which are not related to a problem linked with the airline industry).

We introduce an aircraft routing problem which aims at building routes for the planes of the company. Those routes need to verify two main types of operational constraints: frequency constraints and gate constraints. The frequency constraints ensure that the different origin-destinations the company wants to serve are effectively served. The gate constraints form a new kind of constraints that arise because of the increase in plane traffic at the hub of the company. This increase has made the number of gate slots available at the hub a limiting constraint, which we take into account in our model. We show that the problem considered is NP-hard, and propose a mixed integer linear program to generate solutions efficiently.

The revenue evaluation method used in such scheduling problems is usually leg-based, in the sense that an expected revenue by flight leg is defined. The revenue estimation is the sum of theses expected revenues on the flights operated. This method is an approximation since the revenues of a company are actually generated by the sales of tickets for itineraries, which are formed by successions of legs. We study a stochastic itinerary-based revenue model that aims at evaluating quickly the revenue generated by a schedule. We show that this model can be approximated as a linear program already known in the literature as the Sales Based Linear Program (SBLP). We propose a column generation based on the Dantzig–Wolfe reformulation of the SBLP that performs very well in practice. We also study the convergence of the stochastic model and show that it happens at an exponential rate.

We also present a model that solves the aircraft routing with an itinerary-based revenue. The considered problem can be seen as joining the aircraft routing problem with frequency and gate constraints and the SBLP. Given the size and the structure of the problem, we propose a Benders decomposition method to solve it. We develop an advanced method that takes advantage of the column generation method used to solve the SBLP, to generate cuts for the Benders decomposition.

Finally, this thesis has also been the opportunity to study some problems, independently of the research in the airline industry. We develop some theoretical results on load balancing problems. Those problems aim at sharing “fairly” tasks among workers. We show that several “fair” objective functions are closely related. More particularly, we show that solving the problem with one of the objectives considered provides a solution for the other problems. We also generalize and extend these results to similar problems.

Keywords: operations research, aircraft routing, revenue management, Dantzig–Wolfe decomposition, Benders decomposition, mixed-integer linear programming, load balancing.

Résumé

Cette thèse développe des méthodes d'optimisation appliquées à l'industrie aérienne. Elle a été menée en collaboration avec Air France dans le cadre d'une chaire scientifique avec l'École des Ponts ParisTech. Cette thèse explore plusieurs sujets de recherches : la planification des programmes de vols, l'estimation et l'optimisation des revenus générés pas ces plannings et l'équilibrage de charges de travail.

Nous introduisons un problème de planification de programme de vols qui cherche à établir les routes prises par les différents avions de la compagnie. Ces routes doivent principalement respecter deux contraintes opérationnelles : une contrainte de fréquence et une contrainte de portes. La contrainte de fréquence s'assure que les différentes origines-destinations proposées par la compagnie sont effectivement desservies. La contrainte de portes s'inscrit dans une nouvelle classe de contraintes introduite en raison de l'augmentation du trafic aérien dans le hub de la compagnie. Cette augmentation a rendu essentielle la prise en compte du nombre limité de portes (pour l'embarquement et le débarquement des passagers) disponibles au hub, ce que l'on veut donc inclure dans le modèle. Nous prouvons que le problème ainsi considéré est NP-complet et nous proposons un programme linéaire en nombres entiers capable de trouver une solution efficacement.

Les méthodes d'évaluation du revenu d'un planning de vols sont souvent approchées par une combinaison linéaire des revenus estimés pour chaque vol. Cette façon de faire est approximative puisque le revenu d'une compagnie est en réalité généré par la vente de billets associés à des itinéraires (formés d'une succession de vols). Nous étudions donc un modèle stochastique de revenu basé sur la vente d'itinéraires, qui vise à estimer rapidement le revenu généré par un planning de vols. Nous prouvons que ce modèle peut être approché par un autre problème connu dans la littérature sous le nom de "Sales Based Linear Program" (SBLP). Pour trouver une solution à ce problème, nous proposons d'utiliser une génération de colonnes basée sur sa reformulation de Dantzig-Wolfe. Cette méthode se révèle être très efficace en pratique. Nous étudions aussi l'arrivée dynamique des passagers, qui est utilisée pour l'estimation de revenu, et sa convergence vers une dynamique fluide approchée. Nous montrons qu'il existe une convergence exponentielle de l'espérance du nombre d'achats de billet vers la valeur donnée par la dynamique fluide quand la taille du problème augmente.

Nous présentons également un modèle qui optimise le planning des vols avec une estimation des revenus basée sur la vente d'itinéraires. Au vu de la taille et de la structure du problème, nous utilisons une décomposition de Benders pour résoudre celui-ci. Afin de tirer parti de la génération de colonnes utilisée pour résoudre le SBLP, nous développons une méthode avancée pour générer des coupes de Benders.

Pour finir, cette thèse a aussi été l'occasion d'étudier d'autres problèmes, indépendamment de ceux traités dans le monde de l'aérien. Ainsi, nous avons obtenu des résultats théoriques concernant des problèmes d'équilibrage de charges de travail. Ces problèmes cherchent à partager de façon "équitable" le temps passé sur différentes tâches entre des travailleurs. Nous prouvons que différents critères équitables sont étroitement liés. Plus précisément, nous montrons que résoudre le problème posé avec une fonction objectif particulière donne une solution pour d'autres fonctions objectif. Nous généralisons et étendons ces résultats à des problèmes similaires.

Mots-clés : Recherche opérationnelle, gestion du revenu, décomposition de Dantzig–Wolfe, décomposition de Benders, programme linéaire en nombres entiers, équilibrage de charge.

Contents

List of Figures

List of Tables

1 Introduction

This thesis in operations research has been conducted through an industrial collaboration with Air France and develops several research topics motivated by the company's operation processes.

The use of operations research in the airline industry has been an active topic since the second half of the 20^{th} century. With the increase of air traffic in the last decades, the use of advanced optimization tools has become even more crucial. Nowadays, airlines use advanced optimization technologies to improve their process. Usually, the design of a company's flight schedule is at the center of this process. For an airline, establishing a schedule means that it has to determine the precise departure an arrival times of all the planes of the company, with many different constraints, but also the planning of the crews and the different tools used for managing the planes. Figure 1.1 summarizes the main subjects airlines use operations research for.

Designing the schedule for a company with dozens of planes, hundreds of crews and thousands of legs is a central yet complicated problem. Therefore, this design is usually split into several simpler steps as presented in Figure 1.1. Schedule planning aims at choosing which origin-destinations are going to be served by the company at a very high level. Those decisions are usually taken manually by experts who fix the general guidelines that the schedule should follow. For instance, the company decides at this step to operate one trip between Paris and New York every morning of the week.

When those decisions have been made, the precise schedule still needs to be determined. Optimization problems are leveraged to take good (optimal) decisions for these processes. Those optimization problems precisely establish the schedule of all planes of the company. This means departure and arrival times of the planes, but also the schedule followed by the crews, the ground staff and equipment. According to Barnhart and Cohn [9], the successive steps are usually the following: Fleet assignment, aircraft routing and crew pairing. Fleet assignment takes the trips that have been selected by schedule planning and decides which fleet type should operate these trips. Then the aircraft routing establishes the precise sequence of flights of the different planes of the company, including different kinds of constraints, as maintenance constraints or other practical constraints. The crew pairing then aims at finding which crew should be assigned to the flight legs selected at the previous steps.

In this thesis, we start by introducing a new version of the aircraft routing problem in Chapter 2. The problem we model includes frequency constraints and gate constraints, but no maintenance

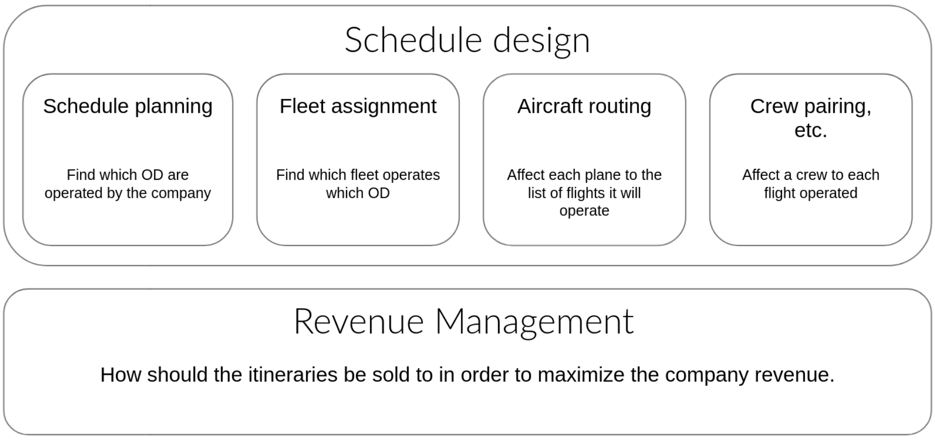

Figure 1.1 – Main optimization problems in the airline industry

constraints. The frequency constraints are an expression of the requirements imposed by the schedule planning and fleet assignment steps. The gate constraints are a new kind of constraint that limits the number of available gate slots for planes of the company in the different airport served by the airline. It is motivated by the increase of traffic during the last decade, which has led to slot shortages.

Another important use of operations research by airlines concerns the evaluation of the revenue generated by a schedule. Precisely estimating the revenue is difficult since the selling strategy highly impacts the revenue. Finding how to sell the different tickets falls into the scope of revenue management which is a very active and complex optimization field.

In Chapter 3, we study a simple revenue model with the objective of being able to quickly estimate the revenue generated by a schedule, but still with high accuracy. In order to be precise, the stochastic aspect of revenue management needs to be included in the model. We therefore include a discrete choice model that simulates the sales of itineraries to customers. In that sense it is "itinerary-based." We show that it can be approximated by a problem known in the literature as the Sales-Based Linear Program (SBLP). We propose a column generation approach that solves efficiently the SBLP.

Revenue management and schedule design are usually handled separately for complexity reasons. The revenue management strategy of the company is updated and solved on a daily basis with a precision that asks for (at least) several hours of daily computations. Combining revenue considerations with scheduling tasks leads therefore to complicated problems, usually intractable.

Even though addressing simultaneously those two optimizations seems intractable with present day technologies, revenue estimation and schedule design have a mutual influence on each other. Indeed, having a precise estimation of the revenue is crucial in order to generate a good schedule. Conversely, even the best revenue management algorithms will perform poorly on a bad schedule.

Chapter 4 introduces an optimization problem that models the two aspects: It seeks a schedule that generates a maximum revenue according to the model of Chapter 3, while including the aircraft routing constraints of Chapter 2.

The last chapter of this thesis presents an independent problem, which is not related to airline operations. It focuses on the "fair" allocation of tasks to workers with different qualifications. We prove that different fair objective functions are in fact closely related.

We give now further details on the content of the different chapters in this thesis.

Aircraft routing with gate and frequency constraints

The aircraft routing problem aims at finding the routes that will be followed by the different planes of the company. In Chapter 2, we introduce an aircraft routing problem with two unusual add-ons. In the standard aircraft routing problem, the precise departure and arrival times of a flight are known. Here we let some additional flexibility: these times must be chosen in a predefined set of possible times. We must be able to operate those times with the planes of the airline. This is the most important aspect of the problem from a combinatorial point of view and the reason why we call it an aircraft routing problem. The selected times must also respect several constraints. First they must respect some frequency constraints: those constraints ensure that the decision made by the schedule planning and the fleet assignment are respected. Second, the aircraft routing solution must respect a gate constraint. That is, it must not require more slots that those available for the airline. Notice that we do not model the maintenance constraints in this model. We show that the resulting problem is NP-hard (Theorem 2 in Chapter 2):

Theorem. *The feasibility version of the aircraft routing problem with frequency and gate constraints is NP-complete.*

We also propose a linear program that models the problem. This problem is formulated as a circulation over a "flight graph" $D = (V, A)$ of the company. The definitions of this graph as well as other quantities used in the program are omitted for simplicity's sake; see Chapter 2 for the exact definitions. However, we introduce the program here to emphasize that it has a small size in the input of the problem. Indeed, the problem can be seen as a circulation problem with a few additional constraints. Namely, the gate constraints (one fore each time step and airport), some constraints on the number of planes available (one for each fleet of the company) and the frequency constraints (one by frequency requirements).

$$\min \sum_{e\in E} c_e x_e \tag{1.1a}$$

$$\text{st: } \sum_{e\in\delta^+(v)} x_e = \sum_{i\in\delta^-(v)} x_e \qquad \forall v\in V \tag{1.1b}$$

$$\sum_{e\in E} o_t^{e,a} x_e \leqslant \gamma_a \qquad \forall t\in[T], \forall a\in\mathcal{A} \tag{1.1c}$$

$$\sum_{e\in T_1^k} x_e = n_k \qquad \forall k\in\mathcal{K} \tag{1.1d}$$

$$\sum_{e\in L_f} x_e = 1 \qquad \forall f\in\mathcal{F} \tag{1.1e}$$

$$x_e\in\{0,1\} \qquad \forall e\in E. \tag{1.1f}$$

In this problem the objective (1.1a) is to maximize the sum of the revenue generated by the used legs. Constraints (1.1b) ensure that the solution found is a circulation, Constraints (1.1c) that the gate constraints are respected, and Constraints (1.1d) that the number of available planes does not exceed the number of available planes. The last Constraints (1.1e) ensures the solution respects the frequency constraint. Experiments confirm that it is tractable: it runs quickly on real size instances.

Itinerary-based revenue estimation

Chapter 3 deals with the problem of estimating the revenue generated by a schedule. The research field of revenue management provides a rich literature on the elaboration of selling strategies in order to optimize the revenue of a company. However, these methods are computationally expensive, and in this chapter, we are interested in the quick computation of the revenues generated by a schedule.

In order to evaluate the revenue, we suppose that a customer is interested in a single market: the set of all itineraries with the same origin destination. We suppose that customer choices can be captured with the "basic attraction model." In other words, for a given itinerary $i \in m$, we suppose that a customer buys this itinerary with probability $g_i = \frac{\gamma_i}{\gamma_m^0 + \sum_{j \in m} \gamma_j}$, where γ_j and γ_m^0 are positive values representing the utility of itinerary j and of the option "not buying" respectively. We denote by D_m the number of customers interested in a market m.

Every time an itinerary is sold, the corresponding seats in the legs $\ell \in L$ are no longer available. Since the number of seats in each leg ℓ is limited to a quantity s_ℓ, the number of available itineraries decreases along the process. Without any control, this could lead to situations in which the company can no more sell some itineraries while it would have been more lucrative to keep them available. Therefore, the company needs to prioritize the sales of more lucrative itineraries taking into account the number of remaining seats. In order to control the sales of the itineraries, we consider that the company can set a maximal number y_i of tickets of itinerary i that can be sold. This is interpreted as a capacity y_i for an itinerary i.

We call "atomic" the arrival dynamic in which customers arrive following the basic attraction model with respect to the capacities set by the company and denote by $Q_{i,T,y}$ the number of customers that bought itinerary i after T arrivals, and with capacity y. Even for fixed capacities, it is hard to estimate the number of customers that have bought a given itinerary at the end of this atomic arrival dynamic. Therefore, we introduce a fluid approximation of this dynamic, and prove that when the atomic dynamic is scaled by a factor θ, it converges at exponential speed toward the fluid approximation when $\theta \to \infty$.

Numerical experiments suggest the existence of a simple rule for discriminating between itineraries whose sales are overestimated by the fluid approximation and those that are underestimated. More precisely, we propose the following conjecture (it is Conjecture 20 presented in Chapter 3).

Conjecture. *When the itineraries are sorted following the quantities $\frac{y_i}{g_i}$, there exists an index k such that the fluid approximation underestimates $\mathbb{E}[Q_{i,T,y}]$ for $i > k$, and overestimates the quantity for $i \leqslant k$.*

Each itinerary i is sold at a price c_i. For a capacity vector y, the total revenue is $\sum_i c_i \mathbb{E}[Q_{i,T,y}]$. As it is hard to estimate the quantities $\mathbb{E}[Q_{i,T,y}]$, even for a given y, the optimization problem is difficult to solve. We show that it can be approximated by solving the SBLP, which takes the following form:

$$\begin{aligned}
V^{SBLP} = \max \; & \sum_{m \in M} \sum_{i \in m} c_i q_i \\
\text{st:} \; & \sum_{i | l \in i} q_i \leqslant s_l && \forall l \in L \\
& \frac{q_i}{\gamma_i} \leqslant \frac{q_m^0}{\gamma_m^0} && \forall i \in m, \forall m \in M \\
& q_m^0 + \sum_{i \in m} q_i = D_m && \forall m \in M \\
& q_i \geqslant 0 && \forall i \in m, \forall m \in M \\
& q_m^0 \geqslant 0 && \forall m \in M,
\end{aligned}$$

The objective of this problem is to maximize the revenue of a given schedule. The first constraint limits the number of tickets sold according to the number of seats available in the planes of the company. The second one ensures that the balance of demand for the itineraries follows the customer choice model used. The third one captures the number of customers that are interested in every market.

We prove that approximating the revenue generated by the atomic model with the SBLP is relevant in the following sense, where we denote by V_θ^{atom} the expected revenue of the atomic model scaled by a factor θ:

Theorem. *We have* $\liminf_{\theta \to \infty} \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant V^{SBLP}$.

We also propose a column generation method that is able to solve quickly the SBLP. Indeed, the problem has a particular structure that can be exploited: The constraints on the seat number link several independent sub-problems (one for each market). This kind of structure can be exploited with a classical method in operation research called the Dantzig–Wolfe decomposition. This decomposition consists in reformulating the problem as an optimization problem over the set of extreme points of the constraint polyhedron. In that case, the number of variables of the problem becomes huge, and thus it is solved using a column generation. With the particular structure described, the pricing problems of the column generation can then be split in independent sub-problems which makes the decomposition efficient.

The Dantzig–Wolfe reformulation of the SBLP can therefore be solved through column generation in order to quicken the resolution. This approach reveals to be particularly efficient because the sub-problems can be solved very quickly: we do not need to use a linear programming solver to generate an optimal solution. We even prove that those sub-problems can be solved in linear time, using the linear time median search in a list.

Aircraft routing with itinerary-based revenue

When needed for schedule optimization, the estimation of the revenue generated by a schedule is built as follows. It relies on an estimation of the revenue generated by each flight potentially operated by the company (independently of the schedule used). The estimation of the revenue generated is then simply the sum of the revenues generated by the flight included in that schedule. In that sense, the estimated revenue is leg-based. It is the estimation that has been used in the aircraft routing problem of Chapter 2. This is a rough approximation since in practice the revenue is generated by the sales of itineraries (and not the sales of seats) to customers as discussed in Chapter 3.

Leg-based revenue models are blind to some customer behaviors that influence the revenue. In particular phenomena like “cannibalization” and “spill and recover” cannot be integrated in a leg-base revenue model. These phenomena occur when several itineraries are sold in the same markets. Cannibalization happens when one itinerary is much more attractive than the other: It can then lead to the more attractive one being full at the expense of the others that might end up empty. Spill and recover designates the report of demand of an itinerary that is full and no longer available to other itineraries that are similar. Taking into account such phenomena in a leg-based revenue model is difficult. Therefore, in order to precisely estimate the revenue of a company, it is natural to rely on itinerary-based revenue models.

Combining the model of Chapter 2 with the SBLP of Chapter 3 leads to a MILP formulation modeling the aircraft routing problem with an itinerary-based objective function. Thanks to the relative simplicity of the SBLP, this program is tractable for small instances. Since the number of variables in the problem grows with the number of itineraries in the network, MILP solvers do not scale well on large instances of that problem.

Since the size of this problem makes it intractable on real size instance, we try to exploit the structure of the problem in order to solve it. There are few integer variables, all located in the aircraft routing part of the problem, and many continuous variables associated with the revenue estimation part of the problem. This structure suggests using a Benders decomposition method in order to isolate the integer variables (which are the “complicating” variables in the Benders terminology) in the master problem and the continuous variables in the slave problem. The master problem has a structure similar to the aircraft routing problem of Chapter 2, and the slave problem has a structure similar to the SBLP of Chapter 3.

Since the slave problem of the Benders decomposition has a structure similar to the SBLP, we naturally try to quicken the resolution of the Benders decomposition by using the column generation method introduced in Chapter 3. Practical experiments reveal that a vanilla use of this column generation does not perform well because of the degenerate nature of the slave problem. The precise study of this method shows however that it is very efficient to generate a primal solution of the slave problem. Yet, the Benders decompositon method needs the dual solutions of the slave problem in order to generate a cut. Those are not easy to deduce from the column generation method.

In order to circumvent this difficulty, we propose an alternative algorithm able to take advantage of the column generation and to find the dual solutions needed to generate a cut. We prove that this method is able to generate good quality Benders cuts when solved to optimality. However, the method does not outperform the direct resolution of the problem with a solver, leaving open the challenge of generating quickly a good solution.

Load balancing problem

This thesis has also been the opportunity to investigate other operations research problems, independently of the studied airline schedule problems. We study in Chapter 5 an assignment problem. The aim of this problem is to assign fairly some tasks among workers while respecting skill constraints. Fairness will be captured by various criteria on the respective workloads of the workers. We show that, surprisingly, minimizing the difference between the worker with maximal load and the worker with minimum load provides a solution that is optimal for other fair criteria. In particular, it is optimal for the objective function that minimizes the maximal load among all workers. Formally, we look at the following problems:

$$\begin{array}{lll} \text{Minimize} & \max_{w\in W} \sum_{e\in\delta(w)} x_e & \\ \text{subject to} & \sum_{e\in\delta(u)} x_e = d(u) & \forall u \in U \\ & x_e \in X & \forall e \in E, \end{array} \tag{Min-max^X}$$

$$\begin{array}{lll} \text{Minimize} & \max_{w,w'\in W} \left(\sum_{e\in\delta(w)} x_e - \sum_{e\in\delta(w')} x_e \right) & \\ \text{subject to} & \sum_{e\in\delta(u)} x_e = d(u) & \forall u \in U \\ & x_e \in X & \forall e \in E, \end{array} \tag{Min-diff^X}$$

where $G = (W \cup U, E)$ is a bipartite graph. The vertices of the graph represent a set of workers W and a set of tasks U. There exists an edge between $w \in W$ and $u \in U$ if the worker w is skilled to perform the task u. The two problems aim at finding an assignment of the tasks to the workers represented as the variable x. The constraints ensure that each task $u \in U$ takes a total of $d(u)$ amount of time to be performed. We consider that the set X can either be equal to $\mathbb{R}_{\geqslant 0}$ or $X = \mathbb{Z}_{\geqslant 0}$. The only difference between Problems (Min-max^X) and (Min-diff^X) lies in the objective function which evaluates the load of work for the workers differently.

Theorem. *Every optimal solution of (Min-diff^X) is simultaneously optimal for (Min-max^X), whether $X = \mathbb{R}_{\geqslant 0}$ or $X = \mathbb{Z}_{\geqslant 0}$.*

Actually, every optimal solution of (Min-diff^X) also gives an optimal solution of the problem that maximizes the minimal load among workers. This theorem is related with a problem known in the literature as the semi-matching problem, which focuses on constraints of the form $d(u) = 1$, and objective functions of the form $\sum_{w\in W} f\left(\sum_{e\in\delta(w)} x_e\right)$ where f is a convex function.

We also show that there exists a polynomial algorithm that finds optimal solutions of the Problem (Min-diff^X) (and therefore also for Problem (Min-max^X)) in the two cases $X = \mathbb{R}_{\geqslant 0}$ and $X = \mathbb{Z}_{\geqslant 0}$.

We extend these results to close problems. Surprisingly, one of them requires a non-trivial topological argument dealing with the connectivity of the feasible set.

1 Introduction (version française)

Cette thèse en recherche opérationnelle a été menée dans le cadre d'une collaboration industrielle avec Air France et développe plusieurs sujets de recherche motivés par les besoins opérationnels de la compagnie.

L'utilisation de la recherche opérationnelle dans l'industrie aérienne est devenue centrale depuis la deuxième moité du XX$^{\text{ème}}$ siècle. L'augmentation du trafic aérien à rendu cruciale l'utilisation de méthodes d'optimisation avancées ces les dernières décennies. Aujourd'hui encore, ces techniques d'optimisation sont un enjeu majeur dans le développement des compagnies aériennes. Elles sont en particulier utilisées pour créer les emplois du temps des avions et du personnel. Les emplois du temps des avions, du personnel naviguant et du personnel au sol doivent tous être précisément définis, coordonnés tout en respectant chacun différentes contraintes. Le Schéma 1.1 résume les principaux sujets traités par la recherche opérationnelle dans les compagnies aériennes.

Pour une compagnie aérienne, mettre au point un emploi du temps avec des centaines d'avions, des milliers d'employés et des milliers de vols est un enjeu majeur. Pour s'attaquer à cette création d'emploi du temps, sa conception est généralement séparée en plusieurs étapes, présentées dans le Schéma 1.1. La planification de l'emploi du temps a pour but de choisir les origines-destinations qui seront desservies par la compagnie. Ces décisions sont généralement prises à échelle humaine par des experts qui fixent les objectifs globaux que les emplois du temps devront suivre. Par exemple, c'est à ce niveau que la compagnie décide qu'elle veut garantir un vol Paris-New York chaque matin de la semaine.

Une fois ces décisions stratégiques prises, l'emploi du temps précis doit encore être déterminé. Différents problèmes d'optimisation sont utilisés afin de faire les meilleurs choix possibles (voire si possible des choix optimaux). Le planning précis de tous les avions est alors fixé. Cela signifie que les horaires de départ et d'arrivée précis sont déterminées, mais cela implique aussi les choix de planning pour le personnel naviguant, le personnel au sol ainsi que celui de touts les équipements utilisés pour le fonctionnement des avions. Selon Barnhart and Cohn [9], les étapes d'optimisation suivies successivement par les compagnies sont généralement les suivantes : l'affectation des flottes, la création des routes pour les avions et le couplage avec les équipages. L'affectation des flottes consiste à déterminer quel type d'avion devrait desservir les différentes destinations de la compagnie. La création des routes vise à établir précisément les séquences de vols qui doivent être effectuées par les différents avions. Le couplage des équipages cherche

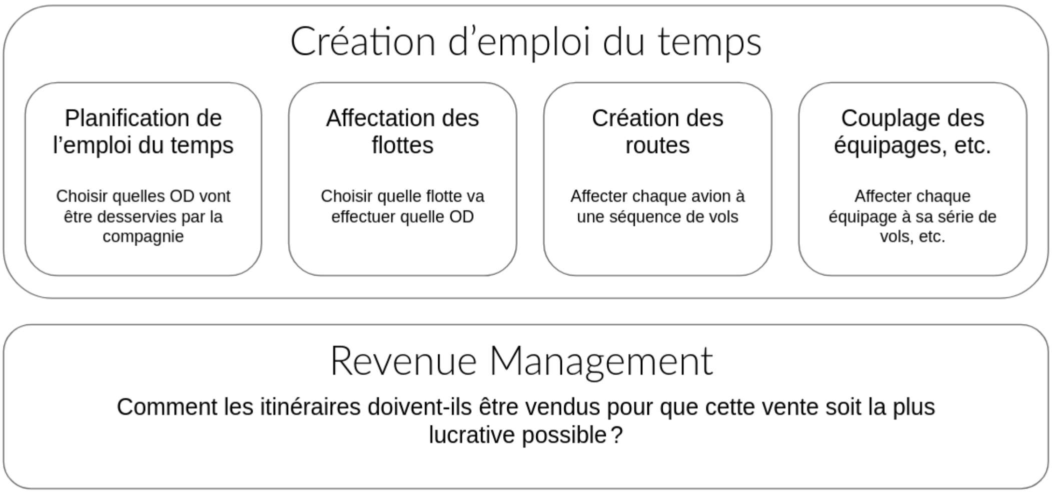

Figure 1.1 – Principaux problèmes d'optimisation dans une compagnie aérienne

ensuite à affecter un équipage à chacun des vols choisis aux étapes précédentes.

Dans cette thèse, nous commençons, dans le chapitre 2, par introduire une nouvelle version du problème de création de routes. Le problème auquel on s'intéresse inclut des contraintes de fréquence et des contraintes de portes, mais pas de contrainte de maintenance. Les contraintes de fréquence reflètent les exigences imposées par les étapes précédentes de planification et d'affectation des flottes sur l'emploi du temps. Les contraintes de portes constituent une nouvelle classe de contraintes. Celles-ci visent à limiter le nombre d'emplacements de portes disponibles pour la compagnie dans les différents aéroports desservis. L'introduction de cette contrainte est motivée par l'augmentation du trafic aérien des dernières années qui a rendu cruciale la prise en compte de la limite de tels emplacements.

Une autre utilisation importante de la recherche opérationnelle pour les compagnies aériennes se concentre sur l'évaluation du revenu généré par un emploi du temps. Estimer précisément ce revenu est une tâche difficile puisque la stratégie de vente impacte grandement les revenus. Trouver comment vendre au mieux les billets est une problématique traitée par le revenue management, qui est un domaine d'optimisation complexe et très actif.

Dans le chapitre 3, nous étudions un modèle de revenu simplifié, ayant pour objectif d'estimer rapidement le revenu généré par un planning de vols, avec une précision élevée. Plus précisément, la dimension stochastique du revenue management doit être incluse dans le modèle. Pour cela, nous utilisons donc un modèle de choix discret qui simule la vente d'itinéraire aux consommateurs. En ce sens, le modèle utilisé a une précision au grain de l'itinéraire. On montre que ce modèle peut être approximé par un modèle connu dans la littérature comme le Sales-Based Linear Program (SBLP). Nous proposons l'utilisation d'une méthode de génération de colonne pour résoudre ce problème, méthode qui se révèle être efficace pour résoudre le problème.

Le revenue management et la création du planning sont généralement menés séparément pour des raisons de complexité. La stratégie de revenue management de la compagnie est mise à jour quotidiennement. Cette mise à jour demande la résolution d'un problème avec une précision qui entraine plusieurs heures de calculs chaque jour. Allier ces optimisations sur le revenu à la création d'un emploi du temps crée donc un problème très difficile, qui ne passe généralement pas à l'échelle.

Même si la résolution simultanée de ces deux problèmes n'est pas envisageable aujourd'hui, le revenue management et la création d'emploi du temps sont des problèmes qui s'influencent mutuellement. En effet, une estimation précise du revenu est très importante pour créer un bon emploi du temps. À l'inverse, le meilleur algorithme de revenue management aura des résultats médiocres si il est utilisé avec un mauvais emploi du temps.

Le chapitre 4 introduit un problème d'optimisation qui modélise ces deux aspects : il vise à créer un emploi du temps qui maximise le revenu estimé à partir du modèle du chapitre 3, tout en prenant en compte les contraintes du chapitre 2.

Le dernier chapitre de cette thèse présente un problème indépendant du monde aérien. Il se concentre sur l'affectation "équitable" de tâches entre des travailleurs avec des compétences différentes. Nous prouvons que différentes fonctions objectifs équitables sont étroitement liées.

Nous allons maintenant présenter plus en détail le contenu des différents chapitres de cette thèse.

Création de routes avec contraintes de fréquence et de portes

Le problème de création de routes (*aircraft routing* en anglais) à pour but de déterminer les séquences de vols (routes) qui devront être suivies par les avions de la compagnie. Dans le chapitre 2, on introduit un problème de création de routes avec deux ajouts inhabituels. Dans le problème standard, l'heure précise de départ et d'arrivée des vols est déjà connue. Ici, on s'autorise plus de flexibilité : ces horaires doivent être choisis parmi un ensemble de possibilités prédéfinies. On cherche donc à choisir des horaires de façon à ce que ceux-ci puissent être utilisés par les avions de la compagnie. D'un point de vue combinatoire, cet aspect du problème est central et justifie la qualification de ce problème comme un problème de création de routes. Les horaires sélectionnés doivent aussi respecter plusieurs contraintes. Premièrement, elles doivent respecter des contraintes de fréquence : celles-ci doivent assurer que les décisions précédentes faites lors de la planification d'emploi du temps et de l'affectation des flottes sont vérifiées. Ensuite, la solution du problème doit aussi respecter des contraintes de portes. Plus précisément le nombre d'emplacements de portes requis ne doit pas excéder la limite fixée pour la compagnie dans les différents aéroports. Il est important de noter que le modèle considéré ne prend pas en compte de contraintes de maintenance. Nous prouvons que le problème issu du modèle considéré est NP-difficile (Théorème 2 du chapitre 2) :

Théorème. *Le problème de décision associé au problème de création de routes avec des contraintes de fréquence et de portes est NP-complet.*

Nous introduisons également un programme linéaire qui modélise le problème. Celui-ci est formulé comme un problème de circulation sur un "graphe des vols" $D = (V, A)$ de la compagnie. Les définitions précises de ce graphe et des autres quantités utilisées pour définir le programme sont omises pour des raisons de simplicité ; voir le chapitre 2 pour les définitions exactes. Nous présentons tout de même le programme afin de mettre en avant sa faible taille relativement aux paramètres d'entrée du problème. En effet, le problème peut être vu comme un problème de circulation auquel s'ajoute d'autres contraintess. Plus précisément, on ajoute les contraintes de portes (une par pas de temps et par aéroport), quelques contraintes qui fixent le nombre de portes (une par flotte de la compagnie) et les contraintes de fréquence (une par fréquence demandée dans le modèle).

$$
\begin{aligned}
\min \sum_{e\in E} c_e x_e & & & \quad (1.1a)\\
\text{st: } \sum_{e\in\delta^+(v)} x_e = \sum_{i\in\delta^-(v)} x_e & & \forall v \in V & \quad (1.1b)\\
\sum_{e\in E} o_t^{e,a} x_e \leqslant \gamma_a & & \forall t\in[T], \forall a \in \mathcal{A} & \quad (1.1c)\\
\sum_{e\in T_1^k} x_e = n_k & & \forall k \in \mathcal{K} & \quad (1.1d)\\
\sum_{e\in L_f} x_e = 1 & & \forall f \in \mathcal{F} & \quad (1.1e)\\
x_e \in \{0,1\} & & \forall e \in E. & \quad (1.1f)
\end{aligned}
$$

Dans ce problème, l'objectif (1.1a) a pour but de maximiser la somme des revenus générés par les vols sélectionnés. Les Contraintes (1.1b) s'assurent que la solution générée est bien une circulation, les Contraintes (1.1c) que les limitations sur les emplacements de portes sont respectées, et les Contraintes (1.1d) que le nombre d'avions utilisés n'excède pas le nombre disponible pour la compagnie. Les dernières Contraintes (1.1e) imposent à la solution de vérifier les contraintes de fréquence. Les résultats numériques menés confirment que le modèle est efficace : il s'exécute rapidement sur des instances de taille réelle.

Estimation de revenue au grain itinéraire

Le chapitre 3 se concentre sur l'estimation du revenu généré par un emploi du temps. Le champ de recherche du revenue management est un champ riche qui développe différentes stratégies de ventes pour optimiser les revenus d'une compagnie. Les méthodes développées demandent cependant une grosse puissance de calcul et un temps non négligeable. Dans ce chapitre, nous sommes intéressées par l'évaluation rapide des revenus générés par un emploi du temps.

Pour évaluer le revenu, on suppose que le consommateur est intéressé par un unique marché : l'ensemble de tous les itinéraires qui ont la même origine destination. On suppose que les choix des consommateurs sont reflétés par un "modèle d'attraction basique". En d'autres termes, pour un itinéraire donné $i \in m$, on suppose qu'un consommateur achète cet itinéraire avec la probabilité $g_i = \frac{\gamma_i}{\gamma_m^0 + \sum_{j\in m}\gamma_j}$, où γ_j et γ_m^0 sont des réels positifs représentant l'utilité de l'itinéraire j et de l'option de non-achat respectivement. On note D_m le nombre de consommateurs intéressés par le marché m.

À chaque fois qu'un itinéraire est vendu, un siège de moins dans les vols de cet itinéraire est disponible. Comme le nombre de places dans chaque vol ℓ est limité par un nombre limite s_ℓ, le nombre d'itinéraires disponibles diminue au cours du processus de vente. Sans contrôle sur les ventes, cela mène à des situations où la compagnie ne peut plus vendre certains itinéraires alors qu'il aurait été plus lucratif de les garder à disposition. Ainsi, la compagnie doit prioriser les ventes des itinéraires lucratifs en prenant en compte le nombre de places disponibles dans les différents vols programmés. Pour contrôler ces ventes d'itinéraires, on considère donc que la compagnie fixe un nombre maximal de billets y_i qui peuvent être vendus pour l'itinéraire i. Cette quantité est interprétée comme la capacité y_i de l'itinéraire i.

On appelle "atomique" le modèle d'arrivée dynamique dans lequel les consommateurs arrivent suivant le modèle d'attraction basique et respectent les capacités. On note $Q_{i,T,y}$ le nombre de consommateurs qui ont acheté l'itinéraire i après T arrivées, avec des capacités y. Même à capacités fixées, estimer le nombre de consommateurs qui ont acheté un itinéraire donné est une tâche difficile. Nous introduisons donc une approximation fluide de cette dynamique, et prouvons que dans la taille de la dynamique atomique est augmentée par un facteur θ, le modèle atomique converge à une vitesse exponentielle (pour $\theta \to \infty$) vers le modèle fluide.

Les expérimentations numériques menées suggèrent l'existence d'une règle simple pour déterminer quels itinéraires ont un remplissage final surestimé et sous-estimé par le modèle fluide. Plus précisément, nous proposons la conjecture suivante (correspondant à la Conjecture 20 du chapitre 3).

Conjecture. *Si les itinéraires sont triés suivant les quantités $\frac{y_i}{g_i}$, il existe un indice k tel que le modèle fluide sous-estime $\mathbb{E}[Q_{i,T,y}]$ pour $i > k$ et surestime cette quantité pour $i \leqslant k$.*

Chaque itinéraire i est vendu à un prix c_i. Pour un vecteur de capacité y, le revenu total vaut $\sum_i c_i \mathbb{E}\left[Q_{i,T,y}\right]$. Comme la quantité $\mathbb{E}\left[Q_{i,T,y}\right]$ est difficile à estimer, même à y donné, le problème d'optimisation est un problème difficile à résoudre. Nous montrons qu'il peut être approché par la résolution du SBLP, qui s'écrit comme suit :

$$
\begin{aligned}
V^{SBLP} = \max \ & \sum_{m \in M} \sum_{i \in m} c_i q_i & \\
\text{st:} \ & \sum_{i | l \in i} q_i \leqslant s_l & \forall l \in L \\
& \frac{q_i}{\gamma_i} \leqslant \frac{q_m^0}{\gamma_m^0} & \forall i \in m, \forall m \in M \\
& q_m^0 + \sum_{i \in m} q_i = D_m & \forall m \in M \\
& q_i \geqslant 0 & \forall i \in m, \forall m \in M \\
& q_m^0 \geqslant 0 & \forall m \in M,
\end{aligned}
$$

L'objectif de ce problème est de maximiser le revenu d'un planning de vols donné. La première contrainte limite le nombre de billets vendus en fonction du nombre de places libre dans les avions de la compagnie. La deuxième s'assure que la demande sur les différents itinéraires suit bien le modèle de choix discret sélectionné. La dernière impose le nombre de consommateurs sur chaque marché.

On prouve que l'approximation par le SBLP du revenu généré par le modèle atomique est pertinente, dans le sens suivant, où V_θ^{atom} désigne le revenu espéré du modèle atomique dont la taille est augmentée par un facteur θ :

Théorème. *On a* $\liminf_{\theta \to \infty} \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant V^{SBLP}$.

Nous proposons également une génération de colonnes qui est capable de résoudre le SBLP rapidement. En effet, le problème a une structure particulière qu'il est possible d'exploiter : Les contraintes qui portent sur le nombre limité de places lie plusieurs sous-problèmes indépendants (un pour chaque marché). Cette structure peut être exploitée avec une méthode classique en recherche opérationnelle appelée décomposition de Dantzig–Wolfe. Cette décomposition consiste à reformuler le problème comme un problème d'optimisation sur l'ensemble des points extrêmes du polyèdre des contraintes. Dans ce cas, le nombre de variables du problème devient immense, et il est naturel d'utiliser une génération de colonnes pour le résoudre. Avec la structure particulière du problème, les sous-problèmes de la génération de colonne sont séparables, ce qui rend cette décomposition efficace.

Une génération de colonnes peut donc être utilisée pour trouver rapidement une solution de la reformulation de Dantzig–Wolfe du SBLP. Cette méthode est très efficace en pratique, car les sous-problèmes peuvent être résolus très rapidement : il n'y a pas besoin d'utiliser un solveur linéaire pour générer une solution optimale. Nous prouvons même que ces sous-problèmes peuvent être résolus en temps linéaire grâce à un algorithme linéaire de recherche de médiane.

Problème de création de routes avec un revenu généré par itinéraires

Quand elle est utilisée pour l'optimisation d'emploi du temps, l'estimation du revenu généré par celui-ci est généralement menée d'une manière particulière. Elle est basée sur une estimation de revenu associée à chaque vol potentiel effectué (indépendamment de l'emploi du temps utilisé). L'estimation de revenu généré par l'emploi du temps est alors calculée comme la somme des estimations pour chaque vol utilisé. En ce sens, le revenu est estimé au grain des vols. C'est ce modèle de revenu qui est utilisé dans le chapitre 2. Cette approximation est grossière puisque le revenu est généré en pratique par la vente d'itinéraires comme présenté dans le chapitre 3.

Les estimations au grain des vols ne prennent pas en compte les comportements des utilisateurs qui influent sur le revenu. En particulier, des phénomènes comme la "cannibalisation" ou le "spill and recover" ne sont pas pris en compte. Ces phénomènes se manifestent quand des itinéraires sont vendus sur un même marché. La cannibalisation se produit quand un itinéraire est beaucoup plus attractif qu'un autre : cela amène l'itinéraire le plus attractif à être plein aux dépens des autres qui peuvent se retrouver vides. Le "spill and recover" désigne le report de la demande qui s'effectue depuis un itinéraire qui n'est plus offert à la vente (car plein) vers d'autres itinéraires similaires. Il est difficile de prendre en compte de tels phénomènes avec un revenu calculé au grain des vols. Pour estimer plus précisément les revenus d'une compagnie, il est donc naturel de se tourner vers un modèle de revenu au grain itinéraire.

En combinant le modèle du chapitre 2 avec le SBLP du chapitre 3, on forme un PLNE qui modélise le problème de création de routes avec un revenu généré par itinéraires. Grâce à la relative simplicité du SBLP, le programme ainsi formé est adapté pour résoudre des instances de taille modeste. Mais comme le nombre de variables grandit avec le nombre d'itinéraires sur le réseau, les solveurs pour PLNE ne sont pas capables de passer à l'échelle pour des instances de grande taille.

Puisque la taille du problème rend la résolution de celui-ci impraticable pour des instances de taille réelle, on essaye d'exploiter la structure du problème afin de le résoudre plus efficacement.

La formulation du problème utilise peu de variables entières, toutes utilisées pour le problème de création de routes, et un grand nombre de variables continues, toutes utilisées pour l'estimation de revenu. Cette structure est adaptée à l'utilisation d'une décomposition de Benders qui sert à isoler les variables entières dans un problème maitre et les variables continues dans un problème esclave. Le problème maitre a alors une structure similaire au problème de création de route du chapitre 2, et le problème esclave a une structure similaire au SBLP du chapitre 3.

Comme le problème esclave à une structure proche du SBLP, nous essayions d'accélérer la résolution de la décomposition de Benders en utilisant la génération de colonne introduite au chapitre 3. Les expériences numériques montrent que l'utilisation directe de cette génération de colonne n'a pas de bons résultats, en raison de la nature dégénérée du problème esclave. Une étude plus poussée révèle toutefois qu'elle est très efficace pour générer une solution primale du problème. Pour générer une coupe de Benders, les variables duales sont cependant nécessaires. Celles-ci ne sont pas facilement trouvables à partir de la génération de colonnes.

Afin de contourner cette difficulté, nous proposons un algorithme alternatif qui profite à la fois de la génération de colonnes et trouve les valeurs duales nécessaires à la génération des coupes de Benders. Nous prouvons que cette méthode génère des coupes de Benders d'une bonne qualité. Cependant, cette méthode n'est pas plus efficace que la résolution directe du problème avec un solveur, ce qui laisse ouverte la question de trouver rapidement une bonne solution au problème.

Problème d'équilibrage de charges de travail

Cette thèse a aussi été l'occasion d'explorer d'autres problèmes de recherche opérationnelle, exterieurs aux problèmes du monde aérien. Dans le chapitre 5, nous étudions un problème d'affectation. Le but de ce problème est d'affecter équitablement différentes tâches à des travailleurs tout en respectant des contraintes sur les qualifications de ceux-ci. L'équité est mesurée par différents critères portant sur les charges de travail des travailleurs. Nous montrons qu'étonnement, minimiser la différence entre le travailleur qui travaille le plus et celui qui travaille le moins génère une solution optimale pour d'autres critères d'optimisation. En particulier, il est optimal pour la fonction objectif qui vise à minimiser la charge maximale de travail entre les travailleurs. Formellement, nous nous intéressons aux problèmes suivants:

$$\begin{aligned}
\text{Minimize} \quad & \max_{w\in W} \sum_{e\in\delta(w)} x_e \\
\text{subject to} \quad & \sum_{e\in\delta(u)} x_e = d(u) \quad && \forall u \in U \\
& x_e \in X && \forall e \in E,
\end{aligned} \tag{Min-max^X}$$

$$\begin{aligned}
\text{Minimize} \quad & \max_{w,w'\in W} \Big(\sum_{e\in\delta(w)} x_e - \sum_{e\in\delta(w')} x_e \Big) \\
\text{subject to} \quad & \sum_{e\in\delta(u)} x_e = d(u) \quad && \forall u \in U \\
& x_e \in X && \forall e \in E,
\end{aligned} \tag{Min-diff^X}$$

où $G = (W \cup U, E)$ est un graphe biparti. Les sommets de ce graphe représentent un ensemble de travailleurs W et un ensemble de tâches U. Il existe une arête entre $w \in W$ et $u \in U$ si le travailleur w est qualifié pour effectuer la tâche u. Les deux problèmes ont pour but de trouver une affectation des tâches aux travailleurs, qui est représentée par la variable x. Les contraintes s'assurent que chaque tâche $u \in U$ prenne un temps total $d(u)$ pour être achevée. On considère que l'ensemble X peut être égal à $\mathbb{R}_{\geqslant 0}$ ou $X = \mathbb{Z}_{\geqslant 0}$. La seule différence entre les problèmes (Min-max^X) et (Min-diff^X) concerne les fonctions objectifs qui évaluent les charges de travail différemment.

Théorème. *Toute solution optimale de (Min-diff^X) est simultanément optimale pour (Min-max^X), pour $X = \mathbb{R}_{\geqslant 0}$ ou $X = \mathbb{Z}_{\geqslant 0}$.*

On a même que toute solution optimale de (Min-diff^X) donne aussi une solution optimale du problème qui vise à maximiser la charge de travail minimale parmi des travailleurs. Ce théorème est lié à un problème connu dans la littérature sous le nom de semi-couplage, qui se concentre sur des contraintes de la forme $d(u) = 1$, et des fonctions objectif de la forme $\sum_{w \in W} f\left(\sum_{e \in \delta(w)} x_e\right)$ où f est une fonction convexe.

Nous montrons également qu'il existe un algorithme polynomial pour trouver une solution optimale du Problème (Min-diff^X) (et donc du problème (Min-max^X)), dans les deux cas $X = \mathbb{R}_{\geqslant 0}$ et $X = \mathbb{Z}_{\geqslant 0}$.

Nous étendons ces résultats à des problèmes proches. De manière surprenante, une de ces généralisations fait appel à un argument non trivial de topologie en rapport avec la structure connexe de l'ensemble des solutions admissibles.

2 Aircraft routing and gate management

2.1 Introduction

In this thesis we are interested in introducing models for the schedule design problem. The schedule design aims at choosing the general schedule operated by the airline. The objective is to establish the destinations and the frequency with which those destinations must be served in order to maximize profit. Of course, the airline needs to be able to operate the flights of the schedule with the available resources: planes, crew members, slots in airports, etc. We cannot model the full complexity of airline operations in a single schedule design problem as it would lead to an intractable optimization problem. We therefore focus on the most critical resources. At Air France, the most critical resources happen to be the number of planes and the gate slots available at the airport.

Due to the increase in demand and the competition between the different companies, the slots available in the hubs have become a limiting resource in terms of schedule design. Before take off or after landing, planes have to park at a *gate* in order to let passengers get in or out. The number of gate slots available for Air France at the hub Charles de Gaulle is limited, which has become a limiting constraint in recent years. When too many planes are parked at the gates dedicated to the company at the same time, any new arriving plane needs to park in a remote location. This generates an important cost for the company since the passengers and the mechanical support for the plane need to be moved to this remote location.

The problem on which we focus deals with the design of the routes of the airplanes and has to consider quite precisely the arrival and departure times because of the gate constraints. The problem shares therefore simultaneously characteristics with classical problems of the literature, namely the aircraft routing, the tail assignment and the flight retiming.

The aircraft routing problem aims at building the sequence of flight legs or *routes* operated by the planes of an airline, in such a way that each flight is covered by exactly one plane. In this problem, the planes are not considered individually but as a fleet of identical planes. The tail assignment aims at precisely determining the route followed by each individual plane of the fleet (identified with the number on its tail).

The flight retiming problem has a particular status in the classical operations research problems for airlines. It does not aim precisely at building a schedule, but rather takes an existing one and seeks changes in the flight departure times in order to improve a given objective. The objective can vary a lot from one version of the problem to another. For instance the aim can be to reduce the delay propagation, to increase the resilience of the planning or to maximize the generated profit.

In this chapter, we propose to enrich the aircraft routing problem with gate constraints. The limiting number of gate slots available at the hub of the company makes the gate slots a limiting resource. The airline therefore needs to adapt the time schedule and park planes away (which is called distant parking) during some periods of time in order to enforce this constraint. We do not however model maintenance constraints. As far as we know, this forms a problem that has not been addressed yet in the literature. Our contribution is twofold. First, we prove that several special cases of the problem with the gate constraints are NP-hard. Second, we introduce a tractable MILP model based on a flow formulation.

From an operational point of view, the aircraft routing problem usually follows the schedule design and fleet assignment that have already been solved by the company. In the context of this chapter, it means that we should respect the decision of those preceding problems. In particular, at Air France, this constraint takes the form of frequency constraints: each origin-destination and date determined by the schedule design should be operated by a flight with a given fleet type.

Finally, we also generated several instances of the problem with different size and structure in order to benchmark the problem. The generated data is of two types: a long-haul flight type database and a mid-haul flight type database. The long-haul flight type database is composed of flights that are either from the hub to a distant airport or from a distant airport to the hub. The mid-haul flight type database is more general in the sense that there can be flights between any pair of airports in the database. We explore the performances of the model with those two database types and with a varying number of airports, flights, and number of legs.

The chapter is organized as follows:

- Section 2.2 gives a general view on the aircraft routing and tail assignment recent literature.
- Section 2.3 provides a mathematical formulation of the problem and discusses its relevance.
- Section 2.4 provides the complexity results.
- Section 2.5 introduces the mixed integer linear program modeling the problem.
- Section 2.6 presents the instances that have been generated in order to benchmark the model and gathers the results of the numerical experiments that have been carried out.

2.2 Literature review

In this section, we give a general overview of the literature that relates to our problem. We begin by looking at the aircraft routing and tail assignment problems because the structure of our problem is very similar to those. After that, we present the flight retiming literature since the model developed can be used as a large scale retiming tool.

2.2.1 Aircraft routing and tail assignment

The aircraft routing and the tail assignment problems aim at building the sequence of flight legs operated by the company, with the inclusion of different operational constraints. The difference between the two problems lies in whether each physical plane of the airline is considered separately or not. The aircraft routing focuses on a given fleet while the tail assignment focuses on each plane separately (identified by the number on their tail). Despite this difference, models and algorithms for the two problems are very similar and therefore, we will present them jointly.

There exists an important variety of models used in order to solve aircraft routing or tail assignment. The different models that can be found usually differ in operational constraints they consider, or in the optimized objective. A large part of the literature focuses on the integration of maintenances in the schedule. Different kinds of maintenance constraints have been studied in the literature and in the industry since the early 60s. Etschmaier and Mathaisel [47] review gathers the early work on the subject.

The first models proposed generally aimed at including maintenance constraints in the creation of a schedule for the different planes of the company. Feo and Bard [50] study the problem of finding the minimum number of maintenance station to meet a 4-day maintenance requirement and propose heuristics to solve it. Gopalan and Talluri [61] show that the maintenance constraints (in their case) can be included in a graph and solved the problem as an Eulerian tour problem. Clarke et al. [32] solve the problem as a Hamiltonian cycle in the line graph associated to the graph of flights. They solve the problem with a Lagrangian relaxation and also propose an alternative polynomial algorithm. Talluri [130] also formalizes a version of the aircraft routing problem with four-day maintenance as an Eulerian tour problem. Levin [84] presents a compact version of the tail assignment problem where they find the minimum chain decomposition of the acyclic graph representing the flight schedule. To solve this problem, they reformulate it as an equivalent max flow problem in a bipartite graph.

Many models in aircraft routing have been developed with the inclusion of maintenance constraints, and in the context of tactical planning (the schedule is designed around two months before the practical operations). More recently, the scope of problems solved has widened. Part of the literature has developed short-term models able to include operational constraints and to provide solutions that will be applied only a few days from the resolution. For instance, Sarac et al. [116] solve the tail assignment problem with a one-day horizon considering some short term operational maintenance constraints. They linearize it and add valid cuts in order to propose en efficient resolution method. Some works also include resilience in the models such as Lan et al. [82], Weide et al. [150] or Jamili [74]. Khaled et al. [77] propose a model based on the formulation of Khaled et al. [76] that optimize the problem with a multi-criteria objective. The compactness of the model developed has also been a concern in the recent years. Haouari et al. [66] present a compact problem for the daily aircraft routing and develop a non-linear program to generate a feasible solution. Khaled et al. [76] formulation also present a compact version of the problem, but their formation takes the form of a linear program.

From an algorithmic perspective, two main model types have been developed during the last decades. Those two types are associated with two kinds of optimization algorithm that are the state-of-the-art solution approaches. A first class of models is based on a set partitioning formulation of the problem. In those models, one variable represents the full sequence of flights, maintenance and movements of a plane during the time horizon considered. In this case, state-of-the-art resolution method use column generation algorithms to seek efficiently solutions. Barnhart et al. [8] develop such a column generation method to solve the problem. Lan et al. [82] further extend the work of [8] using a column generation in a resilient framework. It has the particularity to minimize the propagated delay, while the most common objective is maximizing the revenue. Froyland et al. [53] work on the tail assignment problem, using the model introduced in [8]. They add resilience to the model through a two stage stochastic program. They enhance resilience by allowing cancellation and aircraft re-routing. Furthermore, they solve at the same time the tail assignment problem (first stage problem) and the recovery tail assignment model (second stage problem). They use a Benders decomposition which generates sub-problems for each scenario and strengthen the generated cuts with a Magnanti–Wong method. Moreover, they use column generation to solve the master and the slave problems. Gabteni and Grönkvist [54] solve the tail assignment with a set partitioning formulation. They use a column generation with a branch and bound where the branching is driven by constraint programming to solve this problem.

The second type of models used are compact mathematical programming models. Those models have fewer variables than the number of possible sequences of legs in a set partitioning formulation. Typically, they have a number of variables that is not exponential but linear in the size of the graph. They have the advantage of generating bounds quickly, and can often be solved with commercial solvers directly through a mixed integer program formulation. We can point out that the set partitioning formulation of an aircraft routing is generally a Dantzig–Wolfe reformulation of the corresponding compact model. For instance, Desaulniers et al. [41] work on the daily aircraft routing and scheduling problem, and propose two formulations of the problem. One of those can be formulated as a mixed integer program based on a time-space graph, with decision variables associated with the possible connexion between two flights. The other is a set partitioning type problem, solved with column generation. Sriram and Haghani [125] also present a mixed integer program, for the aircraft routing problem over a week horizon. Their problem has a multi-criteria objective and their heuristic strategy to solve it is to mix branch and bound and random search.

In general, both the compact models and the (pricing subproblems of) set partitioning formulations rely on the concept of state-time graphs where the graphs represent the flights operated by the company and include the links between the different airports at different times. Hane et al. [65] were among the first to introduce this concept of state-time graph and formulated the aircraft routing as a flow problem on this graph. Different variations of the state-time graph have been developed over time. Two main types of graph gather most of the used models. A first kind of state-time graph associates one node of the graph to each airport at each time step considered. The arcs between those vertices represent the different flights of the company. For instance, Liang et al. [87] and Liang and Chaovalitwongse [86] develop this kind of models respectively for a two-days aircraft routing and a weekly aircraft routing problems. Safaei and Jardine [113] work on a model that includes generalized maintenance motivated by the fact that mathematical models in the literature generally only take into account one type of constraint. Their model is a multi-flow mixed integer linear program on a state time graph, with different variables for each plane.

In the second type of state-time graph, each flight leg operated by the company is associated with a vertex. Two vertices are connected if the connection between the two corresponding flights is possible. Salazar-González [114] develops a method integrating aircraft routing and crew pairing with this kind of graph. Recently, Parmentier and Meunier [104] propose a compact formulation of the aircraft routing, including the maintenance constraint for a fixed number of days between maintenance.

2.2.2 Flight retiming

The model we develop in this chapter aims at choosing the time of every flight of the schedule. In that sense, its aim is close to the objective of the flight retiming problem. The general idea behind flight retiming is to consider many possible flight departure times close from an existing schedule, and choose better timings for the aircraft. The aim is usually to increase the general revenue generated, the resilience of the planning or to include last minute operational constraints. Many resolution techniques for this problem rely on iterative approaches or stochastic programming, and most of the models used are formulated as mathematical programs.

A lot of work has been focused on the improvement of the resilience of the aircraft routing problem. Jiang [75] and Warburg et al. [148] were among the first to consider and study the idea of retiming in order to improve the resilience and the revenue of an airline. Burke et al. [25] develop a model to improve the resilience of a schedule with a multi-objective approach. They start from a fixed fleet assignment and perform at the same time flight retiming and aircraft re-routing. The graph structure we use in this chapter is a state-time type of graph that is similar to the one used for their problem.

Aloulou et al. [4] work on the stability of the aircraft schedule and proposed a model that performs flight retiming on an existing schedule. The mixed integer linear program they propose integrate slacks across the connecting flights to prevent possible delays. Dunbar et al. [43] develop a retiming heuristic to further develop the integrated routing and crewing model of Dunbar et al. [42]. They use heuristics that increase the resilience of the solution, through the use of scenarios. That way, they improve their solution of integrated crew paring and aircraft routing, without destroying the aircraft of the crew routes (they treat the two routes simultaneously), and they limit the delay propagation in the solution. Cacchiani and Salazar-Gonzalez [27] have recently incorporated a flight retiming framework to an integrated approach. They further develop the model presented in Cacchiani and Salazar-Gonzalez [26]. Including flight retiming, they add resilience and increase the effective profit of their previous model. They propose four heuristics to solve it, and they compare those heuristics to one another. Their heuristics have two phases, first they find a lower bound of the problem and then generate a feasible solution. Pita et al. [105] propose a mixed-integer linear model that integrates flight scheduling and fleet assignment in order to take into account the congestion in airports. Their demand model is itinerary based, and takes spill and recapture into account.

In the case of our problem, the range of possible times to choose from can be large, and does not necessarily take an existing schedule but only needs the frequency with which each flight should be operated. Therefore, our model can be used as a comprehensive timetable tool. This kind of non-incremental approach has not been very common in the literature, but Wei et al. [149] have recently introduced such an idea.

2.2.3 Slot management

To the best of our knowledge, the management of gate slots has never been investigated from the airline perspective in an aircraft routing problem. It had never been investigated by Air France, and we did not find any previous work in the literature on that topic either. However, the slots management has been considered for airports, since they need to coordinate many arrivals and departure from planes of different companies, at the same time. Two main lines of work exist about this problem. A first one focuses on the auction problem that leads to the assignment of gate slots to the different companies. Another one presents the reorganization of slots assignment in order to enforce some new constraint or to fix changes from an existing schedule. Since those problems are quite different from the flight retiming or the aircraft routing problems, we do not develop those in this literature review, but dedicate a complementary review in Section 2.7.

2.3 Problem statement

Let $\mathcal{A}$ be the set of airports. The airline operates planes split into fleets (e.g., Airbus 330, Boeing 777). We denote by $\mathcal{K}$ the set of fleets, and for each fleet $k \in \mathcal{K}$, we denote by n_k the number of planes available within this fleet. The problem takes place over a time interval that repeats itself, whose time steps are the elements in $[T]$. (Typically, it is a discretization of a week.) For instance, if a leg is supposed to leave New York on Sunday at 11 p.m. and fly during 8 hours, we consider that it will arrive on Monday at 7 a.m. (We suppose here that all times are given with the same reference time zone.) Consequently, the legs $\ell \in L$ that have an arrival time after the horizon T arrive at the beginning of $[T]$. We will denote by $(t_1, t_2)_T$ the cyclic time interval between t_1 and t_2 (meaning that if $t_2 < t_1$, the interval is not empty but it is equal to $[T] \setminus [t_2, t_1]$).

In the problem, a *leg* ℓ is an eight-tuple with a departure airport $a^{\mathrm{d}}_\ell \in \mathcal{A}$, a departure time $t^{\mathrm{d}}_\ell \in [T]$, a gate departure time $t^{\mathrm{d,gate}}_\ell \in [T]$, an arrival airport $a^{\mathrm{a}}_\ell \in \mathcal{A}$, an arrival time $t^{\mathrm{a}}_\ell \in [T]$, a gate arrival time $t^{\mathrm{a,gate}}_\ell \in [T]$, a fleet $k_\ell \in \mathcal{K}$, and a revenue $c_\ell \in \mathbb{R}_+$. We denote by L the set of all legs.

There is a *plane connection* from a leg ℓ to a leg ℓ' if $a^{\mathrm{a}}_\ell = a^{\mathrm{d}}_{\ell'}$ and $k_\ell = k_{\ell'}$. It starts at t^{a}_ℓ and end at $t^{\mathrm{d}}_{\ell'}$. A *route* r is a cyclic sequence of legs such that there is always a plane connection between two consecutive legs of the sequence. Note that each route is associated with exactly one fleet. A route r might wind over the time period multiple times. We denote by w_r the winding number of a route r.

The problem will consist in finding a collection R of disjoint routes ("aircraft routing") such that simultaneously

- $\sum_{r \in \mathsf{R}_k} w_r \leqslant n_k$ for each $k \in K$ where R_k is the set of routes of fleet k (it means that there are enough planes in each fleet k to perform the schedule),
- the revenue is maximal,
- two extra families of constraints, namely the frequency constraints and the gate constraints, are satisfied.

These last constraints still need to be introduced. To ease this, we denote by L_{R} the set of legs selected by the routes in R.

Part of the input is also given by a cover $(L_f)_{f \in \mathcal{F}}$ of L. An f in $\mathcal{F}$ gathers legs from the same departure airport to the same arrival airport, over a time interval. (Each L_f reflects the wish of the airline to operate a leg from an origin to a destination, on a certain period of time, with a certain fleet. The L_f are actually the typical output of the schedule design and fleet assignment problems.) The *frequency constraints* write:

$$\left|L_{\mathsf{R}} \cap L_f\right| = 1, \quad \forall f \in \mathcal{F}. \tag{2.1}$$

We are given a time length δ which corresponds to the following. A plane connection is *short* if it lasts less than δ and *long* otherwise. A short connection requires the plane to stay at a same gate over the whole interval $\left(t_{\ell}^{\mathrm{a,gate}}, t_{\ell'}^{\mathrm{d,gate}}\right)_T$. A long connection allows the plane to only occupy a gate during the two time intervals $\left(t_{\ell}^{\mathrm{a,gate}}, t_{\ell}^{\mathrm{a}}\right)_T$ and $\left(t_{\ell}^{\mathrm{d}}, t_{\ell'}^{\mathrm{d,gate}}\right)_T$. (In this latter case, the plane is using a "distant parking.") We denote by $\mathsf{R}_a^{\mathrm{s}}$ (respectively $\mathsf{R}_a^{\mathrm{l}}$) the set of short (respectively long) plane connections in R in the airport a. We are moreover given a number γ_a of available gate slots for each airport $a \in \mathcal{A}$, and the *gate constraints* write, for all $t \in [T]$, and all $a \in \mathcal{A}$:

$$\left|\left\{(\ell,\ell') \in \mathsf{R}_a^{\mathrm{s}} \middle| t \in \left(t_{\ell}^{\mathrm{a,gate}}, t_{\ell'}^{\mathrm{d,gate}}\right)_T\right\}\right| + \left|\left\{(\ell,\ell') \in \mathsf{R}_a^{\mathrm{l}} \middle| t \in \left(t_{\ell}^{\mathrm{a,gate}}, t_{\ell}^{\mathrm{a}}\right)_T \cup \left(t_{\ell}^{\mathrm{d}}, t_{\ell'}^{\mathrm{d,gate}}\right)_T\right\}\right| \leqslant \gamma_a. \tag{2.2}$$

(Finding an assignment of each selected leg to a gate would seem more relevant from an operational perspective; yet, encoding this constraint simply as a capacity constraint is relevant; see Remark 5 below.)

Remark 1. The problem is introduced with the assumption that the revenue is additive over the legs. However, in practice, it is not the case and the revenue generated by the sale of a itineraries to customers is the result of a more complicated formula. The leg-based revenue assumption done for this problem is therefore an approximation. In this chapter, we do not discuss this assumption, but we will investigate the itinerary-based revenue generation in Chapter 3 and use this revenue estimation with the aircraft routing problem in Chapter 4. The linear objective we use in this chapter is however interesting in itself. Indeed, this kind of model is often used by the companies in order to quickly estimate the revenue or to generate a feasible solution. Moreover, in Chapter 4, we will need a practically efficient algorithm for this problem because we need to solve it as a subroutine of a Benders decomposition algorithm.

Remark 2. In the problem introduced, we consider that the gate management is a constraint on every airport. However an airline usually does not want to model this constraint on every airport; It wants to model the constraint on the airports that need to manage many planes at the same time: for instance the hub of the company. The case where the gate constraints do no need to be modeled on an airport a can easily be included in our model by setting $\gamma_a = +\infty$. In that case, the time intervals associated with the gate occupation at the airport can be considered empty. In practice, the gate occupation time of a plane can therefore be either empty (when we do not consider gate management at the airport), or a single interval (when the plane stays at a same gate for whole connection), or a pair of intervals (when distant parking is used). Note that the problem does not allow the passengers to board or unboard directly on the tarmac, which is indeed something airlines wish to avoid completely.

Remark 3. The formulation of the problem is quite flexible: the set $\mathcal{F}$ is a way to model the possibility for the airline to slightly change the departure and arrival times of the planes so that the number of gates occupied is always under the limit.

Remark 4. Aircraft routing is generally considered at a fleet level, because there is no constraint that binds the solutions of the aircraft routing for the different fleets. However, in our case, the gate slots are shared between all fleets, and the problem no longer decomposes. Consequently, we consider all the fleets at the same time in our problem.

Remark 5. In real-life, once a solution of the problem has been found, it remains to assign the gates to the planes simultaneously present at the hub. This problem can easily be modeled as a circular arc coloring problem with γ colors. However, this problem does not enjoy in general the integral min-max property, which means that it could happen that even if the gate constraints are satisfied, there is no feasible assignment to the gates. Yet, in our case, we can often choose γ to be equal to the number of gates. Indeed, it is very likely that there is a moment in the week where no plane requires to be assigned to a gate—in which case the problem consists in coloring an interval graph, which is perfect. Even if this situation is not met, classical results for circular arc colorings, initiated by Tucker [137], ensures that γ can be chosen really close to the number of gates. For instance, a result by Valencia-Pabon [138] shows that, denoting by n the minimum number of arcs needed to cover the circle, the circular arc graph can be colored with at most $\lceil \frac{n-1}{n-2} N \rceil$ colors, where N is the largest number of arcs containing a common point in the circle. Since the number of gate occupation slots usually needed to cover a week is at least 24 in practice, the result implies that using $\gamma = 50$ for a company with a hub endowed with 53 gates is sufficient in the worst case. Moreover, a simple polynomial greedy algorithm finds such a coloring.

2.4 Complexity result

In this section, we focus on the feasibility version of the problem, which consists in deciding whether there exists at least one feasible solution. Note that the problem is clearly in NP.

By setting $\{L_f \mid f \in \mathcal{F}\}$ to be the collection of all singletons $\{\ell\}$ with $\ell \in L$, we get a special case of the problem. In that case, all legs must be selected, as in the standard aircraft routing problem. So, somehow, we "deactivate" the frequency constraints. Similarly, setting $\gamma_a = +\infty$ for all $a \in \mathcal{A}$, we "deactivate" the gate constraints. The first complexity result deals with these "deactivations."

Proposition 1. *The feasibility version of the problem is polynomial when the frequency and gate constraints are "deactivated."*

Proof. In this special case, there is no interaction between the fleets and we can solve the problem for each fleet independently. Thus, without loss of generality, we assume that there is only one fleet and we denote by n the number of planes in this fleet. Consider the graph whose vertex set is $\mathcal{A} \times \mathcal{T}$, where $\mathcal{T} = \{t_\ell^{\mathrm{d}}, t_\ell^{\mathrm{a}} \mid \ell \in L\}$, and whose arc set is formed by L and all possible arcs $\big((a_\ell^{\mathrm{a}}, t_\ell^{\mathrm{a}}), (a_{\ell'}^{\mathrm{d}}, t_{\ell'}^{\mathrm{d}})\big)$ for legs $\ell \neq \ell'$ such that $a_\ell^{\mathrm{a}} = a_{\ell'}^{\mathrm{d}}$. We can see each arc as a portion of the circle of length T representing the cyclic time interval $[T]$: each arc occupies the portion corresponding to its time realization. (Note that we do not have to discretize this circle, with all T time steps; this could actually be non-polynomial in some cases.) Let A_0 be the set of arcs "crossing" an arbitrary generic point on the circle. Feasible solutions correspond to circulations satisfying the following two properties: the value of the circulation on every arc in L is 1; the total value of the circulation over all arcs in A_0 is at most n.

The existence of such a circulation can be performed in polynomial time: the first property is dealt with via capacities on the the arcs; the second one is achieved by setting a cost of 1 on all arcs in A_0 and of 0 elsewhere, and by finding a minimum-cost circulation. □

When the frequency or gate constraints are "activated," the problem becomes difficult, as expected. For instance, when $\gamma_a = +\infty$ for all $a \in \mathcal{A}$, it is not too difficult to reduce the CYCLIC INTERVAL COLORING PROBLEM (which is NP-complete [59]) to our problem (the colors become the fleets, with one plane in each fleet; intervals become round-trip between a hub and specific airports). We can strengthen this result.

Theorem 2. *The feasibility version of the problem is NP-hard, even when $|\mathcal{K}| = 1$, when there is a time $t \in [T]$ such that all routes must be in the same airport, and one of the following situations is met:*

1. $\gamma_a = +\infty$ *for all* $a \in \mathcal{A} \setminus \{h\}$ *and* $\gamma_h = 1$*, where h is a particular airport in* $\mathcal{A}$.
2. $n_1 = 1$, $\gamma_a = +\infty$ *for all* $a \in \mathcal{A}$.

Theorem 2 states that the extended aircraft routing we consider is NP-complete in two different ways. Notice that NP-hard is to be understood in the sense of Garay and Johnson [58], since the problem used for the reduction is used as a black box in the proof of Item 2. Item 1 states the NP-completeness of the case with only one airport with activated gate constraints (which can correspond in practice to the hub of the company). Item 2 states that even without gate constraints, the frequency constraints suffice to make the problem difficult for a single plane.

The proof consists in proving each item separately.

Proof of item 1. We introduce the INDEPENDENT SET PROBLEM on 2-union graphs (edge-wise union of two interval graphs). A set of vertices $U \subset V$ in a graph $G = (V, E)$ is *independent* if any two vertices in U are non-adjacent.

INDEPENDENT SET PROBLEM
Input: Two interval graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$. An integer N.
Output: (The existence of) N independent vertices in the graph $G = (V, E_1 \cup E_2)$.

Van Bevern et al. [139] proved that the INDEPENDENT SET PROBLEM on 2-union graphs is NP-complete.

Let $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, and N be an instance of INDEPENDENT SET PROBLEM. We compute a representation of G_1 and G_2 such that the intervals have pairwise distinct endpoints in $[2|V|]$. We denote those endpoints $t_1(v)$ and $t'_1(v)$ for G_1 and $t_2(v)$ and $t'_2(v)$ for G_2. We build the following instance of the extended aircraft routing problem.

Let $\mathcal{A}$ be V together with three extra airports h_1, h_2 and h_3. We consider only one fleet, $\mathcal{K} = \{1\}$ with $n_1 = |V|$ planes, and we set $\gamma_a = +\infty$ for all $a \in \mathcal{A} \setminus \{h_1\}$ and $\gamma_{h_1} = 1$ (there are only gate constraints for h_1). We set moreover $T = 4|V| + 5$ and $\delta = 0$. Each $v \in V$ gives rise to exactly 4 legs:

- $l_1(v)$ such that its departure airport is $a^{\mathrm{d}}_{l_1(v)} = h_1$, its departure time $t^{\mathrm{d}}_{l_1(v)} = t_1(v)$, its arrival airport $a^{\mathrm{a}}_{l_1(v)} = v$, and its arrival time $t^{\mathrm{a}}_{l_1(v)} = 2|V|+1$. We set $t^{\mathrm{d,gate}}_{l_1(v)} = t'_1(v)$ and $t^{\mathrm{a,gate}}_{l_1(v)}$ arbitrarily (there is no gate constraint).
- $l_2(v)$ such that its departure airport is $a^{\mathrm{d}}_{l_2(v)} = v$, its departure time $t^{\mathrm{d}}_{l_2(v)} = 2|V|+4$, its arrival airport $a^{\mathrm{a}}_{l_2(v)} = h_1$, and its arrival time $t^{\mathrm{a}}_{l_2(v)} = 2|V|+4+t'_2(v)$. We set $t^{\mathrm{d,gate}}_{l_2(v)}$ arbitrarily (there is no gate constraint) and $t^{\mathrm{a,gate}}_{l_2(v)} = t_2(v)+2|V|+4$.
- $l_3(v)$ such that its departure airport is $a^{\mathrm{d}}_{l_3(v)} = h_1$, its departure time $t^{\mathrm{d}}_{l_3(v)} = 1$, its arrival airport $a^{\mathrm{a}}_{l_3(v)} = h_2$, and its arrival time $t^{\mathrm{a}}_{l_3(v)} = 2|V|+1$. We set $t^{\mathrm{d,gate}}_{l_3(v)} = 1$ (there is no gate occupation time) and $t^{\mathrm{a,gate}}_{l_3(v)}$ arbitrarily (there is no gate constraint).
- $l_4(v)$ such that its departure airport is $a^{\mathrm{d}}_{l_4(v)} = h_3$, its departure time $t^{\mathrm{d}}_{l_4(v)} = 2|V|+4$, its arrival airport $a^{\mathrm{a}}_{l_4(v)} = h_1$, and its arrival time $t^{\mathrm{a}}_{l_4(v)} = 4|V|+5$. We set $t^{\mathrm{d,gate}}_{l_4(v)}$ arbitrarily (there is no gate constraint) and $t^{\mathrm{a,gate}}_{l_4(v)} = 4|V|+5$ (there is no gate occupation time).

We also build $|V|-N$ legs l' with a departure airport $a^{\mathrm{d}}_{l'} = h_2$, a departure time $t^{\mathrm{d}}_{l'} = 2|V|+2$, an arrival airport $a^{\mathrm{a}}_{l'} = h_3$, an arrival time $t^{\mathrm{a}}_{l'} = 2|V|+3$ and gate occupation times defined arbitrarily.

We set $\mathcal{F}$ in the following way. For each for each $v \in V$, we set two sets $L_f = \{l_1(v), l_3(v)\}$ and $L_{f'} = \{l_2(v), l_4(v)\}$. For each $i \in [|V|-N]$, we also add the sets $L_f = \left\{l'_i\right\}$ (each leg of the form l' must be used).

Suppose we have a solution of the INDEPENDENT SET PROBLEM. There exist N independent vertices $v_1, \ldots, v_N$ on G. We build R as follows. For every $i \in [N]$, we build one route associated with the vertex v_i that uses the legs $l_1(v_i)$ and $l_2(v_i)$. By construction, for every pair $j \neq j' \in [N]$, the occupations of the gate at h_1 by $l_1(v_j)$ and $l_1(v_{j'})$ cannot intersect (otherwise v_j and $v_{j'}$ would not be independent). Similarly, the occupations of the gate at h_1 by $l_2(v_j)$ and $l_2(v_{j'})$ do not intersect. For each vertex v in the $|V|-N$ remaining, we build a route of the form $l_3(v)$, l'_i and $l_4(v)$ with no gate occupation. Thus the set of legs does not use more than one gate and verifies all the frequency constraints. The routes generated are operated by $|V|$ planes.

We suppose given a solution of the extended aircraft routing problem. In the solution all the l'_i are used because of the frequency constraints. Therefore $|V|-N$ legs of the form l_4 are used, and thanks to the frequency constraints, N legs of the form l_2 are used. Therefore, at time $4|V|+5$ all the planes must be at airport h_1. Thus, the routes can be split in $|V|$ routes that last T and that use one plane each. The frequency constraints enforce the fact that $|V|-N$ planes take a route of the form l_3, l', l_4 that do not use any gate slots. The gate occupation time intervals of the legs $l_1(v)$ and $l_2(v)$ for all $v \in V$ generate two graph intervals G_1 and G_2. The frequency constraints makes that we have exactly N elements of V such that the legs $l_1(v)$ and $l_2(v)$ are used. Since there is only one gate available at the hub, the set of vertices K is independent in G_1 and independent in G_2. This set of vertices is independent on $G = (V, E_1 \cup E_2)$ and therefore, a solution of the aircraft routing problem generates a solution of the INDEPENDENT SET PROBLEM. □

Proof of item 2. We perform a polynomial reduction from the JOB INTERVAL SELECTION PROBLEM which is defined as follows.

JOB INTERVAL SELECTION PROBLEM

Input: $I_1, \dots, I_n$ sets of intervals, each of size m. An integer J. Each interval is represented as a starting time and an ending time.

Output: Find a set of intervals $\mathcal{J}$ such that $|\mathcal{J} \cap L_i| \leqslant 1$ for all i, $|\mathcal{J}| \geqslant J$, and two intervals in $\mathcal{J}$ do not intersect.

The JOB INTERVAL SELECTION PROBLEM has been proven to be NP-complete (see for instance Spieksma [124]). By binary search, deciding whether there is a set of size exactly J intervals in the problem is also NP-complete.

Let $I_1, \dots, I_n$ and J be an instance of the JOB INTERVAL SELECTION PROBLEM. We consider a representation as open intervals (t_i, t'_i) for $i \in [nm]$. Without loss of generality, we suppose that the endpoints of the intervals are integers, pairwise distinct and that $t'_i \geqslant t_i + 2$. We build an instance of the extended aircraft routing problem as follows.

Let $\mathcal{A}$ be a set of nm airports (numbered between 1 and nm) with two extra airports h_1 and h_2. We consider a unique fleet $\mathcal{K} = \{1\}$ with $n_1 = 1$ plane, and we set $\gamma_a = +\infty$ for all $a \in \mathcal{A}$ (there is no gate constraint). We set moreover $T = \max_i t'_i + 2(n - J) + 1$. Each interval I_i gives rise to exactly two legs, ℓ_i, ℓ'_i. We set the departure airport of ℓ_i as $a^{\mathrm{d}}_{\ell_i} = h_1$, its departure time as $t^{\mathrm{d}}_{\ell_i} = t_i$, its arrival airport as $a^{\mathrm{a}}_{\ell_i} = i$, and its arrival time as $t^{\mathrm{a}}_{\ell_i} = t_i + 1$. The gate arrival and departure times are fixed arbitrarily. Similarly we set the departure of ℓ'_i as $a^{\mathrm{d}}_{\ell'_i} = i$, its departure time as $t^{\mathrm{d}}_{\ell'_i} = t_i + 1$, its arrival airport as $a^{\mathrm{a}}_{\ell'_i} = h_1$, and its arrival time as $t^{\mathrm{a}}_{\ell'_i} = t'_i$. For each $p \in [n - J]$, we also build $2n$ legs as follows. for each $j \in [n]$, we build ℓ^p_{2j} and ℓ^p_{2j+1}. We set the departure airport of ℓ^p_{2j} as $a^{\mathrm{d}}_{\ell^p_{2j}} = h_1$, its departure time as $t^{\mathrm{d}}_{\ell^p_{2j}} = 2(nm+1) + 2j$, its arrival airport as $a^{\mathrm{a}}_{\ell^p_{2j}} = h_2$, and its arrival time as $t^{\mathrm{a}}_{\ell^p_{2j}} = 2(nm+1)+2j+1$. The gate arrival and departure times are fixed arbitrarily. Similarly we set the departure of ℓ^p_{2j+1} as $a^{\mathrm{d}}_{\ell^p_{2j+1}} = h_2$, its departure time as $t^{\mathrm{d}}_{\ell^p_{2j+1}} = 2(nm+1) + 2j + 1$, its arrival airport as $a^{\mathrm{a}}_{\ell^p_{2j+1}} = h_1$, and its arrival time as $t^{\mathrm{a}}_{\ell^p_{2j+1}} = 2(nm+1) + 2j + 2$. Finally, we build $\mathcal{F} = [3n - J]$. For each $j \in [n]$, the set L_j is formed by $\{\ell_i\}_{i \in [mj, \dots, m(j+1))}$ and $\left\{\ell^p_{2j}\right\}_{p \in [n-J]}$. For each $j \in \{n+1, \dots, 2n\}$, the set L_j is formed by $\left\{\ell'_i\right\}_{i \in [m(j-n), \dots, m(j-n+1))}$ and $\left\{\ell^p_{2(j-n)+1}\right\}_{p \in [n-J]}$. The remaining $n - J$ elements of $\mathcal{F}$ are of formed of the legs $\left\{\ell^p_{2j}\right\}_{j \in [n]}$ for all $p \in [n - J]$.

With this construction, a solution of the JOB INTERVAL SELECTION PROBLEM naturally gives a solution of the extended aircraft routing problem. Indeed, the J intervals selected correspond to the legs used by the only plane available to go to the J corresponding airports. Those fulfill the corresponding $2J$ frequency constraints. The remaining frequency constraints are fulfilled with the selection of $n - J$ legs of the form ℓ^p.

Conversely, if we can find a route that fulfills all the frequency constraints with one plane, the selected legs that perform round trips to the different airports correspond to intervals that do not intersect (there is only one plane). There is exactly J such intervals selected since for each $p \in [n - J]$, one element has to be chosen in order to satisfy the frequency constraints. This leaves exactly J intervals that need to be been chosen among the n groups. □

Remark 6. Notice that a proof very similar also proves that the SELECTIVE GRAPH COLORING PROBLEM is NP-complete on interval graphs, where the problem is defined as follows.

SELECTIVE GRAPH COLORING PROBLEM
Input: A graph $G=(V,E)$, a partition of V, $(V_1,V_2,\dots,V_n)$, and integer K.
Output: (The existence of) a set of vertices W, so that $|W\cap V_i|=1,\ \forall i\in[n]$, and the graph induced from G with W has a K-coloring.

This problem is NP-complete has been studied for instance by Demange et al. [40] and is NP-complete in the general case. The JOB INTERVAL SELECTION PROBLEM can be seen as a SELECTIVE COLORING PROBLEM on an interval graph with only one color.

2.5 Compact MILP

In this section, we propose a compact MILP that models the extended aircraft routing problem. To that purpose, we define a directed graph $D=(V,E)$ as follows.

The vertex set of D is given by

$$V=(\mathcal{A}\cup\mathcal{A}^{\text{distant}})\times[T]\times\mathcal{K},$$

where $\mathcal{A}^{\text{distant}}$ is a copy of $\mathcal{A}$. (It will allow to capture the "distant parking" alternative.) We denote by a^{distant} the copy of the airport a.

The set E of arcs is formed by two types of arcs. Each leg ℓ in L gives rise to one arc between each of the following pairs of vertices:

- $\left(a_\ell^{\text{d}},t_\ell^{\text{d}},k_\ell\right)$ and $\left(a_\ell^{\text{a}},t_\ell^{\text{a}},k_\ell\right)$,
- $\left(a_\ell^{\text{d}},t_\ell^{\text{d}},k_\ell\right)$ and $\left(a_\ell^{\text{a,distant}},t_\ell^{\text{a}}+\delta,k_\ell\right)$,
- $\left(a_\ell^{\text{d,distant}},t_\ell^{\text{d}},k_\ell\right)$ and $\left(a_\ell^{\text{a}},t_\ell^{\text{a}},k_\ell\right)$,
- $\left(a_\ell^{\text{d,distant}},t_\ell^{\text{d}},k_\ell\right)$ and $\left(a_\ell^{\text{a,distant}},t_\ell^{\text{a}}+\delta,k_\ell\right)$.

In addition to these arcs, we have also arcs of the form $((a,t,k),(a,t+1,k))$ for all $a\in\mathcal{A}\cup\mathcal{A}^{\text{distant}},k\in\mathcal{K},t\in[T]$. An example of a graph build this way is given in Figure 2.1.

We define moreover

$$o_t^{e,a}=\begin{cases}1 & \text{if } e \text{ is associated with a leg } \ell \text{ such that } a=a_\ell^{\text{d}} \text{ and } t\in\left(t_\ell^{\text{d}},t_\ell^{\text{d,gate}}\right)_T,\\ 1 & \text{if } e \text{ is associated with a leg } \ell \text{ such that } a=a_\ell^{\text{a}} \text{ and } t\in\left(t_\ell^{\text{a,gate}},t_\ell^{\text{a}}\right)_T,\\ 1 & \text{if } e \text{ is an edge of the form } ((a,t,k),(a,t+1,k)),\\ 0 & \text{otherwise.}\end{cases}$$

We also define T_1^k the set of all arcs associated with a fleet k that correspond to a leg or a connection that operates during the first time step of the schedule. Then the problem can be formalized as follows:

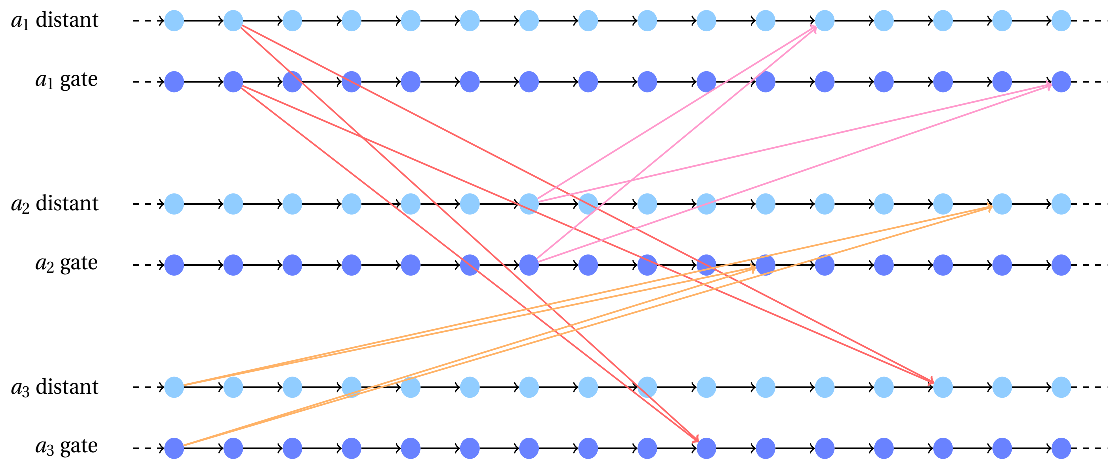

Figure 2.1 – Example of the graph built with 3 airports and 3 legs.

$$\max \sum_{e\in E} c_e x_e \tag{2.3a}$$

$$\text{st: } \sum_{e\in\delta^+(v)} x_e = \sum_{i\in\delta^-(v)} x_e \qquad \forall v \in V \tag{2.3b}$$

$$\sum_{e\in E} o_t^{e,a} x_e \leqslant \gamma_a \qquad \forall t \in [T], \forall a \in \mathcal{A} \tag{2.3c}$$

$$\sum_{e\in T_1^k} x_e = n_k \qquad \forall k \in \mathcal{K} \tag{2.3d}$$

$$\sum_{e\in L_f} x_e = 1 \qquad \forall f \in \mathcal{F} \tag{2.3e}$$

$$x_e \in \{0,1\} \qquad \forall e \in E. \tag{2.3f}$$

As stated in Remark 1, the objective (2.3a) is simply the sum over all the edges of the costs of the arcs of the graph. Notice that the costs c_e for $e \in E$ are simply the costs c_ℓ of the leg ℓ that e represents.

Constraint (2.3b) is a flow constraint on the graph. It ensures that the solution found corresponds to a set of routes and can therefore be operated by the company.

Constraint (2.3c) enforces that the solution generated by the problem will not use more than γ_a gate slots at any time of the schedule for each airport $a \in \mathcal{A}$. By definition, the binary values $o_t^{e,a}$ link each arc of the graph with the gate occupation at time t and at airport a, which ensures that Equation (2.2) is satisfied by the solution (the constraint being linear).

Constraint (2.3d) limits the number of planes for each fleet $k \in \mathcal{K}$ to its limit n_k. Since the set T_1^k contains all the arcs of the graph such that a plane using this arc will be occupied between time step 1 and 2, this constraint ensures that the routes used will not need more than the number of available planes.

Constraint (2.3e) ensures that the frequency constraint given by Equation (2.1) is satisfied. With a slight abuse of notation, the set L_f stands for the edges in the graph corresponding to a leg in L_f. That way, for each frequency constraint, the solution generated by the flow passes through exactly one edge for each frequency.

Remark 7. In practice, it might happen that in the schedule design, the company wants to see if it is possible to include a certain frequency (corresponding to a set of legs) in the schedule, without being mandatory to choose a leg in the set, as presented in the problem. The model presented makes this kind of constraint easy to include, by simply adding the corresponding edges in the graph and a constraint similar to Constraint (2.3e) with $\leqslant$ in place of $=$.

Remark 8. Notice that in the case where the gate constraints need to be modeled only on some airports, the creation of the graph described can easily be adapted. Indeed, for an airport without the management of gates, the “distant” copy of the associated vertices are not created and the arcs directed to this kind of distant vertices are not created either.

Remark 9. The aircraft routing problem we consider does not include maintenance constraints. Since the model presented to solve it has a small size in the entries of the problem, some maintenance constraints could be included. Some usual maintenance constraints enforce the fact that planes should have a maintenance every d days. Parmentier and Meunier [104] propose a graph construction consisting in having d copies of the graph of flights, and linking the different copies so that the legs used reflect the time since the last maintenance. An interesting enhancement to the model could be to apply this kind of technique to our model.

2.6 Experiments

2.6.1 Instances

In order to benchmark the performances of the model presented in Section 2.5, we consider different instances of the problem. All those instances aim at being realistic instances. Therefore, we consider a time horizon of a week, with a discretizeation every 15 minutes ($T = 672$). Since in practice, the gate constraints are limiting for the company exclusively in the hub of the company, our instances have a gate constraint active on a single hub airport. Therefore we always consider an airport $h \in \mathcal{A}$ and $\gamma_a = +\infty$ for all $a \in \mathcal{A} \setminus \{h\}$.

We consider two kinds of instances, based on two different use cases encountered in practice by the company. The first one reflects the instances that are generated when considering long-haul flights for the company. In that case, since long-haul flights take in general a long time, the legs considered in the problem are exclusively round trips from the hub of the company. The second type reflects a schedule that includes mid-haul and short-haul flights for the company. In that case, the legs can link two airports that are not the hub. Therefore, this second kind of instances is more general than the first one.

2.6.2 Results

The numerical experiments have been performed on a computer with 7.7 Gb RAM, 8 cores at 1.9 GHz. The algorithms have been developed in `Julia` [19] with the modeling library `JuMP` [44]. All linear programs have been solved using `Gurobi 9.1` [64].

Instance				MILP (2.3)		
$\|L\|$	$\|\mathcal{F}\|$	n	γ_h	objective	time (s)	memory (Mb)
860	179	16	8	241083	0.11	20
			6	241083	0.15	21
			4	240965	0.11	20
			2	236280	0.17	20
1737	334	25	9	457755	0.20	25
			7	457755	0.18	25
			5	457547	0.20	25
			3	452918	0.26	25
3373	708	44	10	945624	0.28	32
			8	945624	0.30	32
			6	944930	0.28	32
			4	936334	0.61	32
6538	1368	75	11	1826975	0.47	45
			9	1825390	0.53	45
			7	1820332	0.86	45
			5	1794218	3.55	45

Table 2.1 – Performances of Model (2.3) on mid-haul type of instances.

Table 2.1 gathers the results of Gurobi used on on different mid-haul instances of MILP (2.3). On each line, a different instance is given. The instances are gathered by groups of 4 which correspond to the same instance, with only the number of gates changing. The first four columns describe the instances in more details. Namely, they gather the number of legs $|L|$ in the model, the number of frequency constraints $|\mathcal{F}|$, the number of planes used n, and the number of gate slots available at the hub of the company γ_h. The three following columns present the results. The first one gives the objective value obtained, the second one gives the time (in seconds) needed to solve the problem to optimality and the last column provides the memory used by the program (in Mb).

Table 2.2 has exactly the same structure as Table 2.1. The only difference lies in the instances which are long-haul type of instances.

2.6.3 Comments

As expected, when the number of gates available decreases, the objective function is impacted and solving the problem takes more time. This phenomenon is more important as the size of the problem increases in size. It is particularly important when the number of gates gets near the minimal possible value (in the tables, the last line for each example corresponds to the minimal number of gate slots needed for the problem to be feasible).

As we can see on randomly generated instances, our generalized aircraft routing problem can be solved even with restrictive gate constraints. The impact of the gate constraints is more important on long-haul type of instances, which is expected since it is formed by round trips to the hub of the company, and therefore the concentration of planes in the hub is more important than in mid-haul type of instances.

Instance				MILP (2.3)		
$\|L\|$	$\|\mathcal{F}\|$	n	γ_h	objective	time (s)	memory (Mb)
690	106	9	5	152951	0.13	20
			4	152762	0.18	21
			3	151688	0.17	20
			2	148221	0.20	20
1465	266	26	10	376466	0.24	26
			8	376329	0.23	26
			6	374799	0.35	26
			4	368434	0.41	26
2900	482	34	11	678879	0.51	35
			9	677995	0.51	35
			7	674790	0.60	35
			5	657293	1.72	35
7505	1236	73	17	1764743	1.46	64
			15	1760406	1.48	63
			13	1750285	1.49	63
			11	1713613	5.70	63

Table 2.2 – Performances of Model (2.3) on long-haul type of instances.

These experiments also show that instances with a real life size of MILP (2.3) that can be solved in reasonable time.

2.7 Bibliographical remarks on slot management

A particularity of the aircraft routing problem with flight retiming we have investigated in this chapter is the inclusion of the constraints about the number of slots available in the airports of the company. The management of slots for aircraft has been a focus mostly from the airport perspective. In the present section, we provide an overview of the literature on the slot management problem in this context. From the airport perspective, the slot allocation is usually split into two types of problems the mid-term allocation and the short-term allocation. Since the routing problem for airlines is a mid-term problem, we do not present any short-term allocation model but Harsha [67] provides a general overview of this problem.

Most of the literature focuses on the auction mechanism followed in order to attribute the slots available to the different companies. The optimization of slot allocation can have several objectives and is generally formulated with a multi-criteria objective function. Vaze and Barnhart [142] present a model that maximizes airline profitability and passenger welfare while minimizing delays. Jacquillat and Odoni [72] develop an integer program that takes an existing schedule and modifies it while taking into account airport demand and capacity, preserving the connectivity of existing itineraries; and minimizing the changes to flight schedule. Pyrgiotis and Odoni [107] present a model that adapts existing schedules in order to enforce new limits. Ribeiro et al. [109] have recently developed a model for airports that optimizes the slot allocation decisions based on slot availability and airline slot requests. This model incorporates the International Air Transport Association guidelines and minimizes the number of slots displaced and the number of requests rejected.

To our knowledge, few models were developed with an aircraft route management consideration. This is partially because most of the slot optimization literature focuses on single-airports models. Some work has been dedicated to network-wide slot allocation. For instance, Bertsimas et al. [18] propose a traffic flow optimization that takes into account the capacity in airports. Castelli et al. [30] reuse a variant of this model and propose mechanisms that adapt the created schedule. Corolli et al. [37] adopt a stochastic approach and present two models that allocate time slots on multiple airports at the same time. Harsha [67] develops a model that takes the congestion in the airports into account: a congested airport will be time discretized a lot, while a non-congested airport will be associated with only a few nodes. Those network-wide models focus explicitly on slot allocation for airports and the model they use are developed from a global management perspective.

3 Demand models and seat allocation

3.1 Introduction

Evaluating the revenue generated by an aircraft schedule is not easy. In an airline, a whole revenue management department usually manages the prices of the tickets in order to optimize the revenue of the company. The problem of revenue management focuses on choosing which tickets should be offered for sale, at which price. In order to have the best possible selling strategy, the company needs to evaluate the demand on every itinerary, to gather information about the possible competitors, and to manage the resource used to sell those tickets. Advanced topics such as overbooking and accurate evaluations of customer preferences are even included in the models used and specialized algorithms are developed to manage the revenue management. Companies use such revenue management algorithms to optimize the profit generated with their aircraft schedule. This optimization is usually updated on a daily basis, after several hours of computation.

When a company needs to estimate the revenue generated by an alternative selling strategy or a particular schedule, this revenue management process needs to be simulated. Simulating such a complex process can be done with various approaches, like Monte-Carlo evaluation, and industrial tools exist to evaluate different revenue management strategies (such as the passenger origin destination simulator tool used by MIT [106]). These approaches might take hours if not days in order to produce results and to benchmark a new revenue management policy. Therefore, they cannot be used in many situations. For instance, estimating the revenue of an alternative and hypothetical schedule with new flights for commercial prospects needs to be quicker. In a more general optimization process, when this kind of estimation needs to be run an important number of times, such simulation algorithms become intractable.

When a quick estimation is needed, companies associate a revenue with each leg (or flight) of the aircraft schedule. They suppose that the revenue generated by the schedule is the sum of those leg revenues. However, estimating the revenue that way is not precise enough to provide a reliable optimization criterion. In this chapter, we study a demand model that evaluates the revenue of a company with itinerary-based revenue and a discrete choice model. We introduce this itinerary-based model because the leg-based approximation has several drawbacks that we want to avoid. In particular two important facts are omitted.

First, the product that is sold by the company is not the flights as the approximation supposes, but the itineraries, which can be composed of several flights. This consideration highly influences the revenue estimation. At the same time, it has a strong impact over the revenue estimation, and is a computational challenge. The number of itineraries is by nature exponentially larger than the number of flights. This implies that an optimization taking the itineraries into account is more difficult than an optimization managing only the flights. Some phenomena arise when managing the sale of itineraries: When the company proposes several itineraries with the same characteristics, those itineraries are in competition with one another. This phenomenon is called cannibalization and makes the optimization problem non-linear. Indeed, cannibalization between itineraries need to be limited in order to maximize profit. Another phenomenon important to consider is the “spill and recover”. It designates the way customers report on less attractive itineraries when their first choice is not available. On top of those difficulties, it is important to remark that itineraries share the same resource: the seats in the flight operated by the company. Hence, selling an itinerary can consume the last seat on a flight that is used by other itineraries. This impacts the itinerary sales as selling an itinerary to a customer influences the sales on other itineraries. With a good selling strategy, when there exist several itineraries with the same origin-destination, it might be interesting to sell preferentially one that uses the least attractive flights.

Second, since customers buy itineraries, estimating the revenue of a flight properly requires to identify all the itineraries containing it. Without any model on the global pricing of the network, it is therefore very difficult to have a good idea of how much profit a flight might generate. It is especially true when the company wants to estimate the revenue generated by a flight that has never been proposed before (either because the characteristics of the flight change or because the flight is a new one). Without a proper way to estimate the revenue generated by a flight based on historical data, the leg-based estimation of the revenue lacks precision.

We now introduce the stochastic model considered in that chapter, which reflects the itinerary purchases of the customers. It will be our *atomic model.* It aims at evaluating the revenue quickly and tackles the two challenges mentioned before. It is atomic in the sense that we consider that each customer arrives one after the other. It is stochastic in the sense that each customer makes a choice randomly according to a discrete choice model. We are interested in the expected number of tickets sold on each itinerary at the end of the process.

To that end we consider the following stochastic itinerary-based customer dynamic. We suppose that each market (or origin-destination) has a given demand. Each market is composed of different itineraries (with the same origin-destination). When a customer arrives, we suppose that he chooses among the itineraries available on his market. His choice follows a given probability distribution over the itineraries in this market. As customers arrive and buy the available itineraries, the number of available seats decreases, until there is no more left for a given flight. From that moment on, we consider that the customers continue to arrive as before but that the company only sells the itineraries that have still seats available. This requires to recompute the probabilities of choices of the customers. Discrete choice models are handy in this context because, when using them, it is easy to evaluate and recompute those probabilities.

When airlines use revenue management to manage the sales of their itineraries, they may take decision at any time during the process. An airline can therefore control the list of itineraries for sale regularly. This level of control is important for achieving the optimal selling strategy.

However, in the context of a rapid estimation, this level of precision does not seem reasonable. In our setting, we simplify this hypothesis by supposing that the airline fixes beforehand the limit number of tickets sold for each itinerary. We suppose that after allocating seats between itineraries, the company has no more control over the sales. This is an important difference with the classical models of revenue management as they are modeled since the work of Talluri and van Ryzin [129]. In the classical revenue management problem the company has to choose at each time step which itineraries it will sell. Our modelling choice therefore undermines the real ability of the company to control the sales of itineraries. However, in the context of a quick estimation, it still captures the essential phenomena in revenue generation of the company.

Studying precisely the dynamic of this atomic model reveals to be quite challenging. In the perspective of evaluating quickly the revenue generated by a flight schedule it becomes too long to evaluate. In a similar context, Gallego et al. [56] have introduced a linear program—the *Sales Based Linear Program* (SBLP)—as an approximation of the "network revenue management problem," which shares similar stochastic and dynamic features with our problem. They were able to prove that, when the number of customers increase, the optimal value of the network revenue management problem converges to that of the SBLP. The main theoretical contribution of this chapter is a similar convergence result for our problem. Our atomic problem being more constrained than the network revenue management problem, their proof technique cannot be applied and a substantial part of our proof consists in studying the stochastic process of the customers arrivals in detail. A practical contribution of this chapter is an efficient algorithm based on column generation to solve the SBLP. From the convergence result, we expect that the solution of the SBLP provides a good solution for our atomic problem.

An important tool for proving the convergence result is an intermediate optimization problem between the atomic problem and the SBLP. In the proof, it is mostly a pure mathematical object. Yet, it admits a natural interpretation: it is the counterpart of the atomic problem when we replace the atomic arrival model by an approximate fluid arrival model. Another contribution of this chapter is a concentration inequality assessing the quality of this approximation.

The chapter is organized as follows:

- Section 3.2 introduces the atomic problem and the SBLP and states the convergence theorem mentioned above.
- Section 3.3 gives a general view of the literature on revenue management and discrete choice models that are close to our setting.
- Section 3.4 presents the column generation approach to solve the SBLP. It is based on a Dantzig–Wolfe decomposition.
- Section 3.5 gathers the numerical experiments and discusses the performance of the proposed algorithm for solving the SBLP.
- Section 3.6 contains the proof of the convergence theorem.
- Section 3.7 presents and proves complementary results. In particular, it provides the natural interpretation of the fluid approximation, and gives a proof of the concentration inequality.

3.2 Revenue estimation models and seat allocation optimization problems

3.2.1 Definition and notation

For an airline, the revenue management problem focuses on the sale of itineraries. When a customer wants to buy a ticket associated with an itinerary, he usually wants to go from a given origin to a given destination. We call the set of itineraries with the same origin-destination a *market*, and we denote by M the set of markets operated by the company. We also denote by I the set of all itineraries operated by the company. The markets $m \in M$ thus form a partition of I. The different itineraries are composed of legs operated by the company. We denote these legs by L. By construction, an itinerary $i \in I$ is a subset $i \subset L$ such that the legs that compose i can be practically used by a customer. This set of legs can be seen as issued from a flight schedule (for instance a flight schedule generated as presented in Chapter 2). For a leg $l \in L$, we denote by s_l the number of available seats, and by s the vector $(s_l)_{l \in L}$.

The behavior of customers is supposed to follow a *basic attraction model*, which has been introduced by Luce [91]. This means each itinerary is associated with a positive weight γ_i, and that the probability for a customer to choose itinerary $i \in m$ of a market $m \in M$ when the subset $J \subset m$ is offered is $\frac{\gamma_i}{\sum_{j \in J} \gamma_j}$.

The aim of the company is to set the number of itineraries it is going to sell so that it satisfies the limitation on the number of seats available. We denote by y_i the decision variable that fixes the capacity of i (which will be sold by the company). We suppose here that the company fixes the y_i for all $i \in I$ and that the number of seats available evolves with the arrival of customers with no further action of the company.

3.2.2 Atomic model

Consider a capacity vector $y = (y_i)_{i \in I}$ such that $\sum_{i \ni l} y_i \leqslant s_l$ for all $l \in L$ and $y_i \geqslant 0$ for all $i \in I$. We suppose that D_m customers arrive for each market $m \in M$. We consider that there are $T = \sum_{m \in M} D_m$ time steps. At each time step a customer arrives, the market he is interested in is selected randomly among the markets which have not reached D_m customers. He can then be satisfied by any itinerary $i \in m$. A scenario is the realization of a sequence of purchases by customers over the T time steps. The choices follow the basic attraction model of Section 3.2.1. We denote by $\Omega_{T,y}$ the set of all possible scenarios. (The subscripts y and T remind the dependency of the scenario set to the number of time steps and the capacity vector.) Denote by $Q_{i,t,y}$ the random variable counting the number of customers who have bought itinerary i by time t. (The dependency to the scenario $\omega \in \Omega_{T,y}$ is implicit throughout the chapter.)

Denoting by c_i the cost of itinerary i, the optimization problem based on the basic attraction model can be formulated as follows. We call it the *atomic problem*.

$$V^{\text{atom}} = \max \mathbb{E}\left[\sum_{m\in M}\sum_{i\in m} c_i Q_{i,T,y}\right] \tag{3.1a}$$
$$\text{st:} \quad \sum_{i|l\in i} y_i \leqslant s_l \qquad \forall l \in L \tag{3.1b}$$
$$y_i \geqslant 0 \qquad \forall i \in m, \forall m \in M. \tag{3.1c}$$
$$\tag{3.1d}$$

Problem (3.1) has a formulation that is very close to the network revenue management studied by Gallego et al. [56]. The major difference between the two problems lies in the decision variables. In the network revenue management problem, the set of products offered for sale can be changed at each time step while in the case of Problem (3.1), the decisions are taken before the arrival of customers and the sets offer for sale only depends on the arrival and the behavior of customers.

Gallego et al. [56] proved that the network revenue management problem can be approximated by the SBLP defined as follows.

$$V^{SBLP} = \max \sum_{m\in M}\sum_{i\in m} c_i q_i \tag{3.2a}$$
$$\text{st:} \quad \sum_{i|l\in i} q_i \leqslant s_l \qquad \forall l \in L \tag{3.2b}$$
$$\frac{q_i}{\gamma_i} \leqslant \frac{q_m^0}{\gamma_m^0} \qquad \forall i \in m, \forall m \in M \tag{3.2c}$$
$$q_m^0 + \sum_{i\in m} q_i = D_m \qquad \forall m \in M \tag{3.2d}$$
$$q_i \geqslant 0 \qquad \forall i \in m, \forall m \in M \tag{3.2e}$$
$$q_m^0 \geqslant 0 \qquad \forall m \in M, \tag{3.2f}$$

where each itinerary $i \in I$ is associated with the cost c_i, and D_m is the expected number of customers on each market $m \in M$.

A particular itinerary associated with the value q_m^0 appears in the SBLP. This itinerary is associated with the option “not buying ” for the customers. There is one such itinerary for each market. It has infinite capacity, a cost of 0, an attractiveness γ_m^0 and does not consume any seat available.

Their proof developed in [56] does not hold in the case of Problem (3.1). In Section 3.6, we provide a proof of the convergence of the atomic model for fixed capacity to the SBLP. It has then the following consequence, where V_θ^{atom} denotes the optimal value of the problem when the number of time steps and the capacities are multiplied by θ:

Theorem 3. *The following convergence holds:*

$$\liminf_{\theta\to\infty} \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant V^{SBLP}. \tag{3.3}$$

The customer arrival dynamic described by Problem (3.1) is close to the real process. However, solving precisely Problem (3.1) remains a very difficult task when the size of the problem grows. Therefore, Theorem 3 shows the SBLP provides a lower bound to the optimal value achievable when the number of customers is sufficiently large. It motivates the focus on the SBLP in the rest of the chapter.

3.3 Literature review

Two important challenges in our model are the need to model the behavior of customers accurately, and the management of resources on a network in the context of the revenue management problem. In the current section, we present a bird's eye view of the literature that tackles those challenges. Section 3.8 provides a deep dive in the related literature.

3.3.1 Discrete choice models

Discrete choice models are one of the main mathematical tool that are used to study the choices of a customer among a finite set of alternative. Those models propose different approaches to predict the behavior of customers based on historical data. Since we are interested in integrating the behavior of customers in the schedule optimization problem, we start by looking at the existing literature about discrete choice models.

The most classical model used is called the *random utility model* (first introduced by Thurstone [133]). We consider that a customer is proposed a set of n alternatives. The random utility model supposes that each alternative $i \in [n]$ is associated with a utility $u_i = v_i + \epsilon_i$, where v_i is a deterministic value based on the characteristics of alternative i and ϵ_i is a random variable. The general idea behind this definition is that the utility of an alternative can be split into a deterministic part which reflects the attractiveness of the alternative based on it characteristics (intuitively, a cheaper and faster alternative will be preferred to a more expensive one), and a random part which reflects the diversity and the variety in the choices.

The random utility model states that the probability p_i that a customer chooses alternative $i \in [n]$ has the form $p_i = \mathbb{P}\big(i = \arg\max_{k\in[n]}(v_k + \epsilon_k)\big)$. In order to provide a more precise formula of the probability of choice, a random utility model needs to decide which distribution rules the random part of the utility. Several distribution choices are possible. For tractability and computational reasons, the most widely used has been introduced by McFadden et al. [94] and uses a Gumble law. In that case, it can be shown that $p_i = \frac{\exp(\beta v_i)}{\sum_{j\in[n]}\exp(\beta v_j)}$. This random utility model is called the *multinomial logit model*.

In many operational applications, having a particular expression derived from the utility of each alternative is not a necessity. The probability only needs to have the form $p_i = \frac{\gamma_i}{\sum_j \gamma_j}$ to be used. That kind of models is called a basic attraction model and has been developed axiomatically by Luce [91].

The multinomial logit model is the most wildly used model in practice since it is can be easily estimated and stays tractable when coupled with an optimization problem ([39], [147], [110]). This model has limits that take root in the independence assumptions on the random variables (see Ben-Akiva et al. [14]). We give a deeper view on the existing choice models and their use in Section 3.8.

3.3.2 Assortment optimization

In this literature review, we are more specifically interested in the links that exist between the discrete choice models and the schedule optimization problem. The assortment problem reveals to be a key component in this matter. This problem aims at selecting the best assortment of product that can be offered to customers, in order to optimize the profit generated. Formally, the model supposes that customers behave following a discrete choice model, and the objective is to select which products should be offered so that the expected profit is maximal.

The multinomial logit model remains the most widely used discrete choice model in that case. Kök et al. [81] provide a comprehensive review on the assortment problem. It is still an active field of research. For instance, Wang [147] studies the capacitated assortment problem and price optimization under this model. Davis et al. [38] develop the links between the assortment problem and a linear program in that particular case.

Our atomic model fits in a variant of the problem called the dynamic assortment planning problem. Contrary to the classical assortment problem where the expected revenue is calculated over one time step, the dynamic version of the problem considers a longer time horizon. During this time, events like stock-out can occur, which dynamically change the assortment available for the customers. The dynamic assortment problem has been introduced in the seminal work of Mahajan and van Ryzin [93] and has recently received more attention. Honhon et al. [70] propose a dynamic programming algorithm to compute the optimal assortment. Topaloglu [135] develops the non-linear program formulation of the problem. He uses a decomposition method and solves it with a dynamic programming approach. He also give a deterministic version of the problem that models the fluid estimation we use in the chapter. This deterministic approximation has the same structure as the fluid approximation we consider. He proves that the deterministic approximation gets close to the stochastic model with a polynomial bound. In this chapter, we motivate the introduction of the fluid approximation differently, and provide a stronger bound on the difference between the atomic model and its approximation. Many recent work focus on approximation algorithms to solve the dynamic assortment problem. Goyal et al. [62] propose a polynomial time approximation scheme algorithm for the problem. Aouad et al. [6] and Aouad and Segev [7] respectively develop a greedy algorithm that approximates the solution and an approximation scheme that estimates the revenue with theoretical guarantees.

More recently, different choice models have been studied for the assortment problem, like Feldman and Topaloglu [48] with the Markov-chain choice model or Davis et al. [39] who use a nested multinomial logit model. Gallego and Topaloglu [57] provide a recent overview of the way discrete choice models have been used in the context of assortment optimization.

3.3.3 Revenue management

Revenue management in airlines has been a subject of studies for a long time. Initially, the problem has focused on the finding which fare classes should be open at every time step in order to maximize the expected revenue. Several algorithms have been developed, especially in the case of the management of sales of one leg with different fare classes. For instance, Belobaba [13] proposes a very efficient algorithm called EMSRb and Brumelle and McGill [24] develops an exact algorithm in the case of nested fare classes. McGill and van Ryzin [95] give a general review of the revenue management in this context.

A turning point in the literature is the paper of Talluri and van Ryzin [129]. They were among the first to introduce a revenue management model with discrete choice model. In that context, the problem is closer to the assortment problem: at each time step the company has to choose which assortment of product are going to be offered. The main difficulty comes from the fact that due to shared resources, all those assortment problems are linked. Talluri and van Ryzin [129] solves the revenue management problem on a single leg under the basic attraction model. This model has set the bases for the research in revenue management since then. Talluri and Van Ryzin [131]'s book is an important reference on revenue management.

Gallego et al. [55] study the revenue management problem under the assumption that the products sold share a common resource. They propose a dynamic programming formulation of this network revenue management problem. They also develop a relaxation that is called the choice-based deterministic linear program. Liu and van Ryzin [89] study theoretically this problem and prove convergence properties. Many approximations and extensions of this program have been proposed and are discussed in Section 3.8. Strauss et al. [126] present a general review on the subject of revenue management with customer models. We are interested in this chapter by the SBLP approximation problem introduced by Gallego et al. [56].

3.3.4 SBLP

In this chapter we propose a column generation approach to solve the SBLP. We want to solve this problem because it is an approximation of the dynamic assortment problem as Topaloglu [135] were the first to point out. Historically, the SBLP was introduced by Gallego et al. [56] as an approximation of the network revenue management.

Since its introduction, the SBLP has been reused in different contexts. Liang et al. [85] presented a two stages version of the SBLP called sales based quadratic program. This problem consists in solving the SBLP in the first stage and use this solution to solve a second stage quadratic problem able to differentiate the better solution among the optimal solutions of the first stage. Recently, Grani et al. [63] study a variation of the SBLP where the continuous variables are replaced by integer variables. They call this problem the sales based integer program and motivate the study of this problem by the practical difference existing between the integer problem and its relaxation. It is solved through a market decomposition and the authors propose a concave approximation to effectively produce a solution. Talluri [128] presents a program called SBLP+ where he uses a variation of the cuts proposed by Meissner et al. [97] to tighten a relaxation of the network revenue management problem they introduced. He proposes a version of the SBLP with cuts focusing on the intersection of the consideration sets of different customer.

3.4 Column generation approach

We have seen that solving Problem (3.1) can be approximated by solving the SBLP (3.2). In the present section, we focus on the practical resolution of the SBLP. In particular, since it is a linear program with a particular structure, we focus on a column generation approach to generate solutions of the SBLP quickly.

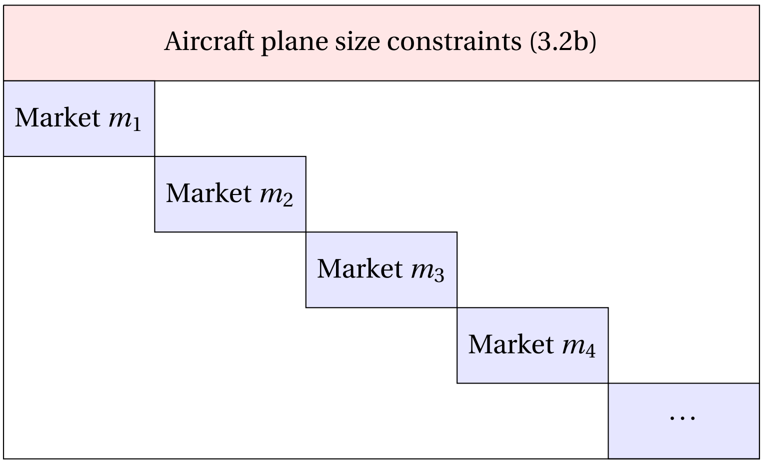

Figure 3.1 – Structure of the SBLP

3.4.1 Dantzig–Wolfe reformulation

As the size of the SBLP is very large on our instances, solving it frontally can be a challenge. It is possible to speed up the resolution of Problem (3.2) using a column generation with Dantzig–Wolfe reformulation of Problem (3.2).

The Dantzig–Wolfe reformulation aims at exploiting the structure of the SBLP. Indeed, when looking at the constraint structure of the SBLP, we can see that the constraints (3.2b) links the variables of all markets together. Without those constraints, the problem would be separable per market as constraints (3.2c) and (3.2d) only gather variables in the same markets. The form of the constraint matrix of Problem (3.2) has the particular structure illustrated in Figure 3.1.

From a mathematical point of view, the Dantzig–Wolfe reformulation is based on the Minkowski–Weyl theorem. Instead of representing the linear program with constraints (hyperplanes) the reformulation rather uses the extreme points of the demand constraints polyhedron.

Denote by G_m the set of extreme points of the polyhedron associated with the following constraints (where the variables are the q_i and q_m^0):

$$\begin{aligned}
& q_i \leqslant \frac{\gamma_i}{\gamma_m^0} q_m^0 && \forall i \in m \\
& \sum_{i \in m} q_i = D_m && \\
& q_i \geqslant 0 && \forall i \in m.
\end{aligned}$$

The SBLP can then be reformulated as follows, where c'_g is the cost of the extreme point $g \in G_m$ and β_g^l the constraint cost of the extreme point g for the constraint (3.2b) on leg l:

$$\max \sum_{m\in M}\sum_{g\in G_m} c'_g \mu_g \tag{3.4a}$$

$$\text{st:} \sum_{g\in G_m} \mu_g = 1 \qquad \forall m \in M \tag{3.4b}$$

$$\sum_{m\in M}\sum_{g\in G_m} \beta^l_g \mu_g \leqslant s_l \qquad \forall l \in L \tag{3.4c}$$

$$\mu_g \geqslant 0 \qquad \forall g \in G_m, \forall m \in M. \tag{3.4d}$$

3.4.2 Column generation for the SBLP

Problem (3.4) has too many variables to fit in a MILP solver. This motivates a column generation approach to find its optimal solution.

In the context of column generation, we start with a restriction of Problem (3.4) with only few variables, and iteratively add variables in order to converge toward the optimal value of Problem (3.4). At iteration $k \in \mathbb{N}$, for each $m \in M$, the set of variables considered, called columns, is denoted by $G_m(k) \subset G_m$, and the master problem solved is the following:

$$\max \sum_{m\in M}\sum_{g\in G_m(k)} c'_g \mu_g \tag{3.5a}$$

$$\text{st:} \sum_{g\in G_m(k)} \mu_g = 1 \qquad \forall m \in M \tag{3.5b}$$

$$\sum_{m\in M}\sum_{g\in G_m(k)} \beta^l_g \mu_g \leqslant s_l \qquad \forall l \in L \tag{3.5c}$$

$$\mu_g \geqslant 0 \qquad \forall g \in G_m(k), \forall m \in M. \tag{3.5d}$$

In order to generate new columns for the master problem, several strategies can be considered. A natural approach is to generate the variable with the largest reduced cost. We denote by λ_l the dual values of capacity constraints (3.5c). The pricing subproblem of the column generation which generates a new column is the following:

$$\max \sum_{m\in M}\sum_{i\in m} (c_i - \sum_{l\in i} \lambda_l) q_i \tag{3.6a}$$

$$\text{st:} \quad \frac{q_i}{v_i} \leqslant \frac{q^0_m}{v^0_m} \qquad \forall i \in m, \forall m \in M \tag{3.6b}$$

$$q^0_m + \sum_{i\in m} q_i = D_m \qquad \forall m \in M \tag{3.6c}$$

$$q_i \geqslant 0 \qquad \forall i \in m, \forall m \in M \tag{3.6d}$$

$$q^0_m \geqslant 0 \qquad \forall m \in M. \tag{3.6e}$$

3.4.3 Pricing subproblem

Problem (3.6) has the interesting property of being separable by market. Hence, it can be split into $|M|$ different problems. At each step of the column generation, the dual values λ_l, which are solution of the master problem at that step, change. This changes the objective values of the subproblems. For a systematic study, we denote the coefficients of the objective by $\tilde{c}_i$.

For a given market $m \in M$, we are interested in the following problem written in the standard form with the slack variables being denoted by ω:

$$\max \sum_{i \in m} \tilde{c}_i q_i \tag{3.7a}$$

$$\text{st:} \quad \frac{q_i}{\gamma_i} + \omega_i = \frac{q_m^0}{\gamma_m^0} \qquad \forall i \in m \tag{3.7b}$$

$$q_m^0 + \sum_{i \in m} q_i = D_m \tag{3.7c}$$

$$q_i \geqslant 0, \quad \omega_i \geqslant 0 \qquad \forall i \in m \tag{3.7d}$$

$$q_m^0 \geqslant 0. \tag{3.7e}$$

The previous problem is an LP and can be solved with classical algorithms as the simplex method or the interior point method. However, as it is pointed out by Gallego et al. [56], this problem can be seen as an assortment problem and there exists a better way to find an optimal solution. Especially, they prove the following result.

Proposition 4 ([56])**.** *Problem* (3.7) *can be solved in* $O(n \log n)$.

Up to sorting the itineraries, they assume the $\tilde{c}_i$ are ordered in a non-increasing way. Then they find the smallest index $i^\star$ defined as follows:

$$i^\star = \operatorname{argmin}\left\{ i \in m \,\middle|\, \frac{\sum_{j \leqslant i} \tilde{c}_j \gamma_j}{\gamma_m^0 + \sum_{j \leqslant i} \gamma_j} > \tilde{c}_{i+1} \right\}. \tag{3.8}$$

An optimal solution is obtained by setting $q_i = D_m \frac{\gamma_i}{\gamma_m^0 + \sum_{i \leqslant i^\star} \gamma_i}$ for $i \leqslant i^\star$ and $q_i = 0$ otherwise. The computation of $i^\star$ can be done in linear time by updating the numerator and the denominator iteratively.

While their proof relies on linear programming duality, we propose in this subsection an alternate approach for showing the correctness of this algorithm.

Lemma 5. *Suppose that $D_m > 0$. Let $J \subseteq m$. There exists a unique feasible basis of Problem* (3.7) *such that the variables q_i in the basis are exactly those with i in J. Moreover, for such a basis, we have $\omega_i = 0$ for all $i \in J$.*

Proof. It is clear that such a basis exists. We prove that in such a basis, we have necessarily $\omega_i = 0$ for $i \in J$. Uniqueness follows along the same line.

Consider the basic solution (q, ω) associated with this feasible basis. Note that there are $|m| + 1$ constraints in the problem, they are linearly independent, and thus a basis has size $|m| + 1$. Since $D_m > 0$, we necessarily have $q_m^0 > 0$. For $i \notin J$, we have $q_i = 0$ and thus $\omega_i > 0$. Moreover, the variable q_m^0 being present in the basis, there remain exactly $|J|$ variables to complete the basis, and they are necessarily the q_i for $i \in J$. □

Consider a basis as in the statement of Lemma (5). For $i \in J$, the lemma shows that $q_i = D_m \frac{\gamma_i}{\gamma_m^0 + \sum_{i\in J}\gamma_i}$. The value of the objective function on this basis is given by

$$D_m \frac{\sum_{i\in J} \tilde{c}_i \gamma_i}{\gamma_m^0 + \sum_{i\in J}\gamma_i}.$$

Therefore solving Problem (3.7) amounts to find the subset $J \subseteq m$ maximizing the quantity

$$\overline{c}_J = \frac{\sum_{i\in J} \tilde{c}_i \gamma_i}{\gamma_m^0 + \sum_{i\in J}\gamma_i}. \tag{3.9}$$

Lemma 6. *For every $i \in m$ and $J \subseteq m$, the quantity $\overline{c}_{\{i\}\cup J}$ is between $\tilde{c}_i$ and $\overline{c}_J$.*

Proof. We have

$$\begin{aligned}
\tilde{c}_i - \overline{c}_{\{i\}\cup J} &= \frac{\left(\gamma_0 + \sum_{j\in\{i\}\cup J}\gamma_j\right)\tilde{c}_i - \sum_{j\in\{i\}\cup J}\tilde{c}_j\gamma_j}{\gamma_0 + \sum_{j\in\{i\}\cup J}\gamma_j} = \frac{\left(\gamma_0 + \sum_{j\in J}\gamma_j\right)\tilde{c}_i - \sum_{j\in J}\tilde{c}_j\gamma_j}{\gamma_0 + \sum_{j\in\{i\}\cup J}\gamma_j} \\
&= \frac{\gamma_0 + \sum_{j\in J}\gamma_j}{\gamma_0 + \sum_{j\in\{i\}\cup J}\gamma_j}(\tilde{c}_i - \overline{c}_J) < (\tilde{c}_i - \overline{c}_J).
\end{aligned}$$

□

The following proposition shows the correctness of the algorithm proposed by Gallego et al.

Proposition 7. *Assume that the $\tilde{c}_i$ are ordered in a non-increasing way and let $i^\star$ be defined as above. The set $I^\star = \{1, \ldots, i^\star\}$ is a subset of m such that $\overline{c}_{I^\star}$ is maximum.*

Proof. Let $J \subseteq m$. We have $J = (I^\star \cup J) \setminus (I^\star \cap J)$. By definition of $i^\star$, the inequality $\tilde{c}_j \leqslant \overline{c}_{I^\star}$ holds for every $j > i^\star$. Applying repeatedly Lemma 6 for $j \in J \setminus I^\star$ in the decreasing order, we get then $\overline{c}_{I^\star\cup J} \leqslant \overline{c}_{I^\star}$. By this latter inequality, we have $\overline{c}_{I^\star\cup J} \leqslant \tilde{c}_j$ for every $j \in I^\star$. Applying repeatedly Lemma 6 for $j \in I^\star \cap J$ in the increasing order, we get $\overline{c}_J \leqslant \overline{c}_{I^\star\cup J}$. Therefore, $\overline{c}_J \leqslant \overline{c}_{I^\star}$. □

The algorithm proposed by Gallego et al. [56] is very efficient in practice. Sorting the itineraries in order to find a solution of this kind of problem seems to be a classical result in revenue management. However, to our knowledge, the methods to find the index of the optimal itinerary rely on a sorting procedure. We prove that in theory, finding a solution can be done in linear time (without explicitly sorting the itineraries).

Proposition 8. *Problem* (3.7) *can be solved in $O(n)$.*

The proof of this result highly relies on the linear time computation of the median in a list of numbers, which has been proved by Blum et al. [22]. We introduce the following recursive Algorithm 1, which moves itineraries in the list and returns the index $i^\star$. We point out here that the definition of $i^\star$ does not need all the itineraries to be sorted, but only all the itineraries with a smaller index having a cost c_i smaller than $c_{i^\star}$ (and the itineraries with a bigger index having a cost bigger than $c_{i^\star}$). The algorithm exploits this fact in order to run in linear time. We denote by f the recursive function used for the algorithm.

Algorithm 1 Recursive function $f(u, v, i_{\text{down}}, i_{\text{up}}, dir)$

If $i_{\text{down}} == i_{\text{up}}$
  If $dir ==$ up
    Return $i_{\text{up}} + 1$
  Else
    Return i_{down}
Find The median $\tilde{c}_{\text{med}}$ of the list $(\tilde{c}_i)_{i\in[i_{\text{down}}, i_{\text{up}}]}$.
Reorganize (in linear time) the indices so that $\tilde{c}_i > \tilde{c}_{\text{med}}$ for all $i \in [i_{\text{down}}, \text{med}]$.
If $dir ==$ up
  $u_{\text{current}} = \sum_{i\in[i_{\text{down}}, i_{\text{med}}[} \tilde{c}_i\gamma_i \quad ; \quad v_{\text{current}} = \sum_{i\in[i_{\text{down}}, i_{\text{med}}[} \gamma_i$
  If $\frac{u+u_{\text{current}}}{v+v_{\text{current}}} < \tilde{c}_{\text{med}}$
    Return $f(u + u_{\text{current}}, v + v_{\text{current}}, i_{\text{med}}, i_{\text{up}}, \text{up})$
  Else
    Return $f(u + u_{\text{current}}, v + v_{\text{current}}, i_{\text{down}}, i_{\text{med}}, \text{down})$
Else ($dir ==$ down)
  $u_{\text{current}} = \sum_{i\in[i_{\text{med}}, i_{\text{up}}[} \tilde{c}_i\gamma_i \quad ; \quad v_{\text{current}} = \sum_{i\in[i_{\text{med}}, i_{\text{up}}[} \gamma_i$
  If $\frac{u-u_{\text{current}}}{v-v_{\text{current}}} < \tilde{c}_{\text{med}}$
    Return $f(u - u_{\text{current}}, v - v_{\text{current}}, i_{\text{med}}, i_{\text{up}}, \text{up})$
  Else
    Return $f(u - u_{\text{current}}, v - v_{\text{current}}, i_{\text{down}}, i_{\text{med}}, \text{down})$

Proof of Proposition 8. The proof consists in proving that Algorithm 1 is a linear time algorithm and that it computes $i^\star$ (which is enough to solve Problem (3.7) by Proposition 7).

Algorithm 1 takes $O(|m|)$ steps. Indeed, a call of the function f with two indices i_1 and i_2 takes $O(i_2 - i_1)$ computations plus a call to f with two indices having a difference of $\frac{i_2-i_1}{2}$. Since f is called with indices 1 and $|m|$ to solve Problem (3.7), and denoting by C the complexity of f, we get that $C(|m|) = C(\frac{|m|}{2}) + O(|m|)$ and thus $C(|m|) = O(|m|)$.

We will now prove that the set $[i']$ where i' is returned by the computation of the quantity $f(0, \gamma_m^0, 1, |m|, \text{up})$ is optimal for Problem (3.7). At each step of the algorithm, we have $\tilde{c}_i > \tilde{c}_{\text{med}}$ for all indices i in $[i_{\text{med}-1}]$ and $\tilde{c}_i \leqslant \tilde{c}_{\text{med}}$ for all indices i in $\{i_{\text{med}}, \dots, |m|\}$. Moreover, by recursion, at each step of the algorithm, the quantity $\frac{u-u_{\text{current}}}{v-v_{\text{current}}}$ or $\frac{u+u_{\text{current}}}{v+v_{\text{current}}}$ is equal to $\frac{\sum_{i<i_{\text{med}}} \tilde{c}_i\gamma_i}{\gamma_m^0 + \sum_{i<i_{\text{med}}} \gamma_i}$.

Thus at the last iteration we have that $\frac{\sum_{j\leqslant i'} \tilde{c}_j\gamma_j}{\gamma_m^0 + \sum_{j\leqslant i'} \gamma_j} > \tilde{c}_{i'+1}$ and $\tilde{c}_i \geqslant \tilde{c}_{i'}, \forall i \leqslant i'$ with $\tilde{c}_i \leqslant \tilde{c}_{i'}, \forall i \geqslant i'$ which proves that $i' = i^\star$. □

3.5 Numerical results

The data used to perform the experimental results has been constructed in the following way. We consider several instances of tail assignment problem with mid-haul flights as presented in Section 2.6.1. For a tail assignment problem, we generate several flight schedules by changing randomly the costs of the objective function and solving the linear program introduced in

Instance	# itineraries	Frontal LP		Column Generation	
		time (seconds)	memory (Mb)	time (seconds)	memory (Mb)
# 1	827	0.0315	5.73	0.0429	13.59
# 2	881	0.0329	5.85	0.0220	9.40
# 3	885	0.0372	5.87	0.0222	9.42
# 4	896	0.0341	5.89	0.0399	12.62
# 5	917	0.0313	5.95	0.0244	9.59
# 6	988	0.0377	6.11	0.0238	9.79
# 7	1001	0.0378	6.17	0.0326	10.73
# 8	1009	0.0427	6.18	0.0436	13.89
# 9	1032	0.0411	6.26	0.0231	9.52
# 10	1090	0.0389	6.41	0.0187	8.50
Average	952.6	0.0365	6.04	0.0293	10.71

Table 3.1 – Comparison between LP and CG (203 legs, 25 markets)

Instance	# itineraries	Frontal LP		Column Generation	
		time (seconds)	memory (Mb)	time (seconds)	memory (Mb)
# 1	31 674	1.1503	81.81	0.0590	14.18
# 2	32 751	1.2398	85.09	0.0590	14.35
# 3	32 804	1.3110	86.02	0.0629	14.41
# 4	32 882	1.3204	86.43	0.0625	14.39
# 5	33 167	1.3402	87.03	0.0644	14.44
# 6	33 423	1.2357	87.70	0.0704	14.48
# 7	34 076	1.3045	89.27	0.0628	14.60
# 8	34 314	1.3797	89.91	0.0645	14.63
# 9	34 433	1.3220	90.10	0.0638	14.63
# 10	34 630	1.3156	90.57	0.0653	14.72
Average	33 415.4	1.2919	87.39	0.0634	14.48

Table 3.2 – Comparison between LP and CG (1136 legs, 25 markets)

Chapter 2. That way, we generate a group of different networks with comparable sizes that can be used to benchmark our approach to solve Problem (3.2).

The numerical experiments have been performed on a computer with 7.7 Gb RAM, 8 cores at 1.9 GHz. The algorithms have been developed in `Julia` [19] with the modeling library `JuMP` [44]. All linear programs have been solved using `Gurobi 9.1` [64].

Tables 3.1, 3.2 and 3.3 are organized in the same way. The only difference lies in the instances used. Each line in the tables corresponds to a different instance with the characteristics described in the tables. The last line summarizes the results on the 10 examples by averaging the values obtained. Each table compares the frontal linear program and the column generation method on two aspects: the time spent in order to find an optimal solution (given in seconds), and the memory used (given in Mb.)

The different numerical experiments show that the column generation provides a very efficient

Instance	# itineraries	Frontal LP		Column Generation	
		time (seconds)	memory (Mb)	time (seconds)	memory (Mb)
# 1	78882	3.6220	207.73	0.1695	42.67
# 2	79212	5.0649	208.41	0.1613	42.74
# 3	79432	3.9543	208.94	0.1650	42.80
# 4	79587	4.0463	209.33	0.1703	42.78
# 5	79665	4.6699	209.42	0.1669	42.80
# 6	79719	3.7697	209.57	0.1668	42.82
# 7	79991	3.9992	210.15	0.1634	42.83
# 8	81016	5.2432	212.75	0.1717	43.02
# 9	81565	4.1194	213.84	0.1758	43.10
# 10	82391	5.1484	215.94	0.1825	43.24
Average	80146	4.3637	210.61	0.1693	42.88

Table 3.3 – Comparison between LP and CG (2075 legs, 36 markets)

way to solve the SBLP. Indeed, while the frontal LP and the column generation have fairly similar performance on small instances (as shown in Table 3.1), as the size of the problem increases, the relative difference in performance between the two method increases. In the case presented in Tables 3.2 and 3.3, we can see that the time and memory used by the frontal LP increases much more than the column generation. In the case of the biggest database tested, the column generation in more than 20 times more efficient.

3.6 Convergence of the atomic model

3.6.1 Proof of Theorem 3

We provide here the main steps of the proof of Theorem 3. A few intermediary results are necessary. Their proof is postponed to Section 3.6.2.

A key element is the following optimization problem, which we call the *fluid problem*. An intuitive interpretation of this problem, yet non-necessary for the proof, is possible and provided in Section 3.7.

$$V^{\text{fluid}} = \max \sum_{m\in M} \sum_{i\in m} c_i \, q_i^{\text{fluid}}(y) \tag{3.10a}$$

$$\text{st:} \sum_{i\ni l} y_i \leqslant s_l \qquad \forall l \in L \tag{3.10b}$$

$$y_i \geqslant 0 \qquad \forall i \in I, \tag{3.10c}$$

where the $q_i^{\text{fluid}}(y)$ are defined as follows.

We proceed on each market independently. Consider a market m. Let $g_i = \frac{\gamma_i}{\gamma_m^0 + \sum_{k\in m}\gamma_k}$ for each $i \in m$. Sort the indices so that $\frac{y_1}{g_1} \leqslant \frac{y_2}{g_2} \leqslant \cdots \leqslant \frac{y_{|m|}}{g_{|m|}}$. (This ordering is assumed to hold only within the definition of the $q_i^{\text{fluid}}(y)$.) Set

$$\bar{i} = \arg\min\left\{ \frac{D_m - \sum_{j<i} y_j}{1 - \sum_{j<i} g_j} \leqslant \frac{y_i}{g_i} \middle| i \in m \right\} \qquad \text{and} \qquad \tau = \frac{D_m - \sum_{j<\bar{i}} y_j}{1 - \sum_{j<\bar{i}} g_j}.$$

In case the set is empty, we define $\bar{i} = |m| + 1$ and keep the same definition for τ.

Then, we define

$$q_i^{\text{fluid}}(y) = \begin{cases} y_i & \text{if} \quad i \leqslant \bar{i}, \\ g_i \tau & \text{else.} \end{cases} \tag{3.11}$$

It turns out that the fluid problem and the SBLP are closely related.

Proposition 9. *The fluid problem* (3.10) *and the SBLP* (3.2) *have the same optimal value.*

On the other hand, the objective functions of the atomic problem scaled by θ and the fluid problem are related. An immediate consequence of this proposition is that for a feasible solution y of the fluid problem (3.10), the objective function of the scaled atomic problem converges in probability to the objective function of the fluid problem.

Proposition 10. *Let $\epsilon > 0$ and y such that $\sum_{i \ni l} y_i \leqslant s_l$ for all $l \in L$ and $y_i \geqslant 0$ for all $i \in I$. Then we have*

$$\lim_{\theta \to \infty} \mathbb{E}\left(\frac{Q_{i,\theta T,\theta y}}{\theta} \right) = q_i^{\text{fluid}}(y) \quad \textit{for all } i \in I.$$

With all these elements, the proof of Theorem 3 is straightforward.

Proof of Theorem 3. Let y be a feasible solution of the fluid problem (3.10). We have

$$\frac{1}{\theta} V_\theta^{\text{atom}} \geqslant \mathbb{E}\left[\frac{1}{\theta} \sum_{i \in I} c_i Q_{i,\theta T,\theta y} \right]$$

By Proposition 10, the objective function of the scaled atomic problem converges in distribution to the objective function of the fluid problem, and thus

$$\liminf_{\theta \to +\infty} \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant \sum_{i \in I} c_i q_i^{\text{fluid}}(y).$$

This implies that $\liminf \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant V^{\text{fluid}}$. With the help of Proposition 9, we get $\liminf \frac{1}{\theta} V_\theta^{\text{atom}} \geqslant V^{\text{SBLP}}$. □

Remark 10. Denote by f the function $(\theta, y) \mapsto \mathbb{E}\left[\frac{1}{\theta} \sum_{i \in I} c_i Q_{i,\theta T,\theta y}\right]$. Under the hypothesis that the function $f(\theta, \cdot)$ is continuous for every θ, and that the convergence (in Proposition 10) of the function $f(\theta, \cdot)$ is uniform then we even have a stronger result than Theorem 3. More precisely, we have $\lim_{\theta \to \infty} \frac{1}{\theta} V_\theta^{\text{atom}} = V^{SBLP}$.

Indeed, the following arguments provide the stated result. We denote by y_θ an element of $\arg\max_y f(\theta, y)$. By compacity, successively extracting a subsequence of θ and a subsequence of y, we take a subsequence $y_{\theta_k} \to y^\star$ such that $\lim_{k \to \infty} f(\theta_k, y_{\theta_k}) = \limsup \frac{1}{\theta} V_\theta^{\text{atom}}$. We then have that $V^{\mathrm{SBLP}}(y^\star) = \lim_{\theta\to\infty} f(\theta, y^\star)$ by definition. We have that $|f(\theta_k, y_{\theta_k}) - f(\theta_k, y^\star)| \leqslant |f(\theta_k, y_{\theta_k}) - V^{\mathrm{SBLP}}(y^\star)| + |V^{\mathrm{SBLP}}(y^\star) - f(\theta_k, y^\star)|$. Thus, by continuity and uniform convergence we get $\lim_{k\to\infty} f(\theta_k, y_{\theta_k}) = \lim_{k\to\infty} f(\theta_k, y^\star)$, and we get that $\limsup_{\theta\to\infty} f(\theta, y_\theta) \leqslant V^{\mathrm{SBLP}}(y^\star) \leqslant V^{SBLP}$, which in turn gives $\lim_{\theta\to\infty} \frac{1}{\theta} V_\theta^{\mathrm{atom}} = V^{SBLP}$.

3.6.2 Proofs of the intermediary steps

3.6.2.1 Proof of Proposition 9

In order to prove the equivalence between Problem (3.10) and Problem (3.2) we use yet another intermediate problem, with a formulation motivated by the following property.

Proposition 11. *For a given market m, and a given vector $(y_i)_{i\in m}$, the vector $q^{\mathrm{fluid}}(y)$ is a feasible solution of the following linear complementarity set of constraints.*

$$\begin{cases} \dfrac{q_i}{\gamma_i} \leqslant \dfrac{q_m^0}{\gamma_m^0} \perp q_i \leqslant y_i & \forall i \in m \\ q_m^0 + \sum\limits_{i\in m} q_i = D_m & \\ q_i \geqslant 0,\ q_m^0 \geqslant 0 & \forall i \in m. \end{cases} \tag{3.12}$$

Proof. Let m be a given market and $(y_i)_{i\in m}$ be a given capacity vector. We assume that the itineraries are sorted by increasing values of $\frac{y_i}{g_i}$. We set q^{fluid} as defined in (3.11).

By definition of q^{fluid} there exists $\bar{i}$ such that:

$$q_i^{\mathrm{fluid}} = \begin{cases} y_i & \text{if} \quad i \leqslant \bar{i} \\ g_i\tau & \text{else.} \end{cases} \tag{3.13}$$

We have in particular for a given $i \geqslant \bar{i}$

$$q_0^{\mathrm{fluid}} = g_0\tau = \frac{g_0}{g_i} q_i^{\mathrm{fluid}} = \frac{\gamma_m^0}{\gamma_i} q_i^{\mathrm{fluid}},$$

and by definition of τ, we also have $q_i^{\mathrm{fluid}} \leqslant y_i$.

For $i < \bar{i}$, we have

$$\begin{aligned} q_i^{\mathrm{fluid}} = y_i &< g_i \frac{D_m - \sum_{j<i} y_j}{1 - \sum_{j<i} g_j} \\ &\leqslant g_i \frac{D_m - \sum_{j<\bar{i}} y_j}{1 - \sum_{j<\bar{i}} g_j} \quad \frac{1 - \sum_{j<\bar{i}} g_j}{1 - \sum_{j<i} g_j} \quad \frac{D_m - \sum_{j<i} y_j}{D_m - \sum_{j<\bar{i}} y_j} \\ &\leqslant g_i \frac{D_m - \sum_{j<\bar{i}} y_j}{1 - \sum_{j<\bar{i}} g_j} \\ &\leqslant \frac{\gamma_i}{\gamma_m^0} q_0^{\mathrm{fluid}}. \end{aligned}$$

Thus, the complementarity constraint is satisfied.

Moreover, we have

$$
\begin{aligned}
q_0^{\text{fluid}} + \sum_{i\in m} q_i^{\text{fluid}} &= \sum_{i<\bar{i}} y_i + \sum_{i\geqslant \bar{i}} g_i \tau \\
&= \sum_{i<\bar{i}} y_i + \left(\sum_{i\geqslant \bar{i}} g_i\right) \frac{D_m - \sum_{i<\bar{i}} y_i}{1-\sum_{i<\bar{i}} g_i} \\
&= \sum_{i<\bar{i}} y_i + \left(1-\sum_{i<\bar{i}} g_i\right) \frac{D_m - \sum_{i<\bar{i}} y_i}{1-\sum_{i<\bar{i}} g_i} \\
&= D_m
\end{aligned}
$$

Therefore the demand constraint is also satisfied which ends the proof. □

Proposition 11 motivates the introduction of the following problem:

$$
\begin{aligned}
V^{\perp} = \max \sum_i c_i q_i & & & \text{(3.14a)}\\
\text{st:} \sum_{i\ni l} y_i \leqslant s_l & & \forall l & \text{(3.14b)}\\
\frac{q_i}{\gamma_i} \leqslant \frac{q_m^0}{\gamma_m^0} \perp q_i \leqslant y_i & & \forall i \in m,\ \forall m \in M & \text{(3.14c)}\\
q_m^0 + \sum_{i\in m} q_i = D_m & & \forall m \in M & \text{(3.14d)}\\
q_i \geqslant 0, \quad y_i \geqslant 0 & & \forall i & \text{(3.14e)}\\
q_m^0 \geqslant 0 & & \forall m \in M. & \text{(3.14f)}
\end{aligned}
$$

Problem (3.14) is a linear program with complementarity constraints. Proposition 11 shows the optimal solution of Problem (3.10) is a feasible solution of Problem (3.14). We show that the two problems are in fact equivalent in the following sense:

Proposition 12. *Problem* (3.10) *and Problem* (3.14) *have the same optimal value.*

Proof. By Proposition 11, any feasible solution of Problem (3.10) provides a feasible solution of Problem (3.14) with the same objective value. Therefore, we know that $V^{\text{fill}} \leqslant V^{\perp}$.

We now prove that an optimal solution $(\hat{q}, \hat{y})$ of Problem (3.14) provides a feasible solution of Problem (3.10) with the same value. We focus here on a single market m. Let suppose that the itineraries are indexed by increasing value of $\frac{y_i}{\gamma_i}$

The following implication holds: $\hat{q}_i < \hat{y}_i \Rightarrow \hat{q}_{i+1} < \hat{y}_{i+1}$. Indeed, suppose for a contradiction that $\hat{q}_i < \hat{y}_i$ and $\hat{q}_{i+1} = \hat{y}_{i+1}$. The complementary constraint for index i implies that $\frac{\hat{q}_i}{\gamma_i} = \frac{\hat{q}_m^0}{\gamma_m^0}$. Thus we have:

$$
\frac{\hat{q}_{i+1}}{\gamma_{i+1}} = \frac{\hat{y}_{i+1}}{\gamma_{i+1}} \geqslant \frac{\hat{y}_i}{\gamma_i} > \frac{\hat{q}_i}{\gamma_i} = \frac{\hat{q}_m^0}{\gamma_m^0},
$$

which is impossible since $(\hat{q}, \hat{y})$ satisfies (3.14c).

This implication makes that there exists an index $\hat{i}$ in the solution such that $\hat{q}_i = \hat{y}_i$ for all $i \leqslant \hat{i}$ and $\hat{q}_i < \hat{y}_i$ and $\hat{q}_i = \frac{\gamma_i}{\gamma_m^0}\hat{q}_m^0$ for all other indices. By Proposition 11, $\left(q^{\text{fluid}}(\hat{y}), \hat{y}\right)$ is a feasible solution of Problem (3.14). Therefore, proving the following claim enables to conclude

Claim: For a given vector $\hat{y}$, there exists at most one solution $\hat{q}$ to Problem (3.14).

We now prove the claim, in order to finish the proof. Let f be the index function defined as follows: $f(k) = \sum_{i=1}^{k} \hat{y}_i + \sum_{i=k+1}^{|m|} \hat{y}_k \frac{\gamma_i}{\gamma_k} + \hat{y}_k \frac{\gamma_m^0}{\gamma_k}$. Since the indices are sorted by increasing values of $\frac{y_i}{\gamma_i}$, f is increasing: For a given k, $\hat{y}_{k+1} \geqslant \hat{y}_k \frac{\gamma_{k+1}}{\gamma_k}$ which gives $f(k) \leqslant \sum_{i=1}^{k+1} \hat{y}_i + \sum_{i=k+2}^{|m|} \hat{y}_k \frac{\gamma_i}{\gamma_k} + \hat{y}_k \frac{\gamma_m^0}{\gamma_k}$ and for all $i > k+1$, $\hat{y}_{k+1} \frac{\gamma_i}{\gamma_{k+1}} \geqslant \hat{y}_k \frac{\gamma_i}{\gamma_k}$ which gives $f(k) \leqslant f(k+1)$.

For any index $i^\star$ solution to the problem, we know that any solution $q^\star$ satisfies $q_m^{0\star} > q_{i^\star}^\star \frac{\gamma_m^0}{\gamma_{i^\star}}$ and $q_i^\star = q_m^{0\star} \frac{\gamma_i}{\gamma_m^0} > y_{i^\star} \frac{\gamma_i}{\gamma_{i^\star}}$ for all $i > i^\star$. Thus:

$$D_m = q_m^{0\star} + \sum_{i \in m} q_i^\star \geqslant f(i^\star).$$

Moreover, for all $i > i^\star$, we have $q_m^{0\star} \frac{\gamma_i}{\gamma_m^0} = q_i^\star$ which implies $q_m^{0\star} < y_{i^\star+1} \frac{\gamma_m^0}{\gamma_{i^\star+1}}$ and thus $q_i^\star < y_{i^\star+1} \frac{\gamma_m^0}{\gamma_{i^\star+1}}$.Therefore $D_m = q_m^{0\star} + \sum_{i \in m} q_i^\star < f(i^\star + 1)$. Thus, there can exist only one index $i^\star$ solution for a given $\hat{y}$ which proves the claim. □

Proposition 13. *Problem* (3.14) *and the SBLP* (3.2) *have the same optimal value.*

Proof. Denote by $\bar{q}_i$, for all $i \in I$ and $\bar{q}_m^0$, for all $m \in M$ an optimal solution of the SBLP. Setting $q_i = \bar{q}_i$, $q_m^0 = \bar{q}_m^0$ and $y_i = \bar{q}_i$ provides a feasible solution of Problem (3.14), with the same objective value as the SBLP. We have $V^{SBLP} \leqslant V^\perp$.

To show the reverse, we only need to show that there always exists an optimal solution with the constraints $q_i = y_i$ being satisfied. Let $\hat{y}_i$, $\hat{q}_i$ and $\hat{q}_m^0$ be an optimal solution of Problem (3.14). If there exists $i_0 \in I$ such that $\hat{y}_{i_0} \geqslant \hat{q}_{i_0}$, decreasing the value $\hat{y}_{i_0}$ to $\hat{q}_{i_0}$ does not change the objective value and is still feasible. Indeed, for all legs l such that $l \in i_0$, we have that $\hat{q}_{i_0} + \sum_{i \ni l, i \neq i_0} \hat{y}_i \leqslant \sum_{i \ni l} \hat{y}_i \leqslant s_l$. Hence, setting the variables $y_i = q_i$ provides an optimal solution of Problem (3.14). In that case, the solution is feasible for the SBLP, which gives $V^\perp \leqslant V^{SBLP}$. □

Proof of Proposition 9. The proof can be naturally deduced from the successive use of Proposition 12 and Proposition 13. □

3.6.2.2 Proof of Proposition 10

In the present subsection, we suppose that the capacities y are fixed. In that context, the problem can be split by markets and therefore, we focus here on a single market m. We are interested by the number of customers $Q_{i,T,y}$ in each itinerary i after T time-steps. Since on a given market, the total demand is D_m, and that at each time step one customer arrives, we have in that case $T = D_m$.

For simplicity, we also consider that the itinerary “not buying” is included in the set of itineraries I.

Proposition 10 is used in the proof of Theorem 3. In this proposition, we want to compare the quantities $Q_{i,T,y}$ and $q^{\text{fluid}}(y)$. More precisely, $Q_{i,\theta T,\theta y}$ represents the value of the random variable Q where the capacities and the number of customers arriving have been multiplied by θ, and we prove that this random variables $\frac{Q_{i,\theta T,\theta y}}{\theta}$ converges in probability toward the value $q^{\text{fluid}}(y)$.

We introduce an alternative description of the process. We consider $(W(n))_{n\in\mathbb{N}}$ the random walk in discrete time in $\mathbb{Z}^I$ that starts at 0 with probability $\mathbb{P}(W(k+1) = W(k) + e_i) = \frac{\gamma_i}{\sum_{j\in I}\gamma_j}$ where e_i is the i^{th} canonical vector of $\mathbb{R}^I$. The random walk W represents the arrival process in the case where there are no capacities that limit the choice of customers.

We denote by $\wedge$ the component-wise minimum operator. We consider $(Z(n))_{n\in\mathbb{N}}$ the random walk with limit capacities y defined as follows:

$$Z(n,y) = (Z_1(n),\ldots,Z_I(n)) = W(n)\wedge y.$$

Let $\overline{T}_\theta = \min\big(n\in\mathbb{N} \mid \sum_{i\in I} Z_i(n,\theta y) = \lfloor\theta T\rfloor\big)$. It is a stopping time. It can be interpreted as the first moment the number of non-rejected customers in the process W customers is θT. With the definition of $\overline{T}_\theta$ and Z, we have the following lemma, which is intuitive. We give here the general idea of the proof and since the precise proof consists mainly in calculations, we present the details in Appendix C.

Lemma 14. *The random variables $Z(\overline{T}_\theta,\theta y)$ and $Q(\theta T,\theta y)$ have the same distribution.*

Sketch of the proof. The general idea for this proof is to show that $Z(\overline{T}_\theta(t),\theta y)_{t\in\mathbb{Z}}$ is a Markov chain with the same law as $Q(\theta t,\theta y)_{t\in\mathbb{Z}}$, where $\overline{T}_\theta(t) = \min\big(n\in\mathbb{N} \mid \sum_{i\in I} Z_i(n,\theta y) = \lfloor\theta t\rfloor\big)$. This is actually a quite natural result since $Z(\overline{T}_\theta(t),\theta y)_{t\in\mathbb{Z}}$ behave in the same way as $Q(\theta t,\theta y)_{t\in\mathbb{Z}}$, but on different times, which are captured by $\overline{T}_\theta(t)$. The proof mainly consists in the explicit calculations of the transition matrices and is detailed in Appendix C. The result taken for $t = T$ then gives the proof of Lemma 14. □

This lemma will be useful to prove Proposition 10. The process Z can be interpreted as follows: customers arrive one after another following the general probability law $g_i = \frac{\gamma_i}{\sum_{j\in I}\gamma_j}$, for all $i\in I$, and we only count a customer in when the capacities allow it.

We set $w(t) = \big(gt\big)_{t\geqslant 0}$ and $z(t,y) = \big((gt)\wedge y)\big)_{t\geqslant 0}$ the two deterministic versions of W and Z.

Lemma 15. *We have $z(T,y) = q^{\text{fluid}}(y)$.*

Proof. For a given y, let $t' = \arg\min_t\{\sum_{i\in I} z(t,y) = T\}$ and $i' = \arg\min_i\Big\{t' \leqslant \frac{y_i}{g_i}\Big\}$. The function $\sum_{i\in I} z(t)$ is strictly increasing.

For $i < i'$, since $\frac{y_i}{g_i} < t'$, then $\sum_{i\in I} z(\frac{y_i}{g_i}, y) = \sum_{j<i} y_i + \sum_{j\geqslant i} g_j\frac{y_i}{g_i} < T$ which gives $\frac{T-\sum_{j<i} y_j}{1-\sum_{j<i} g_j} > \frac{y_i}{g_i}$.

For the index i', we have $\sum_{i\in I} z(\frac{y_{i'}}{g_{i'}}, y) \geqslant T$ which gives $\frac{T-\sum_{j<i'} y_j}{1-\sum_{j<i'} g_j} \leqslant \frac{y_{i'}}{g_{i'}}$.

Therefore $i' = \overline{i}$. Then $\sum_{i\in I} z(t', y) = T$, which gives $t' = \tau$ and proves the lemma. □

Proof of Proposition 10. As W is a random walk, the renormalized random walk $\left(\frac{W(\lfloor\theta t\rfloor)}{\theta}\right)_{t\geqslant 0}$ converges in distribution to the deterministic process $\left(g_1 t,\ldots,g_{|m|} t\right)_{t\geqslant 0}$. Thus, the process $\left(\frac{Z(\lfloor\theta t\rfloor,\theta y)}{\theta}\right)_{t\geqslant 0}$ converges in distribution to the vector $z(t,y)$. The convergence in distribution of $\left(\frac{Z(\lfloor\theta t\rfloor,\theta y)}{\theta}\right)_{t\geqslant 0}$ toward $z(t,y)$ implies the convergence in distribution for every t and in particular for $t=T$. A classical application of the portmanteau lemma is that convergence in distribution to a deterministic quantity implies convergence in probability. By Lemma 15, we get thus the following convergence in probability $\frac{Z(\lfloor\theta T\rfloor,\theta y)}{\theta}\xrightarrow{\mathbb{P}} q^{\text{fluid}}$. In particular we get

$$\frac{1}{\theta}\sum_{i\in I} Z_i\left(\lfloor\theta T\rfloor,\theta y\right)\xrightarrow{\mathbb{P}}\sum_{i\in I} q_i^{\text{fluid}}(y)=T.$$

We also have $\frac{1}{\theta}\sum_{i\in I} Z_i(\overline{T}_\theta,\theta y)=\frac{1}{\theta}\lfloor\theta T\rfloor\to T$.

Since $\lfloor\theta T\rfloor\leqslant\sum_{i\in I}\min\left(W_i(\overline{T}_\theta),y_i\right)\leqslant\sum_{i\in I} W_i(\overline{T}_\theta)=\overline{T}_\theta$ and since Z is increasing, we have

$$0\leqslant Z_i(\overline{T}_\theta,\theta y)-Z_i(\lfloor\theta T\rfloor,\theta y)\leqslant\sum_{j\in I}\left(Z_j(\overline{T}_\theta,\theta y)-Z_j(\lfloor\theta T\rfloor,\theta y)\right).$$

With the help of two convergence facts established just above, we get that $\frac{1}{\theta}\left(Z_i(\overline{T}_\theta,\theta y)-Z_i(\lfloor\theta T\rfloor,\theta y)\right)$ converges in probability to 0, and in particular $\frac{1}{\theta}\left(Z_i(\overline{T}_\theta,\theta y\right)$ converges in probability to $q_i^{\text{fluid}}(y)$. Lemma 14 show then that $\frac{1}{\theta}Q_i(\theta T,\theta y)$ converges in probability to $q_i^{\text{fluid}}(y)$ (again, because this latter is deterministic).

Let g be the function such that $g(x)=\max(x,D)$ which is continuous and upper bounded. Since $\frac{1}{\theta}Q_i(\theta T,\theta y)$ converges in probability to $q_i^{\text{fluid}}(y)$, the portmanteau lemma gives that $\lim_{\theta\to\infty}\mathbb{E}\left(\frac{1}{\theta}g(Q_i(\theta T,\theta y))\right)=\mathbb{E}g\left(\left(q_i^{\text{fluid}}(y)\right)\right)$, and therefore that $\lim_{\theta\to\infty}\mathbb{E}\left(\frac{1}{\theta}Q_i(\theta T,\theta y)\right)=\mathbb{E}\left(q_i^{\text{fluid}}(y)\right)$ which ends the proof. □

3.7 Complementary results on the fluid model

3.7.1 Modeling relevance of the fluid model

We are interested here in the fluid values q^{fluid} defined in Section 3.6.1. We now provide a natural interpretation for those values, based on a differential equation. Without loss of generality (for a fixed value of the capacity vector, the problem becomes separable per market), we assume that there is only one market.

In this fluid approximation, instead of having customers arriving one after the other, we have a continuous flow arriving of “unitary intensity.” This flow splits among the open itineraries with respect to the discrete choice model probabilities and we expect the flow $q_i(t)$ of customers in the itinerary i at each time $t\in\mathbb{R}_+$ to be continuous and differentiable almost everywhere, and to satisfy the following differential equation:

$$\begin{cases}\frac{\mathrm{d}}{\mathrm{d}t}q_i(t)=\dfrac{\gamma_i}{\sum_{j\in I(t)}\gamma_j}\mathbb{1}(i\in I(t)) & \forall i\in I,\ a.e.\\ q_i(0)=0 & \forall i\in I\end{cases}\tag{3.15}$$

where $I(t) = \{j \in m \mid q_j(t) < y_j\}$ and with the convention that $0/0 = 0$.

(The extra constraint $\sum_{i\in I} \frac{\mathrm{d}}{\mathrm{d}t} q_i(t) \stackrel{\text{a.e}}{=} 1$ would be expected, as long as the total flow of customers has not exceeded $\sum_{i\in I} y_i$. It is actually not necessary. This can be checked with arguments similar to the proof of Lemma 17.)

The following result establishes the link with the expression of Section 3.6.1. Denoting by D the total demand, the values $q_i(t)$ for $t = D$ are interpreted as the amount of customers on each itinerary, at the end of the process.

Proposition 16. *For a given capacity vector y, Problem* (3.15) *has a unique solution and its value at time D is given by $q^{\text{fluid}}(y)$.*

The remainder of the section focuses on the proof of this proposition.

Lemma 17. *Problem* (3.15) *has a unique solution q. This solution satisfies $\sum_{i\in I} q_i(t) = t$ for all $t \leqslant \sum_{i\in I} y_i$. Moreover, each q_i is a non-decreasing piecewise affine function and there exists a t such that $q_i(t) = y_i$.*

Proof. A solution to Problem (3.15) exists. Indeed, we can build a solution by the following recursive process. Define t^* as the smallest t for which there is an $i \in I$ with $\frac{\gamma_i}{\sum_{j\in I}\gamma_j} t = y_i$. (Note that $\gamma_j > 0$ for all j.) In case there is a tie, make an arbitrary choice. We define the functions $q_i(t) = \frac{\gamma_i}{\sum_{j\in I}\gamma_j} t$ for $t \in [0, t^*]$. Finding the solution of Problem (3.15) comes down to finding the solution of the same problem with $|I| - 1$ remaining itineraries and initial values of $q_i(t^*)$. The same process can be applied to the new problem. This recursive process ends since the number of itineraries decreases at each step. Remark that at each step, we show that there is a new itinerary i for which there exists a t such that $q_i(t) = y_i$.

We now show uniqueness of the solution. We claim that any solution q_i is necessarily continuous piecewise affine and non-decreasing. Indeed, consider a solution q_i. It is continuous and has a positive derivative almost everywhere, which implies that q_i is non-decreasing. Thus $I(t)$ is non-increasing along time. Let C be the set of values of t for which $I(t)$ changes. By finiteness of $I(t)$, the set C is also finite. Between two consecutive elements in C, the value of $\frac{\gamma_i}{\sum_{j\in I(t)}\gamma_j} \mathbb{1}(i \in I(t))$ does not change and hence the value taken by $\frac{\mathrm{d}}{\mathrm{d}t} q_i$ is the same almost everywhere between two consecutive elements in C. Integrating this expression shows that q_i is piecewise affine, with changes of slope occurring at elements in C. We finish the proof of the uniqueness by noting that, reduced to that kind of functions, uniqueness of the solution of Problem (3.15) is immediate: an induction shows that the slopes of any two solutions must be equal and that the changes in $I(t)$ occur necessarily at the same values of t.

Similar arguments applied to $\sum_{i\in I} q_i$ shows that $\sum_{i\in I} q_i(t) = t$ for all $t \leqslant \sum_{i\in I} y_i$, where we use the fact that $\sum_{i\in I} \frac{\mathrm{d}}{\mathrm{d}t} q_i(t) = 1$ (and thus $I(t) \neq \varnothing$) for almost every $t \leqslant \sum_{i\in I} y_i$ (obtained by summing over I the equation in Problem (3.15)). □

From now on, we consider a solution q of Problem (3.15). Define t_i as the smallest t such that $q_i(t) = y_i$. Such t exists because of Lemma 17 and t_i is well-defined because q_i is continuous. These quantities are easy to interpret: t_i corresponds to the first time the itinerary i gets full.

Remark that, by definition, $I(t)$ changes only when t is one of the t_i's. We also set $t_0 = 0$, $y_0 = 0$ and $g_0 = 1$.

Let $g_i = \frac{\gamma_i}{\sum_{j\in I}\gamma_j}$. (This notation has already been used in Section 3.6.1.) Without loss of generality, we assume that the itineraries are ordered so that the quantities $\frac{y_i}{g_i}$ are in a non-decreasing order.

Lemma 18. *We have* $t_1 \leqslant t_2 \leqslant \cdots \leqslant t_{|I|}$.

Proof. Assume for a contradiction that $t_i > t_{i+1}$. For almost every $t \in [0, t_{i+1})$, we have $\frac{\mathrm{d}}{\mathrm{d}t} q_i(t) = \frac{\mathrm{d}}{\mathrm{d}t} q_{i+1}(t)\frac{g_i}{g_{i+1}}$. The functions q_i and q_{i+1} are continuous and, according to Lemma 17, piecewise affine. Since $q_i(0) = q_{i+1}(0) = 0$, integrating shows that

$$\frac{q_i(t)}{g_i} = \frac{q_{i+1}(t)}{g_{i+1}} \quad \forall t \leqslant t_{i+1}. \tag{3.16}$$

Now, we have the following relations:

$$\frac{y_{i+1}}{g_{i+1}} = \frac{q_{i+1}(t_{i+1})}{g_{i+1}} = \frac{q_i(t_{i+1})}{g_i} < \frac{y_i}{g_i},$$

where the first holds by definition of t_{i+1}, the second equality is equation (3.16) for $t = t_{i+1}$, and the last inequality holds because $t_{i+1} < t_i$ by assumption and t_i is by definition the smallest t for which $q_i(t) = y_i$. This contradicts the ordering on the itineraries. $\square$

Lemma 19. *We have for all i*

$$t_i - t_{i-1} = \left(\frac{y_i}{g_i} - \frac{y_{i-1}}{g_{i-1}}\right)\left(1 - \sum_{k=1}^{i-1} g_k\right).$$

Proof. The proof works by induction on i. The base case is immediate since for $t \in [0, t_1)$, the set $I(t)$ equals I, which gives $g_1 t_1 = y_1$, and we have $t_1 - t_0 = \frac{y_1}{g_1}$, as required.

Suppose now that the expression to be proved holds for all $i' \leqslant i$. According to Lemma 18, the slope of the piecewise affine function q_{i+1} changes at $t_1, t_2, \ldots, t_i$ in this order. On the interval $(t_{i'}, t_{i'+1})$, it is equal to $\frac{\gamma_{i+1}}{\sum_{j\geqslant i'+1}\gamma_j} = \frac{g_{i+1}}{1-\sum_{j\leqslant i'} g_j}$. Thus,

$$y_{i+1} = \sum_{i'=1}^{i+1} (t_{i'} - t_{i'-1})\frac{g_{i+1}}{1 - \sum_{k=1}^{i'-1} g_k}.$$

An easy manipulation leads then to

$$t_{i+1} - t_i = \left(\frac{y_{i+1}}{g_{i+1}} - \sum_{i'=1}^{i} \frac{d_{i'}}{1 - \sum_{k=1}^{i'-1} g_k}\right)\left(1 - \sum_{k=1}^{i} g_k\right).$$

Using the induction hypothesis, we can substitute $d_{i'}$ and get

$$t_{i+1} - t_i = \left(\frac{y_{i+1}}{g_{i+1}} - \sum_{i'=1}^{i}\left(\frac{y_{i'}}{g_{i'}} - \frac{y_{i'-1}}{g_{i'-1}}\right)\right)\left(1 - \sum_{k=1}^{i} g_k\right),$$

which can be rewritten as

$$t_{i+1} - t_i = \left(\frac{y_{i+1}}{g_{i+1}} - \frac{y_i}{g_i}\right)\left(1 - \sum_{k=1}^{i} g_k\right),$$

as desired. □

Proof of Proposition 16. Uniqueness is ensured by Lemma 17.

Let us prove the second part of the statement. If $D = 0$, the proof is immediate. We assume thus $D > 0$. Denote by d_i the quantity $t_i - t_{i-1}$ for all i. Consider $i^\star$ defined as one plus the biggest index i such that $t_i \leqslant D$. (Note that it is possibly equal to $|I|+1$.) For $i < i^\star$, by definition of the t_i and Lemmas 17 and 18, we have $q_i(D) = y_i$. Using explicit expressions for the successive slopes of the piecewise affine function q_i, as in the proof of Lemma 19, we get for $i \geqslant i^\star$

$$q_i(D) = \sum_{j<i^\star} d_j \frac{g_i}{1-\sum_{k<j} g_k} + \left(D - \sum_{j\leqslant i^\star} d_j\right)\frac{g_i}{1-\sum_{k<i^\star} g_k}.$$

Defining $\Theta = \sum_{j<i^\star} d_j \frac{1}{1-\sum_{k<j} g_k} + \left(D - \sum_{j\leqslant i^\star} y_j\right)\frac{1}{1-\sum_{k<i^\star} g_k}$, this expression takes the form $q_i(D) = g_i\Theta$. According to Lemma 17, $\sum_{i\in I} q_i(D) = D$. Combining this with the expressions of $q_i(D)$ we have just obtained for $i < i^\star$ and $i \geqslant i^\star$, we get the explicit formula

$$\Theta = \frac{D - \sum_{i<i^\star} y_i}{1 - \sum_{i<i^\star} g_i}.$$

This shows that, to finish the proof, we only need to verify that $i^\star = \bar{i}$, where $\bar{i}$ has been defined in Section 3.6.1: we will then have $\Theta = \tau$. This is exactly what we are going to do.

Using the definition of Θ and Lemma 19, a simple calculation gives that $\Theta = \frac{y_i^{\star-1}}{g_{i^\star-1}} + \frac{D-t_{i^\star-1}}{1-\sum_{j<i^\star} g_j}$. Since $D - t_{i^\star-1} \leqslant d_{i^\star}$, using Lemma 19 again, we get that $\Theta \leqslant \frac{y_i^\star}{g_{i^\star}}$. Therefore $i^\star \geqslant \bar{i}$.

For every $i \leqslant i^\star$, we have that $q_i(t_i) = y_i$, and for $i' \geqslant i$, we have $q_{i'}(t_i) = g_{i'}\left(\sum_{j<i} \frac{d_j}{1-\sum_{j<i} g_j}\right)$. Moreover, $\sum_{i'\geqslant i} q_i(t_i) = (t_i - \sum_{j<i} y_j)$. Therefore $\frac{y_i}{g_i} \leqslant \frac{t_i - \sum_{j<i} y_j}{1-\sum_{j<i} g_j} \leqslant \frac{D-\sum_{j<i} y_j}{1-\sum_{j<i} g_j}$. Then we have $i^\star \leqslant \bar{i}$, which ends the proof. □

3.7.2 Practical experiments

According to Proposition 10, the quantity $\frac{Q_{i,\theta T,\theta y}}{\theta}$ converges in probability towards the quantities $q_i^{\text{fluid}}(y)$ for a given capacity vector y. In the present subsection, based on numerical experiments, we discuss the accuracy of approximating the quantity $\mathbb{E}[Q_{i,T,y}]$, which is difficult to compute and in which the airline is interested, by $q_i^{\text{fluid}}(y)$.

In general, there is no reason for the two quantities to be equal. There are two particular cases however where the atomic and the fluid models give the same solution. First, when there is only one itinerary available (whose interest is very limited). Second, if the minimum capacity over all the itineraries is bigger than the total demand (which corresponds to the case where there is no capacities at all). Indeed, in that case, $Q_{i,T,y}$ becomes a classical multinomial distribution, and $q_i^{\text{fluid}} = \tau g$, which is exactly the expectancy of a miltinomial distribution.

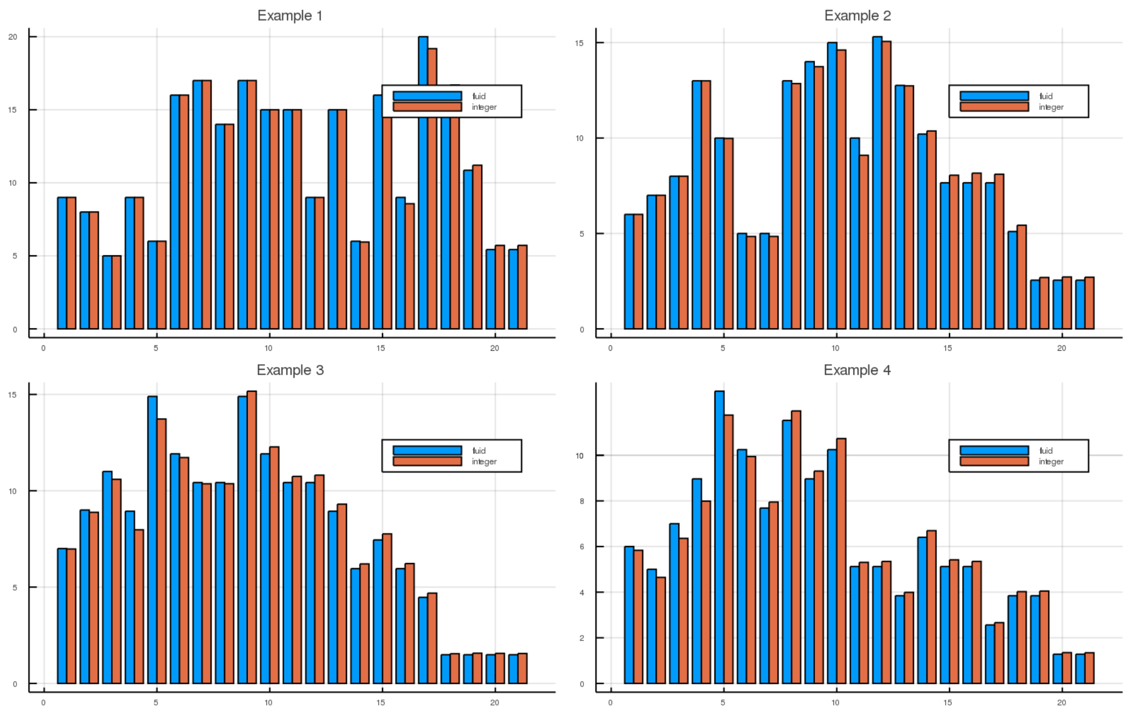

Figure 3.2 – Comparison of $q^{\text{fluid}}(y)$ and $\mathbb{E}[Q_{T,y}]$ for different values of T and y.

We have a closed formula for the quantities $q_i^{\text{fluid}}(y)$ which makes it easy to compute. However the quantities $\mathbb{E}[Q_{i,T,y}]$ are not as easy to compute. In order to get a proper comparison, we therefore perform Monte-Carlo simulations of the arrival process over a large number of scenarios. Some results are given in Figure 3.2. There is only one market with 21 itineraries. We have ordered the itineraries by increasing values of $\frac{y_i}{g_i}$. The heights of the histograms are equal respectively to $\mathbb{E}[Q_{i,T,y}]$ (in orange) and $q_i^{\text{fluid}}(y)$ (in blue). It appears quite clearly that the fluid approximation overestimates the filling of an itinerary before some threshold itinerary, and underestimates after this itinerary. This has been the case in all experiments we have performed and we propose therefore the following conjecture.

Conjecture 20. *When the itineraries are sorted following the quantities $\frac{y_i}{g_i}$, there exists an itinerary l such that for $i \leqslant l$, we have $q_i^{\text{fluid}}(y) \geqslant \mathbb{E}[Q_{i,T,y}]$, and for $i > l$, we have $q_i^{\text{fluid}}(y) \leqslant \mathbb{E}[Q_{i,T,y}]$.*

Note that the inequality $q_i^{\text{fluid}}(y) \geqslant \mathbb{E}[Q_{i,T,y}]$ holds for all $i < \bar{i}$, where $\bar{i}$ has been defined in Section 3.6.1. Indeed, for such i, we have $q_i^{\text{fluid}}(y) = y_i$. Moreover, in the atomic model, even if an itinerary i is very attractive, the probability that free space remains at the end of the filling process is in general non-zero. This implies that $\mathbb{E}[Q_{i,T,y}]$ is strictly lower than the capacity y_i. This strict overestimation suggests that customers transfer on less attractive itineraries and might lead to underestimate these latter itineraries. These elements could serve as an intuitive motivation for the conjecture.

3.7.3 Concentration inequalities

This section gathers work that has been led in collaboration with Danielle Tibi.

3.7.3.1 Main result

Proposition 10 provides a convergence in probability between the quantities $\frac{Q_{i,\theta T,\theta y}}{\theta}$ and $q_i^{\text{fluid}}(y)$ for a given capacity vector y. In this subsection, we are interested in proving a more quantitative estimation of the difference between those two values. In particular we show that the convergence speed is exponential.

Theorem 21. *As $\theta \to \infty$, there exist two constants K_1 and K_2 such that*

$$\left\| \frac{\mathbb{E}[Q_{\theta T,\theta y}]}{\theta} - q^{\text{fluid}} \right\| \leqslant K_1 e^{-K_2 \theta}.$$

In order to prove this proposition, the general idea is to use concentration inequalities on the two itinerary case, and then extend the result to the general case. We therefore prove the following result:

Proposition 22. *In the case where there are only two itineraries, with probability of choice g_1 (respectively g_2), and capacity y_1 (respectively y_2) for the first (respectively second) itinerary, we have as θ goes to infinity:*

- *If $g_1 D < y_1$ and $g_2 D < y_2$*

$$D\left(g_1 - \exp\left(-2\theta \frac{(y_1 - g_1 D)^2}{D}\right)\right) \leqslant \frac{\mathbb{E}(Q_{1,\theta T,\theta y})}{\theta} \leqslant D\left(g_1 + \exp\left(-2\theta \frac{(y_2 - g_2 D)^2}{D}\right)\right). \tag{3.17}$$

- *If $g_1 D < Y_1$ and $g_2 D \geqslant y_2$*

$$D - y_2 \leqslant \frac{\mathbb{E}(Q_{1,\theta T,\theta y})}{\theta} \leqslant D - y_2\left(1 - \exp\left(-2\theta \frac{(y_2 - g_2 D)^2}{D}\right)\right). \tag{3.18}$$

- *If $g_1 D \geqslant Y_1$ and $g_2 D < y_2$*

$$y_1\left(1 - \exp\left(-2\theta \frac{(y_1 - g_1 D)^2}{D}\right)\right) \leqslant \frac{\mathbb{E}(Q_{1,\theta T,\theta y})}{\theta} \leqslant y_1. \tag{3.19}$$

This result that focuses on two itineraries can be applied to more than two itineraries by regrouping the different itineraries in the general case. That way, Proposition 22 gives:

Proposition 23. *For all itineraries i such that $g_i D \geqslant y_i$, we have*

$$y_i\left(1 - \exp\left(-2\theta \frac{(g_i D - y_i)^2}{D}\right)\right) \leqslant \frac{\mathbb{E}(Q_{1,\theta T,\theta y})}{\theta} \leqslant y_i. \tag{3.20}$$

However, the proof of Proposition 23 does not provide a good bound for all $i \in I$. A corollary that can be deduced from the proof however is the following:

Corollary 24. *In the case where all itineraries verify $g_i D < y_i$, we have*

$$D\left(g_i - \exp\left(-2\theta \frac{(y_i - g_i D)^2}{D}\right)\right) \leqslant \frac{\mathbb{E}(Q_{i,\theta T,\theta y})}{\theta} \leqslant D\left(g_i + \sum_{j \neq i} \exp\left(-2\theta \frac{(y_i - g_j D)^2}{D}\right)\right). \tag{3.21}$$

Combining those propositions (that will be proved in Section 3.7.3.2), we now prove Theorem 21.

Proof of Theorem 21. To prove that we have exponential bounds in the general case, we show that we can use the bounds of type Corollary 24 on all itineraries $i \in I$ such that the expectation on the itinerary is less than θy_i, and the bounds of type (3.25) on the other itineraries. Intuitively, when the itineraries are sorted following the values $\left(\frac{y_i}{g_i}\right)_{i\in I}$, the index $\bar{i}$ provides a separation between those two types of itineraries.

For all $i < \bar{i}$, we consider the alternative process $Q' = \left(Q'_i, \ldots, Q''_{|I|}\right)$ which is a process with a demand $\theta D' = \theta\left(D - \sum_{j<i} y_j\right)$, with $|I| - i + 1$ possible itineraries and for each itinerary j a probability $g'_j = \frac{g_j}{\sum_{k\geqslant i} g_k}$ and a capacity y_j. The condition $i < \bar{i}$ gives $g'_i D' \geqslant y_i$. Therefore Proposition 23 gives the following inequality:

$$y_i\left(1 - \exp\left(-2\theta\frac{(g'_i D' - y_i)^2}{D'}\right)\right) \leqslant \frac{\mathbb{E}(Q_{i,\theta T,\theta y})}{\theta} \leqslant y_i. \tag{3.22}$$

For all $i \geqslant \bar{i}$, we introduce the following process $Q'' = \left(Q''_{\bar{i}}, \ldots, Q''_{|I|}\right)$. For each $i \geqslant \bar{i}$, Q''_i has a capacity θy_i and a probability $g''_i = \frac{g_i}{\sum_{k\geqslant\bar{i}} g_k}$. We consider the arrival of $\theta\left(D - \sum_{k<\bar{i}} y_k\right)$ customers.

We claim that $\mathbb{E}(Q''_i) \leqslant \mathbb{E}(Q_{i,\theta T,\theta y})$, for all $i \geqslant \bar{i}$. To prove the claim, we consider again the alternative process Z introduced in Section 3.6.2.2 and Lemma 14.

Let $T''_{\theta(D-\sum_{k<\bar{i}} y_k)}$ be the stopping time $\min\left(t \mid \sum_{k\geqslant i} Z_k(t,\theta y) = \theta(D - \sum_{k<\bar{i}} y_k)\right)$. This stopping time corresponds to the moment where $\theta(D - \sum_{k<\bar{i}} y_k)$ customers have been accepted on the itineraries $\bar{i}, \ldots, I$. We have that $\overline{T}_\theta \geqslant T''_{\theta(D-\sum_{k<\bar{i}} y_k)}$ since in order to get θD customers on the whole process, we need at least $\theta\left(D - \sum_{k<\bar{i}} y_k\right)$ customers on the itineraries $\bar{i}, \ldots, I$.

Therefore, we get that $\left(Z_{\bar{i}}(\overline{T}_\theta, \theta y), \ldots, Z_I(\overline{T}_\theta, \theta y)\right) \geqslant \left(Z_{\bar{i}}(T''_{\theta(D-\sum_{k<\bar{i}} y_k)}), \ldots, Z_I(T''_{\theta(D-\sum_{k<\bar{i}} y_k)})\right)$. This last vector has the same distribution as $Q'' = \left(Q''_{\bar{i}}, \ldots, Q''_I\right)$. Thus, $\mathbb{E}(Q''_i) \leqslant \mathbb{E}(Q_i)$, for all $i \geqslant \bar{i}$.

We have that $g''_i D'' = g_i \frac{(D - \sum_{k<\bar{i}} y_k)}{1 - \sum_{j<\bar{i}} g_j} \leqslant y_i$. Therefore we can apply Corollary 24 to the process Q'', and we have the following bound:

$$\left(D - \sum_{k<\bar{i}} y_k\right)\left(\frac{g_i}{1 - \sum_{k<\bar{i}} g_k} - \exp\left(-2\theta\frac{\left(y_i - g_i\frac{D - \sum_{k<\bar{i}} y_k}{1 - \sum_{k<\bar{i}} g_k}\right)^2}{D - \sum_{k<\bar{i}} y_k}\right)\right) \leqslant \mathbb{E}(Q''_i) \leqslant \mathbb{E}(Q_i).$$

In order to have an upper bound, we know that $\mathbb{E}(Q_i) = D - \sum_{k\neq i} \mathbb{E}(Q_k)$, and having exponential lower bounds for all itineraries in I, we can deduce an exponential upper bound. □

3.7.3.2 Proofs of intermediary results

Proof of Proposition 22

Proof. We focus here on the particular case where only two itineraries are available for sale. For the sake of simplicity, we will denote in this subsection $Q_1 = Q_{1,\theta T,\theta y}$ and $Q_2 = Q_{2,\theta T,\theta y}$. In that case, we know have that $Q_1 \leqslant \theta y_1$, $Q_2 \leqslant \theta y_2$ and $Q_1 + Q_2 = \theta D$. We can suppose that $D \leqslant y_1 + y_2$ since without this hypothesis, we know that the two itineraries are full.

Let B_1 be a random variable with law $\mathcal{B}(\theta D, g_1)$ where $g_1 = \frac{\gamma_1}{\gamma_1+\gamma_2}$, and $B_2 = \theta D - B_1$. The variable Q_1 can be expressed in function of B_1 as follows:

$$Q_1 = \max\big(\min\big(B_1, \theta y_1\big), \theta D - \theta y_2\big). \tag{3.23}$$

Indeed, when we consider a realization of B_1, either

$$\begin{cases} B_1 \leqslant \theta y_1 & \text{and} \quad B_2 = \theta D - B_1 \leqslant \theta y_2 \\ B_1 \leqslant \theta y_1 & \text{and} \quad B_2 \geqslant \theta y_2 \\ B_1 \geqslant \theta y_1 & \text{and} \quad B_2 \leqslant \theta y_2. \end{cases}$$

In all those cases, Equation (3.23) is true. This equation provides the following bounds on Q_1:

$$\min\big(B_1, \theta y_1\big) \leqslant Q_1 \leqslant \theta D - \big(B_2, \theta y_2\big). \tag{3.24}$$

The previous inequalities motivate the study of the random variable $\min\big(B_1, \theta y_1\big)$ to get bounds on $\mathbb{E}(Q_1)$. Since $\mathbb{E}\big(\min\big(B_1, \theta y_1\big)\big) = \mathbb{E}\big(B_1 \mathbb{1}_{B_1 \leqslant \theta y_1}\big) + \theta y_1 \mathbb{P}\big(B_1 > \theta y_1\big)$, we can apply the Hoeffding inequality to get exponential bounds on the difference between the random variable and its expectancy. The Hoeffding inequality and a remainder on the principle of concentration inequalities can be found in Appendix A.2. A more general introduction to concentration inequalities is given by [143]. In the present case, the Hoeffding inequality gives:

If $g_1 D < y_1$:

$$\begin{aligned} \mathbb{E}\big(\min\big(B_1, \theta y_1\big)\big) &\geqslant \mathbb{E}\big(B_1 \mathbb{1}_{B_1 \leqslant \theta y_1}\big) \\ &= \theta g_1 D - \mathbb{E}(B_1 \mathbb{1}_{B_1 > \theta y_1}) \\ &\geqslant \theta g_1 D - \mathbb{P}(B_1 > \theta y_1) \\ &\geqslant \theta D \left(g_1 - \exp\left(-2\theta \frac{(y_1 - g_1 D)^2}{D} \right) \right). \end{aligned}$$

If $g_1 D \geqslant Y_1$:

$$\begin{aligned} \mathbb{E}(\min(B_1, Y_1)) &\geqslant \theta y_1 \mathbb{P}(B_1 > \theta y_1) \\ &= \theta y_1 \big(1 - \mathbb{P}(B_1 \leqslant \theta y_1)\big) \\ &\geqslant \theta y_1 \left(1 - \exp\left(-2\theta \frac{(y_1 - g_1 D)^2}{D} \right) \right). \end{aligned}$$

We have the relations (3.24), and we know that $D - y_2 \leqslant \frac{Q_1}{\theta} \leqslant y_1$ is always true. With the previous bounds on $\mathbb{E}\big(\min\big(B_1, \theta y_1\big)\big)$, we get:

- If $g_1 N < Y_1$ and $g_2 N < Y_2$

$$\max\left(D - y_2, D\left(g_1 - \exp\left(-2\theta\frac{(y_1 - g_1 D)^2}{D}\right)\right)\right) \leqslant \frac{\mathbb{E}(Q_1)}{\theta}$$
$$\leqslant \min\left(y_1, D\left(g_1 + \exp\left(-2\theta\frac{(y_2 - g_2 D)^2}{D}\right)\right)\right).$$

- If $g_1 N < Y_1$ and $g_2 N \geqslant Y_2$

$$\max\left(D - y_2, D\left(g_1 - \exp\left(-2\theta\frac{(y_1 - g_1 D)^2}{D}\right)\right)\right) \leqslant \frac{\mathbb{E}(Q_1)}{\theta}$$
$$\leqslant \min\left(y_1, D - y_2\left(1 - \exp\left(-2\theta\frac{(y_2 - g_2 D)^2}{D}\right)\right)\right).$$

- If $g_1 N \geqslant Y_1$ and $g_2 N < Y_2$

$$\max\left(D - y_2, y_1\left(1 - \exp\left(-2\theta\frac{(y_1 - g_1 D)^2}{D}\right)\right)\right) \leqslant \frac{\mathbb{E}(Q_1)}{\theta}$$
$$\leqslant \min\left(y_1, D\left(g_1 + \exp\left(-2\theta\frac{(y_2 - g_2 D)^2}{D}\right)\right)\right).$$

Asymptotically, the previous relations give the expected result which end the proof of Proposition 22. □

Proof of Proposition 23

Proof. For $i \in I$, let $Q' = (Q'_1, Q'_2)$ be the process with two itineraries with respective capacities θy_i and $\sum_{k\neq i} \theta y_k$ and respective probability choices g_i and $\sum_{k\neq i} g_k$.

Let a scenario be a series of customer arrivals among the I itineraries. For a given scenario, we consider that the two processes Q and Q' fill up in parallel as follows. If the customer chooses itinerary i, he adds up in Q_i and in Q'_1. If the customer chooses an itinerary $k \neq i$, he adds up in Q_k and in Q'_2. We denote by T (respectively T') the moment θD customer have arrived in Q (respectively Q'). During this filling process it may happen that a customer is accepted in Q'_2 and not in $(Q_k)_{k\neq i}$, since the constraint capacity $\sum_{k\neq i} y_k$ is less strong than the constraint capacities $(y_k)_{k\neq i}$. This implies that $T' \leqslant T$, and in particular, we have that $Q_{i,\theta T,y} \geqslant Q'_1$.

This inequality coupled with Proposition 22 means that we can apply asymptotically the bounds found in the case with only 2 itineraries to Q as follows:

- If $g_i D \geqslant y_i$

$$y_i\left(1 - \exp\left(-2\theta\frac{(g_i D - y_i)^2}{D}\right)\right) \leqslant \frac{\mathbb{E}(Q_{i,\theta T,\theta y})}{\theta} \leqslant y_i. \tag{3.25}$$

and

- If $g_i D < y_i$

$$\max\left(D - \sum_{k\neq i} y_k,\ D\left(g_i - \exp\left(-2\theta \frac{(g_i D - y_i)^2}{D}\right)\right)\right) \leqslant \frac{\mathbb{E}(Q_{i,\theta T,\theta y})}{\theta} \leqslant y_i. \tag{3.26}$$

The second inequality is not very relevant since there exists a gap between the two bounds. In order to prove Theorem 21, some more work need to be done. The first inequality suffices however to prove Proposition 23. □

Proof of Corollary 24

Proof. This corollary can be deduced from the equations (3.26) in the proof of Proposition 23. Indeed, when $g_i D < y_i$ for all $i \in I$, we get that the maximum in the expression is always realized for the right hand-side term. Coupled with the equality $\sum_{i\in I} \mathbb{E}(Q_{i,\theta T,\theta y}) = \theta D$, we get the Corollary. □

3.8 Bibliographical remarks

3.8.1 Discrete choice models

Discrete Choice Models.

Many discrete choice models have been developed in the last decades, and while random utility models (and in particular the multinomial logit) remain the most widely used, an important variety of models exists. Since we presented and developed in this chapter a model based on the multinomial logit, we briefly present the recent evolutions in the discrete choice model literature. Most of those models generalize the random utility model and the multinomial logit model and try to fix the issues associated with this model. Among those models, we can find the representative agent model (see Hofbauer and Sandholm [69] for a theoretical presentation), the semi-parametric model (proposed by Natarajan et al. [100]), the Markov chain model (introduced by Zhang and Cooper [157] and studied by Blanchet et al. [20]), the generalized attraction model (Gallego et al. [56]) and even non-parametric models. Recently, Feng et al. [49] prove that semi-parametric and representative agent models are in fact equivalent.

Assortment problem.

The atomic model we presented is a dynamic assortment problem that used a multinomial logit choice model. The recent literature has studied the assortment problem with different choice models. Such models could provide an interesting extension to the model we use. For instance, Jagabathula [73] proposes heuristics to find assortments with a general attraction model. Rusmevichientong et al. [111] work on a mixture of multinomial logit attraction model and prove the NP-hardness of the assortment problem in that case. They also propose a polynomial time approximation scheme algorithm. Désir and Goyal [45] develop an approximation algorithm for the assortment problem under a capacity constraint in that case. Recently, Désir et al. [46] further study the assortment problem with capacity constraint under the Markov-chain choice model. Cao et al. [29] propose a linear program formulation of the assortment problem in the case of a mixture of multinomial logit.

Parameter estimation.

A crucial challenge for the assortment problem or in the revenue management problem is the estimation of the parameters of the models used. Before the introduction of customer behaviour in the revenue management, van Ryzin and McGill [140] provide ways to estimate the protection levels of the seats that are sold. In the more recent literature, the estimations focus on the discrete choice model parameters. Vulcano et al. [145] and Vulcano et al. [146] present EM algorithms to get practical estimations of the parameters used by the attraction models. Gallego et al. [56] successfully apply this EM method in the case of the general attraction model they introduced. van Ryzin and Vulcano [141] recently focus on an EM algorithm with a rank-based customer choice.

Revenue Management.

Since the work of Talluri [128], few work as been dedicated to the improvement of the leg-based revenue management. Topaloglu et al. [136] propose a policy that takes capacity control and overbooking into account. Recently, Feldman and Topaloglu [48] focus on the inclusion of the Markov-chain model in this problem.

A lot of recent work has been dedicated to the network revenue management which has been formulated by Gallego et al. [55]. Two main lines of work can be distinguished here. A first one develops different approximations of the network revenue management problem. Adelman [3] and Zhang and Adelman [156] focus on an affine approximation of the problem and propose different ways to solve it through column generation. Meissner and Strauss [96] take the same approach but with a piecewise linear approximation. Topaloglu [134] use a Lagrangian decomposition to compute bid-prices on the network revenue management. Kunnumkal and Topaloglu [80] use a decomposition method per flight legs, with the capacities used for the decomposition being generated by an auxiliary problem. Recently, Kunnumkal and Talluri [79] propose another approximation and propose a Lagrangian decomposition method to provide solutions.

Another line of research has been on the resolution of the choice-based deterministic linear problem (introduced by Gallego et al. [55] and studied by Liu and van Ryzin [89]). This problem has been a popular approximation and different solving techniques and relaxations have been introduced. Bront et al. [23] prove that in the case of multinomial logit choice model, they can reformulate the problem and solve the problem with a column generation approach where the sub problem is a hyperbolic problem. Meissner et al. [97] introduce a relaxation of the problem and prove that it can be solved efficiently, in order to generate bounds.

4 Benders decomposition for the aircraft routing with itinerary-based revenue

4.1 Introduction

Having set up the foundations with the definition of the schedules that can be operated in Chapter 2 and detailed the revenue evaluation of those schedules in Chapter 3, we can now focus on the optimization of such schedules with an itinerary-based revenue. In the present chapter we are interested in joining those two problems. As discussed in the Introduction, this might lead to more efficient schedule designs. More precisely, we want to solve the "extended" aircraft routing problem, the one with frequency and gate constraints introduced in Chapter 2, while estimating the revenue at an itinerary-based level of precision. This problem reveals to be challenging since it gathers the difficulties of each individual problem. In particular the number of variables that need to be considered is very large, and to our knowledge, no algorithm can solve such a problem on real size networks.

In the present chapter, we propose a mixed integer linear program (MILP) that models the extended aircraft routing problem, with the revenue approximated by the SBLP discussed in Chapter 3. This MILP has two types of variables. The first type of variables is integer and models the network and solves more particularly the aircraft routing problem. The second type is continuous, models the revenue estimation, and represents the estimated number of customers on the different itineraries. The number of continuous variables is much higher than the number of integer variables.

A direct resolution of the MILP is not tractable with present day solvers. Our goal is to investigate an alternate yet natural approach via Benders decomposition. Such a decomposition is expressed with a master problem similar to the extended aircraft routing problem studied in Chapter 2 and with a slave problem almost identical to the SBLP of Chapter 3, the difference being the addition of artificial equality constraints dictated by the Benders decomposition. The first option we consider is the direct resolution of the slave problem with a solver. The second option builds upon the column generation of Section 3.4, which is extremely efficient for solving the original version of the SBLP. The challenge for the second option is thus to keep the efficiency of the original resolution while dealing with the new artificial constraints. We succeed in addressing this challenge by proposing a method relying on duality results, which makes the second option much quicker and efficient than the vanilla decomposition.

The chapter is organized as follows:

- Section 4.2 provides a literature overview of the methods that integrate advanced demand models in the operations-for-airline problems, and a general presentation on the Benders decomposition method.
- Section 4.3 formally introduces the joint problem.
- Section 4.4 introduces the general methodology of the Benders decomposition applied to the joint problem.
- Section 4.5 focuses on the Benders decomposition where the slave problem is solved enhanced with the column generation method.
- Section 4.6 gathers the results of the numerical experiments that have been carried out.

4.2 Literature review

4.2.1 Airline revenue models for schedule optimization problems

The development of advanced revenue estimation models optimized jointly with schedule design problems has been a central research topic in the last decade. Those developments answer two main types of challenge. The first challenge comes from the airline operations, which needs to be more precise in the models that are used. Advanced customer choice models have been developed in order to take into account phenomena like spill and recover, and market effects like cannibalization. We refer to Chapter 3 and Appendix B for a precise literature review on this topic. In this literature review, we focus on the second challenge which is a combinatorial challenge. The objective in this case is to include a revenue model based on a customer choice model which can be integrated in an optimization problem. The general aim is to generate a model that captures the complexity of the revenue generated which remains simple enough to be tractable.

As stated by Barnhart and Cohn [9], an important axis of research has been the integration of a network demand model in the operational optimization problems. This integration to the fleet assignment problem has been an important center of interest. Barnhart et al. [10] present a general framework in order to integrate a network demand to the fleet assignment problem. In this framework the objective function has two parts: a cost for the fleet assignment choice, and a revenue that depends on the fleet assignment choice and the revenue management policy chosen. When the revenue model is chosen to be linear with each leg, we recover the classical fleet assignment model. Many papers try to include a network revenue model in the fleet assignment model. Most of the research is focused on including itinerary based revenue models, which are close from the real-life way revenue is generated for a company.

One of the first operational problem that has been studied with an advanced revenue model is the fleet assignment. Barnhart et al. [11] propose a first model called passenger mix model that does not evaluate the demand only as an aggregate data, but also integrate spill and recover. Spill and recover evaluates how many customers decide to buy an alternative itinerary when their preferred choice is not available. This model has been reused in subsequent works.

Some alternatives have also been investigated in the literature. Jacobs et al. [71] include in the fleet assignment model an upper bound on the revenue that is inspired from the interpretation of the revenue management bid price. Cadarso et al. [28] incorporate a nested multinomial logit in the integrated frequency planning, fleet assignment and timetabling. Their model includes variables that capture spill and recover, and incorporate probabilities of recapture rates that are estimated with a nested multinomial logit. In order to include these dependencies that are non-linear, they use a piece-wise linear model to approximate the cost.

Abdelghany et al. [2] solve a multi-objective non-linear program that maximizes the itinerary-based revenue generated by customers and the resources (in terms of crew and aircraft rotation opportunities) of the company. To solve their model they take a heuristic approach. First they generate a possible planning with a genetic algorithm. Then they evaluate the revenue using a customer simulator which uses a discrete choice model inspired from Coldren [33] and generate randomly atomic clients until the demand is matched. Then they estimate the fitness of the model and iterate over this process until convergence.

Yan and Tseng [153] work on a model that uses two networks: one for the aircraft flow and one for the passenger flow. They take into account the tail assignment problem, with a passenger revenue model that limits the number of passengers in flight according to the capacity available. Their model is a MILP, which they solve with a Lagrangian relaxation to get a lower bound. They generate an upper bound with a heuristic. Yan et al. [151] further extend the model developed by Yan and Tseng [153], adding a logit customer choice model in their problem. They get a non-linear integer problem, and they propose to solve this problem iteratively, alternating the generation of the fleet flow and the generation of the passenger flow. The model is enhanced by Yan et al. [152], who add stochastic demand considerations. They formulate the problem as a bilevel optimization program. To solve it, they take a fixed number of scenarios, and solve the problem presented by Yan et al. [151] for each scenario. With the generated solutions, they propose two heuristics that generate a global demand reflecting the stochastic behavior of customers. Finally, they solve the problem with those fixed demand found.

Wei et al. [149] have recently presented a model very similar to the one we develop. Their model focuses on the fleet assignment problem. They include at the same time comprehensive time retiming and an itinerary based revenue estimation. This estimation is based on the model of Gallego et al. [56].

4.2.2 Decomposition methods

In the second part of this literature review, we focus more precisely on the Benders decomposition method. The method is an advanced technique to solve mathematical programs that have a particular structure with complicating variables. In the recent literature for airlines, this method has become a popular choice (see for instance [36, 98, 99, 102, 118, 119]). We focus here on the theoretical aspects of the Benders decomposition and the different improvements that have been carried out in the last decades.

The Benders decomposition has been introduced in the early 70s by Benders [16]. The aim of the method is to exploit the structure of the problem by separating the complicating variables and the other variables in a master problem and a slave problem respectively. The general idea is to perform a cut generation on the master problem in order to converge quickly toward the optimal solution. The cuts are generated at each iteration through the resolution of the slave problem. A comprehensive and recent review on the Benders decomposition method can be found in the work of Rahmaniani et al. [108]. We also present in more details the method and its links with other decomposition methods in Appendix A.

The Benders decomposition method has been enhanced with different technical improvements in the last decades. The method has been improved following two main directions: primal stabilization and dual enhancement methods.

Primal stabilization aims at finding a better sequence of solutions of the master problem at each iteration in order to accelerate convergence. Ben-Ameur and Neto [15] present this stabilization method in the general case of cutting-plane and column generation. They also experiment the method on a wide variety of problem. Zaourar and Malick [154] summarize the general stabilization variants that can be used in bundle methods and applied to Benders decomposition. They present a quadratic stabilization inspired by different types of bundle methods.

Dual enhancement focuses on improving the cuts generated with the slave problem when several dual solutions provide feasible cuts. Magnanti and Wong [92] are among the first to study cut generation enhancement in the context of Benders decomposition. They propose to strengthen the cuts generated by the Benders decomposition by searching among equivalent dual solutions of the slave problem in order to generate better cuts. This method has been further studied and enhanced by Papadakos [101] and Sherali and Lunday [122]. Papadakos [101] generalize the Magnanti-Wong problem and prove that Pareto optimal cuts can be generated with solutions different from the natural primal solutions generated during a Benders decomposition. Sherali and Lunday [122] prove that Magnanti-Wong cut can be generated by adding small perturbations in the constraints of the dual subproblem.

4.3 Problem statement

The problem we are interested in gathers the problems introduced in the two previous chapters. We briefly remind the notation used to model the problems that were introduced in Chapters 2 and 3.

We consider that the company has a set $\mathcal{K}$ of available fleets with n_k planes available for each fleet $k \in \mathcal{K}$. We denote by L the set of legs that can be operated by the company.

We suppose that we have a graph $G = (V, E)$ that encodes the feasible routes that can be operated by the planes of the company. This graph is a "graph of flights" in which each vertex represents a given time and airport, and the edges are associated with the different legs that can be operated by the company; see Chapter 2 for more thorough definition of this graph. Each leg $\ell \in L$ has one or more edges associated with it and is associated with a number of available seats s_ℓ.

We denote by $\left(L_f\right)_{f\in\mathcal{F}}$ the frequency partition of the legs and for simplicity, we consider that the for each $k \in \mathcal{F}$, the sets L_f can also refer at the set of corresponding edges. For each time step $t \in [T]$, and each edge $e \in E$, we denote by o_t^e the binary equals to 1 if the edge e is associated with a leg that uses a gate in the hub of the company at that time and 0 otherwise. For a fleet k, the set T_1^k gathers the edges that correspond to a move of a plane of fleet k between time steps 1 and 2.

The set of legs L generates naturally a set of itineraries I. Each itinerary is composed of one or several legs. We denote by γ_i the attractiveness of the itinerary i. All the itineraries are split into markets (that correspond to an origin-destination). The set of markets is denoted by M and for each market, we denote by D_m the total demand on this market and γ_m^0 the attractiveness of the option "not buying" for this itinerary. We denote by α_i the ratio $\frac{\gamma_i}{\gamma_m^0}$ where m is the market of i.

The joint problem can be formulated as follows:

$$
\begin{aligned}
\max \sum_{e\in E} c_e x_e + \sum_{m\in M}\sum_{i\in m} c_i q_i & & & \quad (4.1a)\\
\text{st: } \sum_{e\in\delta^+(v)} x_e = \sum_{e\in\delta^-(v)} x_e & & \forall v\in V & \quad (4.1b)\\
\sum_{e\in T_1^k} x_e = n_k, & & \forall k\in\mathcal{K} & \quad (4.1c)\\
\sum_{e\in E} o_t^e x_e \leqslant \gamma, & & \forall t\in[T] & \quad (4.1d)\\
\sum_{e\in L_f} x_e = 1, & & \forall f\in\mathcal{F} & \quad (4.1e)\\
\sum_{i\ni l} q_i \leqslant s_\ell\left(\sum_{e\in\ell} x_e\right) & & \forall \ell\in L & \quad (4.1f)\\
q_i \leqslant \alpha_i q_m^0 & & \forall i\in m,\ \forall m\in M & \quad (4.1g)\\
q_m^0 + \sum_{i\in m} q_i = D_m & & \forall m\in M & \quad (4.1h)\\
q_i \geqslant 0 & & \forall i\in I & \quad (4.1i)\\
q_m^0 \geqslant 0 & & \forall m\in M & \quad (4.1j)\\
x_e \in \left[x_e^{max}\right] & & \forall e\in E & \quad (4.1k)
\end{aligned}
$$

Problem (4.1) is composed of two types of variables: the variables x_e that are integer variables representing a circulation of planes on the graph G, and the variables q_i and q_m^0 representing the number of customers on each itinerary sold by the company. The variables x_e are restricted by Constrains (4.1b), (4.1c), (4.1d), and (4.1e), and which corresponds to Problem (2.3) which has been introduced in Chapter 2. The variables q_i are associated with Constraints (4.1f), (4.1g) and (4.1h) correspond to Problem (3.2) introduced in Chapter 3 for a given value of x. Constraint (4.1f) has however a particular status in the model. This constraint limits the number of customers on each plane to the limit of available seats, and it is the only constraint linking the two kinds of variables.

The cost function of Problem (4.1) is a linear sum over the set of legs and of itineraries. In practice, since the revenue is generated by customer buying itineraries, the values c_i for $i\in I$ are positive and capture this part of the revenue. The values c_e for $e\in E$ reflect however the costs generated by the flight management (for instance an edge representing a distant parking will generate a bigger cost than an edge representing the arrival at a gate slot). In the problem considered here, they have typically a negative value (since they are costs and we maximize the revenue). In that sense, the values c_e are very different from the values used in the problem presented in Chapter 2.

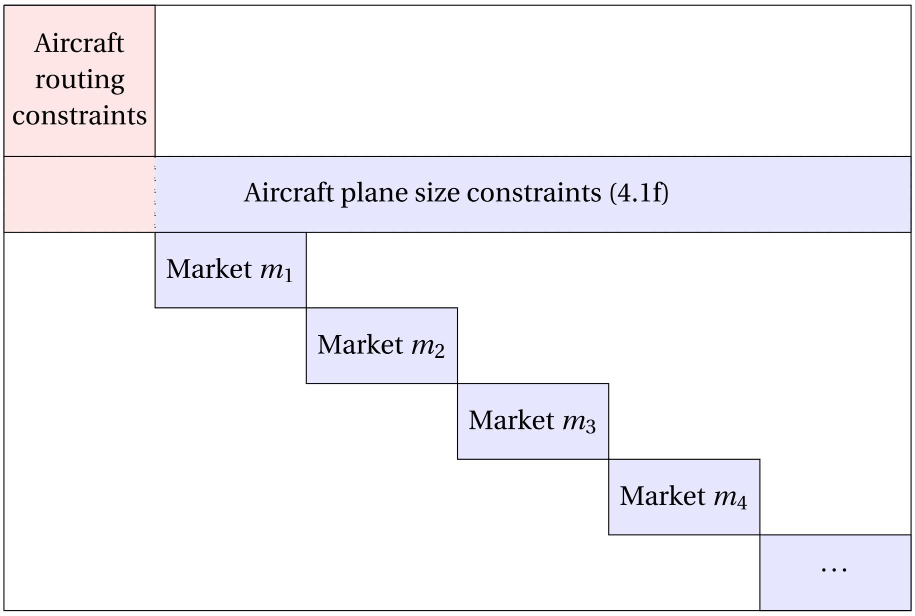

Figure 4.1 – Structure of the joint problem

4.4 Benders decomposition for the joint problem

Problem (4.1) has a particular structure with complicating variables. As it is illustrated in Figure 4.1 the number of continuous variables (the same order of magnitude than the number of itinerary considered) is much higher than the number of integer variables (the same order of magnitude than the number of legs considered). It is then natural to use a Benders decomposition method in order to exploit this structure.

4.4.1 The Benders reformulation

In the present case, we can reformulate Problem (4.1) as follows:

$$
\begin{aligned}
\max \sum_{e \in E} c_e x_e + \eta(x) & & & \quad (4.2a)\\
\text{st: } \sum_{e \in \delta^+(v)} x_e = \sum_{e \in \delta^-(v)} x_e & & \forall v \in V & \quad (4.2b)\\
\sum_{e \in T_1^k} x_e = n_k, & & \forall k \in \mathcal{K} & \quad (4.2c)\\
\sum_{e \in E} o_t^e x_e \leqslant \gamma, & & \forall t \in [T] & \quad (4.2d)\\
\sum_{e \in L_f} x_e = 1, & & \forall f \in \mathcal{F} & \quad (4.2e)\\
x_e \in \left[x_e^{max}\right] & & \forall e \in E. & \quad (4.2f)
\end{aligned}
$$

where

$$\eta(\overline{x}) = \max \sum_i c'_i q_i \tag{4.3a}$$
$$\text{st: } q_i \leqslant \alpha_i q^0_m \qquad \forall i \in m,\ \forall m \in M \tag{4.3b}$$
$$q^0_m + \sum_{i \in m} q_i = D_m \qquad \forall m \in M \tag{4.3c}$$
$$\sum_{i \ni \ell} q_i \leqslant s_\ell \sum_{e \in \ell} \overline{x}_e \qquad \forall \ell \in L \tag{4.3d}$$
$$x_e = \overline{x}_e \qquad \forall e \in E \tag{4.3e}$$
$$q_i \geqslant 0 \qquad \forall i \in I \tag{4.3f}$$
$$q^0_m \geqslant 0 \qquad \forall m \in M. \tag{4.3g}$$

As it is illustrated in Conejo et al. [34], the Benders reformulation of Problem (4.2) takes the following form:

$$\max \sum_{e \in E} c_e x_e + \eta \tag{4.4a}$$
$$\text{st: } \sum_{e \in \delta^+(v)} x_e = \sum_{e \in \delta^-(v)} x_e \qquad \forall v \in V \tag{4.4b}$$
$$\sum_{e \in T^k_1} x_e = n_k \qquad \forall k \in \mathcal{K} \tag{4.4c}$$
$$\sum_{e \in E} o^e_t x_e \leqslant \gamma \qquad \forall t \in [T] \tag{4.4d}$$
$$\sum_{e \in L_f} x_e = 1 \qquad \forall f \in \mathcal{F} \tag{4.4e}$$
$$\sum_{i \in I} \overline{q}^h_i c'_i + \sum_{e \in E} \overline{\lambda}^h_e (x_e - x^h_e) \geqslant \eta \qquad \forall h \in \mathcal{H} \tag{4.4f}$$
$$x_e \in \left[x^{max}_e\right] \qquad \forall e \in E, \tag{4.4g}$$

where $\mathcal{H}$ is the set of extreme points of the polyhedron of Problem (4.3). For $h \in \mathcal{H}$, the values $\overline{q}^h$ are the solution associated with the corresponding extreme point, and $\overline{\lambda}^h$ are the dual solution associated with Constraint (4.3e). Notice that this formulations of the Benders cut is based on the primal solutions of the salve problem. This is an alternative to the more classical formulation based on the dual solutions of the slave problem.

The size of $\mathcal{H}$ is exponentially large and solving this reformulation directly would be particularly inefficient. The Benders decomposition method relies on a constraint generation. In that context, the master problem at iteration 0 is similar to Problem (4.4) but without constraints (4.4f). At each iteration j, we denote by $\mathcal{C}(j)$ the set of elements in $\mathcal{H}$ that have already been added to the master. The aim of iteration j is to find which new constraint will be generated. In order to select this constraint, a solution of the master problem is generated and used to solve the slave problem. With the solution of the slave problem, a new constraint can be generated. The direct resolution of slave problem, mentioned in the introduction, consists in solving Problem (4.3) with a solver. In the next section, we propose an alternative method for solving Problem (4.3) and generating the Benders constraints more efficiently.

Remark 11. In general, for a given solution of the master problem, it might happen that there is no feasible solution of the primal slave problem (or equivalently an extreme ray for the dual slave problem). In that case, another cut called *feasibility cut* is generally added in order to discard the master solution. In our case, we did not consider this case because for every feasible solution of the master problem, there exists an optimal solution of the slave problem.

4.5 Solving the slave problem with column generation

Since Problem (4.3) has the same structure as the linear program introduced in Chapter 3, using the Dantzig–Wolfe reformulation introduced in the previous chapter seems to be a promising strategy. In that case, Problem (4.3) writes as

$$
\begin{aligned}
\max \sum_{m\in M}\sum_{g\in G_m} c'_g \mu_g & & \text{(4.5a)}\\
\text{st: } \sum_{g\in G_m} \mu_g = 1 & \quad \forall m \in M & \text{(4.5b)}\\
\sum_{m\in M}\sum_{g\in G_m} \beta_g^\ell \mu_g \leqslant s_\ell \sum_{e\in\ell} x_e & \quad \forall \ell \in L & \text{(4.5c)}\\
x_e = \overline{x}_e & \quad \forall e \in E & \text{(4.5d)}\\
\mu_g \geqslant 0 & \quad \forall g \in G_m, \forall m \in M. & \text{(4.5e)}
\end{aligned}
$$

In practice, solving this problem with a column generation approach reveals to be inefficient. Indeed, for many ℓ, we have $\sum_{e\in\ell} x_e = 0$, which makes the slave problem highly degenerated. In order to circumvent this problem, an approach can be to use the column generation on the problem restricted to the legs that are effectively used.

Set $L_0 = \left\{\ell \in L \mid \sum_{e\in\ell} \overline{x}_e = 0\right\}$ and set $E_0 = \left\{e \in E \mid \sum_{e'\in\ell \mid e\in\ell} \overline{x}_{e'} = 0\right\}$, and $L_1 = L \setminus L_0$ and $E_1 = E \setminus E_0$. Similarly, we denote by I_1 the set of itineraries whose legs are all in L_1 and $I_0 = I \setminus I_1$. This notation simply splits the sets between the legs, edges and itineraries that can be used given the solution of the master problem and the others.

Up to setting $q_i = 0$ for all $i \in I_0$, Problem (4.3) is equivalent to:

$$
\begin{aligned}
\max \sum_{i\in I_1} c'_i q_i & & \text{(4.6a)}\\
\text{st: } q_i \leqslant \alpha_i q^0_{m(i)} & \quad \forall i \in I_1 & \text{(4.6b)}\\
q^0_m + \sum_{i\in m\cap I_1} q_i = D_m, & \quad \forall m \in M & \text{(4.6c)}\\
\sum_{i\ni\ell, i\in I_1} q_i \leqslant s_\ell \sum_{e\in\ell} x_e & \quad \forall \ell \in L_1 & \text{(4.6d)}\\
x_e = \overline{x}_e & \quad \forall e \in E_1 & \text{(4.6e)}\\
q_i \geqslant 0 & \quad \forall i \in I_1 & \text{(4.6f)}\\
q^0_m \geqslant 0 & \quad \forall m \in M. & \text{(4.6g)}
\end{aligned}
$$

Similarly as Problem (4.3) becoming Problem (4.6), the Dantzig–Wolfe reformulation (4.5) becomes:

$$\max \sum_{m\in M}\sum_{g\in G_m} c'_g \mu_g \tag{4.7a}$$
$$\text{st:} \sum_{g\in G_m} \mu_g = 1 \qquad \forall m\in M \tag{4.7b}$$
$$\sum_{m\in M}\sum_{g\in G_m} \beta_g^\ell \mu_g \leqslant s_\ell \sum_{e\in\ell} x_e \qquad \forall \ell\in L_1 \tag{4.7c}$$
$$x_e = \overline{x}_e \qquad \forall e\in E_1 \tag{4.7d}$$
$$\mu_g \geqslant 0 \qquad \forall g\in G_m, \forall m\in M. \tag{4.7e}$$

Solving Problem (4.7) reveals to be much more efficient than solving Problem (4.5) since the problem is no longer degenerate. In practice the column generation will be an efficient method since at each iteration, the added column improves the objective value of the master problem.

The resolution of Problem (4.7) with a column generation provides a primal solution of the slave problem quickly. However, in order to generate a cut for the Benders decomposition, the dual values λ_e associated with Constraints (4.5d) need to be determined for all $e\in E$.

Solving Problem (4.7) provides optimal values only for the dual variables λ_e for $e\in E_1$. Therefore the values λ_e for $e\in E_0$ still need to be determined. Finding optimal values for those variables reveals to be easy. In our case, the optimal dual values $\tilde{\lambda}_e$ for $e\in E_0$ have an up-monotone structure (as it will become clear in the proof of Theorem 25). Therefore, finding an optimal solution of this dual is easy. However, as it has been observed by Magnanti and Wong [92], finding an optimal solution of the dual slave problem does not necessarily provide "good" Benders cuts. In order to generate cuts of better quality, an alternative problem needs to be solved.

For a given solution $\overline{x}$ of the master problem, let $\overline{\gamma}$ and $\overline{\delta}$ be the optimal dual values of Constraints (4.7b) and (4.7c) in Problem (4.7) (associated with $\overline{x}$). For any vector $(p_e)_{e\in E_0}$, we introduce the following problem:

$$\min \sum_{e\in E_0} p_e\lambda_e \tag{4.8a}$$
$$\text{st:} \sum_{\ell\in i\cap L_0} \delta_\ell + \phi_i \geqslant c_i - \frac{\overline{\gamma}_m}{D_m} - \sum_{\ell\in i\cap L_1} \overline{\delta}_\ell \qquad \forall i\in m, \forall m\in M \tag{4.8b}$$
$$\frac{\overline{\gamma}_m}{D_m} \geqslant \sum_{i\in m} \alpha_i\phi_i \qquad \forall m\in M \tag{4.8c}$$
$$\lambda_e = \sum_{\ell\ni e} \delta_\ell s_\ell \qquad \forall e\in E_0 \tag{4.8d}$$
$$\delta_\ell \geqslant 0 \qquad \forall \ell\in L_0 \tag{4.8e}$$
$$\phi_i \geqslant 0 \qquad \forall i\in I \tag{4.8f}$$
$$\lambda_e \in \mathbb{R} \qquad \forall e\in E_0. \tag{4.8g}$$

Theorem 25. *For a given master solution $\overline{x}$, denote by $\overline{\mu}$ the optimal solution of Problem* (4.7) *associated with $\overline{x}$, and by $\overline{\lambda}$ the optimal dual values associated with Constraint* (4.5d). *For any* *feasible solution* $(\tilde{\lambda}, \tilde{\delta}, \tilde{\phi})$ *of Problem* (4.8), *the cut*

$$\sum_{m\in M}\sum_{g\in G_m} c'_g \overline{\mu}_g + \sum_{e\in E_1} \overline{\lambda}_e (x_e - \overline{x}_e) + \sum_{e\in E_0} \tilde{\lambda}_e (x_e - \overline{x}_e) \geqslant \eta \tag{4.9}$$

is a feasible Benders cut. If the $(p_e)_{e\in E_0}$ *are positive and* $(\tilde{\lambda}, \tilde{\delta}, \tilde{\phi})$ *is an optimal solution of Problem* (4.8), *then this cut is not dominated by other cuts for other values of* λ_e *for* $e \in E_0$.

Proof. For a fixed value of $\overline{x}$, the dual of Problem (4.3) has the following form.

$$\min \sum_{e\in E} \lambda_e \overline{x}_e + \sum_{m\in M} \psi_m D_m \tag{4.10a}$$

$$\text{st: } \sum_{\ell\in i} \nu_\ell + \phi_i + \psi_m \geqslant c_i \qquad \forall i \in m, \forall m \in M \tag{4.10b}$$

$$\psi_m \geqslant \sum_{i\in m} \alpha_i \phi_i \qquad \forall m \in M \tag{4.10c}$$

$$\lambda_e = \sum_{\ell \ni e} \nu_\ell s_\ell \qquad \forall e \in E \tag{4.10d}$$

$$\nu_\ell \geqslant 0 \qquad \forall \ell \in L \tag{4.10e}$$

$$\phi_i \geqslant 0 \qquad \forall i \in I \tag{4.10f}$$

$$\psi_m \in \mathbb{R} \qquad \forall m \in M \tag{4.10g}$$

$$\lambda_e \in \mathbb{R} \qquad \forall e \in E. \tag{4.10h}$$

The dual of Problem (4.5) (which is the Dantzig–Wolfe reformulation of Problem (4.3)) has the following form.

$$\min \sum_{e\in E} \lambda_e \overline{x}_e + \sum_{m\in M} \gamma_m \tag{4.11a}$$

$$\text{st: } \gamma_m + \sum_{\ell\in L} \delta_\ell \beta_g^\ell \geqslant c'_g \qquad \forall g \in G_m, \forall m \in M \tag{4.11b}$$

$$\lambda_e = \sum_{l \ni e} \delta_\ell s_\ell \qquad \forall e \in E \tag{4.11c}$$

$$\delta_\ell \geqslant 0 \qquad \forall \ell \in L \tag{4.11d}$$

$$\lambda_e \geqslant 0 \qquad \forall e \in E \tag{4.11e}$$

$$\gamma_m \geqslant 0 \qquad \forall m \in M. \tag{4.11f}$$

When solving the slave problem of the Benders decomposition with Problem (4.7) instead of the column generation on the whole problem, the dual solution found is optimal for the following problem:

$$\min \sum_{m\in M} \gamma_m + \sum_{e\in E_1} \lambda_e \overline{x}_e \tag{4.12a}$$

$$\text{st:}\gamma_m + \sum_{\ell\in L} \delta_\ell \beta_g^\ell \geqslant c'_g \qquad \forall g \in G_m, \forall m \in M \tag{4.12b}$$

$$\lambda_e = \sum_{l\ni e} \delta_\ell s_\ell \qquad \forall e \in E_1 \tag{4.12c}$$

$$\delta_\ell \geqslant 0 \qquad \forall \ell \in L_1 \tag{4.12d}$$

$$\lambda_e \geqslant 0 \qquad \forall e \in E_1 \tag{4.12e}$$

$$\gamma_m \geqslant 0 \qquad \forall m \in M. \tag{4.12f}$$

Notice that for any feasible solution $(\overline{\gamma}, \overline{\delta}, \overline{\lambda})$ of Problem (4.12) there exist feasible solutions of Problem (4.11) with the same objective value and the same values for $(\overline{\gamma}, \overline{\delta}, \overline{\lambda})$. This is due to the particular structure of the problem in which all the variables λ_e for $e \in E_0$ of Problem (4.11) do not appear in the objective functions, and have an up-monotone structure (an therefore any big enough value of λ_e will provide an optimal solution).

Since Problems (4.10) and (4.11) are the duals of the Benders slave problem and its reformulation, they generate equivalent Benders cuts. For a given optimal solution $(\overline{\gamma}, \overline{\delta}, \overline{\lambda})$ of Problem (4.12), if a set of value λ_e for $e \in E_0$ satisfy the constraints of Problem (4.10) with the other variables set to $\psi_m = \frac{\overline{\gamma_m}}{D_m}$ for all $m \in M$, $\nu_\ell = \overline{\delta}_\ell$ for all $\ell \in L_1$, and $\lambda_e = \overline{\lambda}_e$) for $e \in E_1$, then it generates a feasible Benders cut because its an optimal solution of Problem (4.10). This proves the first part of the theorem. Notice that the variables ϕ_i in Problem (4.10) are not set by the solutions of Problem (4.11), but we know that there exists an optimal solution with those values since the two primal problems are equivalent.

Let $(\lambda', \delta', \gamma')$ be a solution of Problem (4.11). For a given positive vector p, if $(\tilde{\lambda}_e)_{e\in E_0}$ is solution of Problem (4.8), then the cut $\sum_{e\in E_0} \tilde{\lambda}_e x_e + \sum_{e\in E} \lambda'_e x_e + \sum_{m\in M} \gamma'_m \geqslant \eta$ cannot be dominated by other values of λ_e for $e \in E_0$ since in that case, $\tilde{\lambda}$ would not be optimal for Problem (4.8). □

4.6 Numerical results

4.6.1 Instances

These numerical results have been led on mid-haul type instances (see Section 2.6.1 for a detailed presentation of these instances). The following tables gather the results obtained on 10 different instances with different sizes. The same instances have been used to test the direct MILP resolution and the Benders method.

4.6.2 Results

The numerical experiments have been performed on a computer with 7.7 Gb RAM, 8 cores at 1.9 GHz. The algorithms have been developed in `Julia` [19] with the modeling library `JuMP` [44]. All linear programs have been solved using `Gurobi 9.1` [64].

In order to make the Benders decomposition efficient, we have implemented several enhancements to the method. First, since the master problem of the Benders decomposition is an integer

Instance				MILP (4.1)		
n	$\lvert L\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert I\rvert$	gap	memory (Mb)	Relaxation (s)
9	543	115	39852	5.80 %	225	11
11	660	130	51207	6.98 %	254	11
16	860	179	89809	3.67 %	353	40
17	1016	213	116026	2.76 %	417	19
18	1053	226	123497	2.93 %	434	66
20	1270	253	191629	1.48 %	616	128
21	1201	245	143333	2.53 %	485	39
22	1255	253	166517	2.19 %	552	104
22	1597	313	266143	1.47 %	806	293
25	1737	334	359235	1.83 %	1054	546

Table 4.1 – Performances of the direct solve of the MILP (4.1) after 600 seconds

problem, an efficient decomposition usually starts by generating cuts with the continuous relaxation of the problem. When the gap between the upper bound generated by the master problem and the current best solution gets small enough, the master problem can then switch to generating integer solutions. The second enhancement that have been implemented is the primal stabilization of the solution generated by the master problem. This method aims at solving the slave problem for a master solution averaged over the last two iterations. In practice, for a solution x^k generated by the master problem at iteration k solving the slave problem for a solution $x^{\text{sol},k} = (1-\beta)x^k + \beta x^{k-1}$ avoids generating cuts for very distant extreme points of the polyhedron but keeps the cuts generated close from each other.

Table 4.1 presents the results of the direct MILP resolution for Problem (4.1). For each instance, the solver has been stopped after 10 minutes of computation. Each line of the table corresponds to a different instance. The first four columns provide the characteristics of the instances. Namely, the first column gives the number of planes n used, the second one gives the number of legs $|L|$ in the problem, the third one the number of frequency constraints $|\mathcal{F}|$, and the fourth column gives the total number of itineraries $|I|$ in the instance. The following columns present the results obtained with Gurobi. These columns successively indicate the integrality gap obtained at the end of the allocated time, the memory used by the computation (in Mb), and the time (in seconds) used by the solver to get a solution of the linear relaxation of the problem.

Table 4.2 provides the results obtained on the same instances for the two Benders decomposition methods. The lines represent the different instances, and the first four columns give the same information as in Table 4.1. The three following columns present the results obtained for the vanilla Benders decomposition method of Section 4.4. The gap is calculated as the relative difference between the lower bound and the upper bound recorded by the method. In the given time the Benders method did not get to the resolution of the master problem as an integer program. Therefore this gap is not an integrality gap. The column "# iterations" indicates the number of steps performed by the decomposition, and the column "slave" gives the average time taken for solving the slave problem. The last three columns give the same information, but for the Benders decomposition method enhanced by the column generation method described in Section 4.5.

Instance				Classical Benders			CG+Dual Benders		
n	$\lvert L\rvert$	$\lvert\mathcal{F}\rvert$	$\lvert I\rvert$	gap	# iterations	slave	gap	# iterations	slave
9	543	115	39852	31.6 %	32	17.5	16.6%	713	0.4
11	660	130	51207	45.9 %	48	11.8	29.7 %	582	0.5
16	860	179	89809	53.0 %	14	40.3	28.8%	377	0.9
17	1016	213	116026	50.7 %	10	59.3	26.2 %	281	1.5
18	1053	226	123497	53.0 %	6	114.3	31.8 %	235	1.9
20	1270	253	191629	52.4 %	2	252.4	28.1 %	138	3.6
21	1201	245	143333	38.7 %	2	242.5	20.4 %	187	2.3
22	1255	253	166517	37.3 %	3	185.0	21.9 %	174	2.6
22	1597	313	266143	48.6 %	1	576.7	22.3 %	111	4.6
25	1737	334	359235	- %	-	-	31.6 %	64	8.3

Table 4.2 – Performances the two Benders decomposition methods after 600 seconds

4.6.3 Comments

The numerical results show that the Benders decomposition does not compete with the direct use of a solver for the problem. The result in Table 4.1 show that as the size of the problem increases, finding a solution for the relaxation becomes more difficult. However in the time limit of 10 minutes, all the instances are solved with an integrality gap smaller that 10%. For the Benders decomposition method, the given time is not enough to provide a gap small enough between the current solution and the upper bound of the continuous relaxation to get to solving the master problem as an integer program. Therefore, the frontal resolution performs much better than the two Benders decomposition tested.

However, when comparing the two Benders decomposition methods implemented, the method enhanced with the column generation does perform substantially better than the vanilla Benders decomposition. Indeed, with all the instances tested, the number of steps and the gap reached by the enhanced method are always better, as it is illustrated in Table 4.1. In average the gap obtained by the enhanced Benders decomposition method is halved when compared with the vanilla decomposition. The enhanced method even gives results on instances that the vanilla decomposition was not able to start solving in 10 minutes.

5 Load balancing

The present chapter can be read independently of the other chapters. It has no direct relation with the research in airline industry and has been motivated by a question a student asked to Frédéric Meunier in the context of a course in operations research.

5.1 Introduction

Let $G = (V, E)$ be a bipartite graph, whose vertex set V is partitioned into two color classes U and W, and let $d\colon U \to X$ be a "demand" function, where X is $\mathbb{R}_{\geqslant 0}$ or $\mathbb{Z}_{\geqslant 0}$. A vector $\boldsymbol{x} \in X^E$ such that $\sum_{e\in\delta(u)} x_e = d(u)$ for every $u \in U$ *satisfies the demand*. Given such a vector, the *load* of a vertex $w \in W$ is the quantity $\sum_{e\in\delta(w)} x_e$. Here, $\delta(v)$ denotes the edges incident to a vertex v.

We are interested in finding a vector $\boldsymbol{x} \in X^E$ satisfying the demand while "balancing" the loads among the vertices $w \in W$. We consider three natural criteria for balancing the loads: maximize the minimum load; minimize the maximum load; minimize the maximum load difference between any two workers. A concrete motivation for studying such a problem comes from the question of fairly distributing divisible tasks (U) between workers (W): edges model skills and the demand function indicates how much time must be spent on each task.

Formally, the three problems considered in the chapter are the following. They differ only by the objective function.

$$\begin{array}{lll} \text{Minimize} & \displaystyle\max_{w\in W} \sum_{e\in\delta(w)} x_e & \\ \text{subject to} & \displaystyle\sum_{e\in\delta(u)} x_e = d(u) & \forall\, u \in U \\ & x_e \in X & \forall\, e \in E, \end{array} \tag{Min-max^X}$$

$$\begin{array}{lll} \text{Maximize} & \displaystyle\min_{w\in W} \sum_{e\in\delta(w)} x_e & \\ \text{subject to} & \displaystyle\sum_{e\in\delta(u)} x_e = d(u) & \forall\, u \in U \\ & x_e \in X & \forall\, e \in E, \end{array} \tag{Max-min^X}$$

$$\begin{array}{lll} \text{Minimize} & \displaystyle\max_{w,w'\in W} \Big(\sum_{e\in\delta(w)} x_e - \sum_{e\in\delta(w')} x_e \Big) & \\ \text{subject to} & \displaystyle\sum_{e\in\delta(u)} x_e = d(u) & \forall\, u \in U \\ & x_e \in X & \forall\, e \in E. \end{array} \tag{Min-diff^X}$$

We have the following fact, which is the main result of the chapter and its original motivation. This is arguably surprising since the objective functions do not order the feasible solutions in the same way. It means that we can always satisfy the demand in a way that optimally balances the loads according to the three criteria at the same time.

Theorem 26. *Every optimal solution of Problem* (Min-diff^X) *is simultaneously optimal for Problems* (Min-max^X) *and* (Max-min^X), *whether* $X = \mathbb{R}_{\geqslant 0}$ *or* $X = \mathbb{Z}_{\geqslant 0}$.

It is easy to see that the reverse statement does not hold: there are optimal solutions to Problems (Min-max^X) and (Max-min^X) that are not optimal for Problem (Min-diff^X).

We also prove the following proposition. When $X = \mathbb{R}_{\geqslant 0}$, it is immediate since the problem reduces then to a linear program with bounded coefficients in the constraint matrix (which can be solved in strongly polynomial time by Tardos's algorithm [132]).

Proposition 27. *Problem* (Min-diff^X) *can be solved in strongly polynomial time, whether* $X = \mathbb{R}_{\geqslant 0}$ *or* $X = \mathbb{Z}_{\geqslant 0}$.

The rest of the chapter deals with the proof of Theorem 26 and extensions of this result. Section 5.2 presents the proofs of Theorem 26, Proposition 27 and several results specific to Problem (Min-diff^X). We then show in Section 5.3 that the three problems enjoy min-max relations with other problems of combinatorial nature. Section 5.4 considers more general linear constraints and objective functions and shows that Theorem 26 still holds in that case.

5.2 Min-diff problem

We start by introducing the following notation. We denote by D^X the set of feasible solutions for the problems, namely $\{x \in X \mid \sum_{e\in\delta(u)} x_e = d(u), \forall u \in U\}$. For a solution $x \in D^X$, we denote by $\ell^{\max}(x)$ and $\ell^{\min}(x)$ the objective values of Problems (Min-max^X) and (Max-min^X).

In the present section, we start by proving a structural result on the set D^X, which is then used to prove Theorem 26, first in the case $X = \mathbb{R}_{\geqslant 0}$ and then in the case $X = \mathbb{Z}_{\geqslant 0}$. Then, we introduce a generalization of the problems and show that it has an interesting relation with Problem (Min-diff^X). Finally, we focus on the algorithmic result stated in Proposition 27.

5.2.1 Structural property of D^X

The following lemma is crucial for proving Theorem 26 and results stated later in the chapter.

Lemma 28. *Suppose that* $D^{\mathbb{R}_{\geqslant 0}}$ *is nonempty and that* $d(u) > 0$ *for all* $u \in U$. *Then there exist a partition* $U_1, \ldots, U_s$ *of* U *and an element* $\bar{\boldsymbol{x}}$ *in* $D^{\mathbb{R}_{\geqslant 0}}$ *such that:*

(i) *The sets* $W_i = N(U_i) \setminus N(U_1 \cup \cdots \cup U_{i-1})$ *for* $i \in [s]$ *are all nonempty.*

(ii) *For every* $i \in [s]$ *we have*

$$\sum_{e\in\delta(w)} \bar{x}_e = \frac{d(U_i)}{|W_i|} \quad \text{for } w \in W_i\,.$$

(iii) *We have*

$$\frac{d(U_1)}{|W_1|} \geqslant \frac{d(U_2)}{|W_2|} \geqslant \cdots \geqslant \frac{d(U_s)}{|W_s|}\,.$$

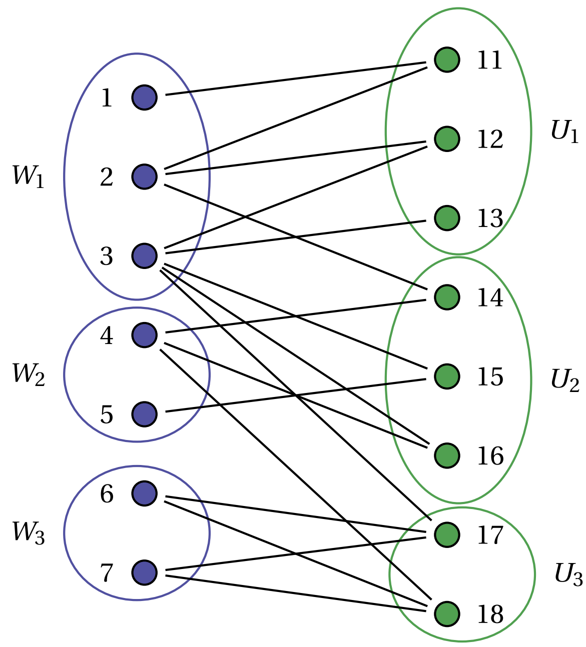

Figure 5.1 – Example of the structure of D^X described in Lemma 28.

The structure of the problem is illustrated in an example in Figure 5.1.

Proof. The proof works by induction on $|U|$. If $|U| = 1$, the conclusion is immediate.

Suppose that $|U| \geqslant 2$. For an element $\boldsymbol{y}$ in $D^{\mathbb{R}_{\geqslant 0}}$, we define $W^{\max}(\boldsymbol{y})$ as the set of vertices w in W for which the maximum is attained in the definition of $\ell^{\max}(\boldsymbol{y})$. We define also $U^{\max}(\boldsymbol{y})$ to be the set of vertices $u \in U$ for which there exists $w \in W^{\max}(\boldsymbol{y})$ with $y_{uw} > 0$. We make the following claim: *For Problem* (Min-max$^{\mathbb{R}_{\geqslant 0}}$)*, if $\boldsymbol{y}$ is an optimal solution such that $N(U^{\max}(\boldsymbol{y})) \neq W^{\max}(\boldsymbol{y})$, then there exists an optimal solution $\boldsymbol{y}'$ such that $W^{\max}(\boldsymbol{y}') \subsetneq W^{\max}(\boldsymbol{y})$ and $\ell^{\max}(\boldsymbol{y}') = \ell^{\max}(\boldsymbol{y})$.* Once proved, this claim will easily lead to the desired conclusion.

We now prove the claim. Let $\boldsymbol{y}$ be an element of $D^{\mathbb{R}_{\geqslant 0}}$ such that $N(U^{\max}(\boldsymbol{y})) \neq W^{\max}(\boldsymbol{y})$. Since $W^{\max}(\boldsymbol{y}) \subseteq N(U^{\max}(\boldsymbol{y}))$ by definition ($d(u) > 0$ for at least one u), there exists $u \in U^{\max}(\boldsymbol{y})$ and $w' \in W \setminus W^{\max}(\boldsymbol{y})$ such that $uw' \in E$. Still by definition, there exists $w \in W^{\max}(\boldsymbol{y})$ such that $y_{uw} > 0$. Set then

$$y'_e = \begin{cases} y_e & \text{if } e \notin \{uw, uw'\}, \\ y_e - \varepsilon & \text{if } e = uw, \\ y_e + \varepsilon & \text{if } e = uw', \end{cases}$$

where $0 < \varepsilon < \min(y_{uw}, \ell^{\max}(\boldsymbol{y}) - y_{uw'})$, chosen arbitrarily. The vector $\boldsymbol{y}'$ is still a feasible solution of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$). The way $\boldsymbol{y}'$ is built implies that $\ell^{\max}(\boldsymbol{y}') \leqslant \ell^{\max}(\boldsymbol{y})$. Since $\boldsymbol{y}$ is an optimal solution of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$), we have $\ell^{\max}(\boldsymbol{y}') = \ell^{\max}(\boldsymbol{y})$. The claim follows then from the fact that $W^{\max}(\boldsymbol{y}')$ is a strict subset of $W^{\max}(\boldsymbol{y})$.

By finiteness, the claim implies the existence of an optimal solution $\bar{\boldsymbol{x}}'$ to Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$)

such that $N(U^{\max}(\bar{\boldsymbol{x}}')) = W^{\max}(\bar{\boldsymbol{x}}')$. Since we have

$$|W^{\max}(\bar{\boldsymbol{x}}')|\ell^{\max}(\bar{\boldsymbol{x}}') = \sum_{w\in W^{\max}(\bar{\boldsymbol{x}}')}\sum_{e\in\delta(w)} \bar{x}'_e = \sum_{u\in U^{\max}(\bar{\boldsymbol{x}}')}\sum_{e\in\delta(u)} \bar{x}'_e = d(U^{\max}(\bar{\boldsymbol{x}}'))\,,$$

we have $\ell^{\max}(\bar{\boldsymbol{x}}') = \frac{d(U^{\max}(\bar{\boldsymbol{x}}'))}{|N(U^{\max}(\bar{\boldsymbol{x}}'))|}$. Set $U_1 = U^{\max}(\bar{\boldsymbol{x}}')$ and $W_1 = N(U_1)$ (which is obviously nonempty). By induction, there exist a partition $U_2,\dots,U_s$ of $U \setminus U_1$ and an element $\bar{\boldsymbol{x}}'' \in \mathbb{R}_{\geqslant 0}^{E\setminus\delta(W_1)}$ such that $\sum_{e\in\delta(u)} x''_e = d(u)$ for all $u \in U \setminus U_1$ and such that:

- The sets $W_i = N(U_i) \setminus N(U_1 \cup \cdots \cup U_{i-1})$ for $i \in \{2,\dots,s\}$ are all nonempty.
- For every $i \in \{2,\dots,s\}$ we have

$$\sum_{e\in\delta(w)} \bar{x}''_e = \frac{d(U_i)}{|W_i|} \quad \text{for } w \in W_i\,.$$

- We have

$$\frac{d(U_2)}{|W_2|} \geqslant \cdots \geqslant \frac{d(U_s)}{|W_s|}\,.$$

Item (i) is satisfied for all $i \in [s]$. Define

$$\bar{x}_e = \begin{cases} \bar{x}'_e & \text{if } e \in \delta(U_1)\,,\\ \bar{x}''_e & \text{if } e \in \delta(W_i) \text{ for } i \in \{2,\dots,s\}\,,\\ 0 & \text{otherwise}\,. \end{cases}$$

It is an element of $D^{\mathbb{R}_{\geqslant 0}}$, which satisfies Item (ii) for all $i \in [s]$ (for $s = 1$, it is because $W^{\max}(\bar{\boldsymbol{x}}') = W_1$). On the other hand, we have $\sum_{e\in\delta(w)} \bar{x}'_e \geqslant \frac{d(U_2)}{|W_2|}$ for at least one $w \in W_2$ because there is no edge between U_2 and W_i for $i \geqslant 3$ and $\bar{x}'_e = 0$ for every edge e between U_2 and W_1. Since $\ell^{\max}(\bar{\boldsymbol{x}}') = \frac{d(U_1)}{|W_1|}$, Item (iii) is satisfied as well. $\square$

5.2.2 Proof of Theorem 26 when $X = \mathbb{R}_{\geqslant 0}$

Proof when $X = \mathbb{R}_{\geqslant 0}$. Let us denote by $l^{\star\max}$ (respectively $l^{\star\min}$) the optimal value of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$) (respectively Problem (Max-min$^{\mathbb{R}_{\geqslant 0}}$)).

Let $U_1,\dots,U_s$ be a partition as in Lemma 28. Notice that $d(U_1)$ must be allocated exclusively among elements of W_1. Since in a collection of numbers there is always one number non-smaller than the average, there is at least one w in W_1 for which $\sum_{e\in\delta(w)} x_e$ is non-smaller than $\frac{d(U_1)}{|W_1|}$ for any $\boldsymbol{x} \in D^X$. It implies that $l^{\star\max} \geqslant \frac{d(U_1)}{|W_1|}$.

Similarly, notice that the elements of W_s are only linked with elements of U_s. Since in a collection of numbers there is always one number non-larger than the average, there is at least one w in W_s for which $\sum_{e\in\delta(w)} x_e$ is non-larger than $\frac{d(U_s)}{|W_s|}$. It implies that $l^{\star\min} \leqslant \frac{d(U_s)}{|W_s|}$.

Denoting by $l^{\star\text{diff}}$ the optimal solution of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$), we get $l^{\star\text{diff}} \geqslant \frac{d(U_1)}{|W_1|} - \frac{d(U_s)}{|W_s|}$. Since Lemma 28 provides a solution $\overline{\boldsymbol{x}}$ that realizes this bound, we get $l^{\star\text{diff}} = \frac{d(U_1)}{|W_1|} - \frac{d(U_s)}{|W_s|}$.

Take now any optimal solution $\boldsymbol{x}^\star$ of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$). Since $\ell^{\max}(\boldsymbol{x}^\star) \geqslant l^{\star\max}$ and $\ell^{\min}(\boldsymbol{x}^\star) \leqslant l^{\star\min}$ the equation $l^{\star\text{diff}} = \frac{d(U_1)}{|W_1|} - \frac{d(U_s)}{|W_s|}$ gives that $\ell^{\max}(\boldsymbol{x}^\star) = l^{\star\max}$ and thus that $\boldsymbol{x}^\star$ is an optimal solution of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$). The same results goes for Problem (Max-min$^{\mathbb{R}_{\geqslant 0}}$). This ends the proof of Theorem 26 in the case $X = \mathbb{R}_{\geqslant 0}$. □

Remark 12. The proof also shows that for every optimal solution $\boldsymbol{x}^*$ of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$), there exist $U_1 \subseteq U$ and $W_s \subseteq W$ such that

$$\ell^{\max}(\boldsymbol{x}^*) = \frac{d(U_1)}{|N(U_1)|} \qquad \text{and} \qquad \ell^{\min}(\boldsymbol{x}^*) = \frac{d(N(W_s))}{|W_s|}.$$

5.2.3 Proof of Theorem 26 when $X = \mathbb{Z}_{\geqslant 0}$

A natural idea to approach the integer version is to start from an optimal solution of the continuous version, and to round it via a flow argument. This is actually the right way to proceed.

Assume that Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) is feasible (in particular, all $d(u)$ are integral). Let $\boldsymbol{x}^*$ be any optimal solution of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$).

We define a directed graph $D = (V', A)$ on which a "rounding flow" argument will be applied. Its vertex set, V', is $V = U \cup W$, to which we add two new vertices s and t. Its arc set, A, is obtained as follows. We orient all edges in E from U to W and we add all arcs of the form (s, u), with $u \in U$, and all arcs of the form (w, t), with $w \in W$.

Every arc a in A has *lower capacity* $\underline{c}(a)$ and *upper capacity* $\overline{c}(a)$:

- For $a = (s, u)$ with $u \in U$, we set $\underline{c}(a) = \overline{c}(a) = d(u)$.
- For $a = (u, w)$ with $uw \in E$, we set $\underline{c}(a) = 0$ and $\overline{c}(a) = +\infty$.
- For $a = (w, t)$ with $w \in W$, we set $\underline{c}(a) = \lfloor \ell(w) \rfloor$ and $\overline{c}(a) = \lceil \ell(w) \rceil$, where $\ell(w)$ is the load $\sum_{e \in \delta(w)} x_e^*$ of vertex w.

Note that all capacities are integer numbers.

The solution $\boldsymbol{x}^*$ shows the existence of a feasible s-t flow in D. According to the integrality property of flows, there exists an integer feasible solution $\boldsymbol{z}^*$ to Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) such that

$$\lfloor \ell^{\max}(\boldsymbol{x}^*) \rfloor \leqslant \ell^{\max}(\boldsymbol{z}^*) \leqslant \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil \quad \text{and} \quad \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor \leqslant \ell^{\min}(\boldsymbol{z}^*) \leqslant \lceil \ell^{\min}(\boldsymbol{x}^*) \rceil. \tag{5.1}$$

We have actually more, as stated by the following proposition.

Proposition 29. *The solution $\boldsymbol{z}^*$ is simultaneously optimal for Problems* (Min-max$^{\mathbb{Z}_{\geqslant 0}}$), (Max-min$^{\mathbb{Z}_{\geqslant 0}}$), *and* (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$), *and the following equalities hold:*

$$\ell^{\max}(\boldsymbol{z}^*) = \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil \qquad \textit{and} \qquad \ell^{\min}(\boldsymbol{z}^*) = \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor.$$

Proof. Following Remark 12, the quantity $\ell^{\max}(\boldsymbol{x}^*)$ is the optimal value of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$) and thus, we have $\ell^{\max}(\boldsymbol{z}^*) \geqslant \ell^{\max}(\boldsymbol{x}^*)$ since $\boldsymbol{z}^*$ is also a feasible solution of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$). Combining this inequality with the left-hand one of Equation (5.1) and with the integrality of $\ell^{\max}(\boldsymbol{z}^*)$, we get that $\ell^{\max}(\boldsymbol{z}^*) = \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil$. Similarly, we have $\ell^{\min}(\boldsymbol{z}^*) = \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor$.

Consider an optimal solution $\bar{\boldsymbol{y}}$ of Problem (Min-max$^{\mathbb{Z}_{\geqslant 0}}$). Since $\bar{\boldsymbol{y}}$ is a feasible solution of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$), we have, thanks to Remark 12, the inequality $\ell^{\max}(\bar{\boldsymbol{y}}) \geqslant \ell^{\max}(\boldsymbol{x}^*)$, which in turn implies $\ell^{\max}(\bar{\boldsymbol{y}}) \geqslant \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil$ since $\bar{\boldsymbol{y}}$ is an integer solution. The solution $\boldsymbol{z}^*$ is a feasible solution of Problem (Min-max$^{\mathbb{Z}_{\geqslant 0}}$) and satisfies $\ell^{\max}(\boldsymbol{z}^*) = \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil$. Therefore, $\lceil \ell^{\max}(\boldsymbol{x}^*) \rceil$ is the optimal value of Problem (Min-max$^{\mathbb{Z}_{\geqslant 0}}$) and $\boldsymbol{z}^*$ is an optimal solution of that problem. The case of Problem (Max-min$^{\mathbb{Z}_{\geqslant 0}}$) is dealt with similarly.

Consider now an optimal solution $\boldsymbol{y}^*$ of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$). Since $\boldsymbol{x}^*$ is simultaneously optimal for Problems (Min-max$^{\mathbb{R}_{\geqslant 0}}$) and (Max-min$^{\mathbb{R}_{\geqslant 0}}$), we have $\ell^{\max}(\boldsymbol{y}^*) \geqslant \ell^{\max}(\boldsymbol{x}^*)$ and $\ell^{\min}(\boldsymbol{y}^*) \leqslant \ell^{\min}(\boldsymbol{x}^*)$. Hence, we have $\ell^{\max}(\boldsymbol{y}^*) \geqslant \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil$ and $\ell^{\min}(\boldsymbol{y}^*) \leqslant \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor$ since $\boldsymbol{y}^*$ is an integer solution. Thus, we have $\ell^{\max}(\boldsymbol{y}^*) - \ell^{\min}(\boldsymbol{y}^*) \geqslant \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil - \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor$, which shows in turn that the optimal value of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) is at least $\lceil \ell^{\max}(\boldsymbol{x}^*) \rceil - \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor$. Therefore, $\lceil \ell^{\max}(\boldsymbol{x}^*) \rceil - \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor$ is the optimal value of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) and $\boldsymbol{z}^*$ is an optimal solution of that problem. □

With Proposition 29 in hand, we can use the result in the case $X = \mathbb{R}_{\geqslant 0}$ to finish the proof of Theorem 26.

Proof of Theorem 26 when $X = \mathbb{Z}_{\geqslant 0}$. Suppose that Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) is feasible. Consider an optimal solution $\boldsymbol{y}^*$ of that problem. The result of the theorem in the case $X = \mathbb{R}_{\geqslant 0}$ that $\boldsymbol{x}^*$ is simultaneously optimal for Problems (Min-max$^{\mathbb{R}_{\geqslant 0}}$) and (Max-min$^{\mathbb{R}_{\geqslant 0}}$). We have thus $\ell^{\max}(\boldsymbol{y}^*) \geqslant \ell^{\max}(\boldsymbol{x}^*)$ and $\ell^{\min}(\boldsymbol{y}^*) \leqslant \ell^{\min}(\boldsymbol{x}^*)$, i.e.,

$$\ell^{\max}(\boldsymbol{y}^*) \geqslant \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil \qquad \text{and} \qquad -\ell^{\min}(\boldsymbol{y}^*) \geqslant -\lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor\,, \tag{5.2}$$

since $\boldsymbol{y}^*$ is an integer solution. The equality

$$\ell^{\max}(\boldsymbol{y}^*) - \ell^{\min}(\boldsymbol{y}^*) = \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil - \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor \tag{5.3}$$

holds since its right-hand side is the optimal value of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) by Proposition 29. In Equation (5.2), the sum of the left-hand sides of the two inequalities is at least the sum of the right-hand sides. Equation (5.3) shows that these sums are equal, which implies that

$$\ell^{\max}(\boldsymbol{y}^*) = \lceil \ell^{\max}(\boldsymbol{x}^*) \rceil \quad \text{and} \quad \ell^{\min}(\boldsymbol{y}^*) = \lfloor \ell^{\min}(\boldsymbol{x}^*) \rfloor\,. \tag{5.4}$$

The solution $\boldsymbol{y}^*$ is an optimal solution of Problems (Min-max$^{\mathbb{Z}_{\geqslant 0}}$) and (Max-min$^{\mathbb{Z}_{\geqslant 0}}$) since the right-hand sides of Equations (5.4) are the optimal values of these problems by Proposition 29. □

5.2.4 A more general problem

We consider the following problem where the function f is assumed to be convex.

$$\begin{array}{ll} \text{Minimize} & \displaystyle\sum_{w \in W} f\Big(\sum_{e \in \delta(w)} x_e\Big) \\ \text{subject to} & \boldsymbol{x} \in D^X\,. \end{array} \tag{Min-sum^{f,X}}$$

The particular case of this problem when $X = \mathbb{Z}_{\geqslant 0}$ and $d(u) = 1$ for all $u \in U$ has been considered by Harvey et al. [68]. Feasible solutions are then in this particular case called "semi-matchings."

The following theorem for $X = \mathbb{Z}_{\geqslant 0}$ is a consequence of their Corollary 3.6, which was stated only for semi-matchings but the extension to the general integer case is immediate. The result for $X = \mathbb{R}_{\geqslant 0}$ is new.

Theorem 30. *Every optimal solution of Problem* (Min-sum^{f,X}) *with respect to some strictly convex function f is optimal with respect to all convex functions, whether $X = \mathbb{R}_{\geqslant 0}$ or $X = \mathbb{Z}_{\geqslant 0}$.*

Harvey et al. [68] also proved (Theorem 3.12 of their paper) that every optimal semi-matching provides an optimal solution of Problem (Min-max$^{\mathbb{Z}_{\geqslant 0}}$). We prove a stronger result (because of Theorem 26).

Proposition 31. *Every optimal solution of Problem* (Min-sum^{f,X}) *with respect to some collection of strictly convex functions is optimal for Problem* (Min-diff^{X})*, whether $X = \mathbb{R}_{\geqslant 0}$ or $X = \mathbb{Z}_{\geqslant 0}$.*

It is easy to come up with examples showing that the reverse of this proposition is not true. So, there is no direct way to prove Theorem 26 for $X = \mathbb{Z}_{\geqslant 0}$ with the help of Theorem 3.12 of Harvey et al.

The main contributions of Harvey et al. [68] are efficient polynomial algorithms for computing optimal semi-matchings. There is seemingly no easy way to derive the next theorem from their algorithms or other known polynomial algorithms for computing optimal semi-matchings.

Theorem 32. *Problem* (Min-sum^{f,X}) *can be solved in strongly polynomial time, whether $X = \mathbb{R}_{\geqslant 0}$ or $X = \mathbb{Z}_{\geqslant 0}$.*

Notice that Theorems 30 and 32 in the integer cases can be deduced from results of Frank and Murota [51, 52] where they develop results and algorithms of decreasingly minimization problems on M-convex sets.

5.2.4.1 Technical tools for the continuous case

Lemma 33. *Suppose that $D^{\mathbb{R}_{\geqslant 0}}$ is nonempty and that $d(u) > 0$ for all $u \in U$. Consider a partition $U_1, \ldots, U_s$ and an element $\bar{\boldsymbol{x}} \in D^{\mathbb{R}_{\geqslant 0}}$ as in Lemma 28.*

If an element $\boldsymbol{x}^* \in D^{\mathbb{R}_{\geqslant 0}}$ satisfies for every $i \in [s]$ and every $w \in W_i$ the equality

$$\sum_{e \in \delta(w)} x_e^* = \frac{d(U_i)}{|W_i|}, \tag{5.5}$$

then $\boldsymbol{x}^*$ is an optimal solution of Problem (Min-sum$^{f,\mathbb{R}_{\geqslant 0}}$)*. In particular, the element $\bar{\boldsymbol{x}}$ is such an optimal solution.*

If f is strictly convex, then the converse holds: each optimal solution $\boldsymbol{x}^*$ of Problem (Min-sum$^{f,\mathbb{R}_{\geqslant 0}}$) *satisfies Equation* (5.5) *for every $i \in [s]$ and every $w \in W_i$.*

Proof. We start by proving that, for every $\boldsymbol{x} \in D^{\mathbb{R}_{\geqslant 0}}$, we have

$$\sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e \in \delta(W_i)} x_e}{|W_i|}\right) \geqslant \sum_{i=1}^{s} |W_i| f\left(\frac{d(U_i)}{|W_i|}\right). \tag{5.6}$$

Let $\tilde{\boldsymbol{x}} \in D^{\mathbb{R}_{\geqslant 0}}$ minimize $\sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e\in\delta(W_i)} x_e}{|W_i|}\right)$. Choose $\tilde{\boldsymbol{x}}$ such that the number of edges e satisfying

$$\tilde{x}_e > 0 \quad \text{and} \quad e \in \delta(U_i) \cap \delta(W_{i'}) \quad \text{with} \quad i \neq i' \tag{5.7}$$

is as small as possible. Set $a_i = \sum_{e\in\delta(W_i)} \tilde{x}_e$ for all i.

Suppose for a contradiction that there exists i such that $a_i \neq d(U_i)$. In particular, since $\sum_i a_i = \sum_i d(U_i)$, there exists j such that $a_j < d(U_j)$. Choose such a j as small as possible. There exists $k < j$ such that $\tilde{x}_{\bar{e}} > 0$ for some $\bar{e} \in \delta(U_j) \cap \delta(W_k)$. Pick any edge $\bar{e}'$ in $\delta(U_j) \cap \delta(W_j)$ (which is nonempty by definition) and define $\tilde{\boldsymbol{x}}' \in \mathbb{R}^E_{\geqslant 0}$ as follows:

$$\tilde{x}'_e = \begin{cases} \tilde{x}_e & \text{if } e \notin \{\bar{e}, \bar{e}'\} \\ 0 & \text{if } e = \bar{e} \\ \tilde{x}_{\bar{e}'} + \tilde{x}_{\bar{e}} & \text{if } e = \bar{e}'. \end{cases}$$

The vector $\tilde{\boldsymbol{x}}'$ belongs to $D^{\mathbb{R}_{\geqslant 0}}$. Since we have

$$\frac{a_j}{|W_j|} < \frac{d(U_j)}{|W_j|} \leqslant \frac{d(U_k)}{|W_k|} \leqslant \frac{a_k}{|W_k|},$$

the convexity of f implies that

$$\frac{|W_j|}{\tilde{x}_{\bar{e}}}\left(f\left(\frac{a_j}{|W_j|} + \frac{\tilde{x}_{\bar{e}}}{|W_j|}\right) - f\left(\frac{a_j}{|W_j|}\right)\right) \leqslant \frac{|W_k|}{\tilde{x}_{\bar{e}}}\left(f\left(\frac{a_k}{|W_k|}\right) - f\left(\frac{a_k}{|W_k|} - \frac{\tilde{x}_{\bar{e}}}{|W_k|}\right)\right).$$

Set $a'_i = \sum_{e\in\delta(W_i)} \tilde{x}'_e$ for all i. The previous inequality can be written:

$$|W_j| f\left(\frac{a'_j}{|W_j|}\right) + |W_k| f\left(\frac{a'_k}{|W_k|}\right) \leqslant |W_j| f\left(\frac{a_j}{|W_j|}\right) + |W_k| f\left(\frac{a_k}{|W_k|}\right), \tag{5.8}$$

which shows that $\tilde{\boldsymbol{x}}'$ is also an element of $D^{\mathbb{R}_{\geqslant 0}}$ minimizing $\sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e\in\delta(W_i)} x_e}{|W_i|}\right)$. But this contradicts the fact that $\tilde{\boldsymbol{x}}$ was chosen with a minimum number of edges satisfying (5.7). This shows that $a_i = d(U_i)$ for all $i \in [s]$. Now,

$$\sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e\in\delta(W_i)} \tilde{x}_e}{|W_i|}\right) = \sum_{i=1}^{s} |W_i| f\left(\frac{d(U_i)}{|W_i|}\right),$$

and Equation (5.6) is therefore satisfied for all elements in $D^{\mathbb{R}_{\geqslant 0}}$.

Consider $\boldsymbol{x}^*$ satisfying Equation (5.5) for every $i \in [s]$ and every $w \in W_i$. We get the first part of the lemma by noticing that, for every $\boldsymbol{x} \in D^{\mathbb{R}_{\geqslant 0}}$, we have

$$\sum_{w\in W} f\Big(\sum_{e\in\delta(w)} x_e\Big) \geqslant \sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e\in\delta(W_i)} x_e}{|W_i|}\right) \geqslant \sum_{i=1}^{s} |W_i| f\left(\frac{d(U_i)}{|W_i|}\right) = \sum_{w\in W} f\Big(\sum_{e\in\delta(w)} x^*_e\Big) \tag{5.9}$$

(the first inequality is a consequence of the convexity of f), and thus $\boldsymbol{x}^*$ is an optimal solution of Problem (Min-sum$^{f,\mathbb{R}_{\geqslant 0}}$).

To finish the proof, consider now the case when f is strictly convex. We prove first that the inequality in Equation (5.6) is strict if there is a i for which $\sum_{e\in\delta(W_i)} x_e \neq d(U_i)$. Let again $\tilde{\boldsymbol{x}} \in D^{\mathbb{R}_{\geqslant 0}}$ minimize $\sum_{i=1}^{s} |W_i| f\left(\frac{\sum_{e\in\delta(W_i)} x_e}{|W_i|}\right)$. This time, do not assume anything about the number of edges satisfying (5.7). Define $\tilde{\boldsymbol{x}}'$ as before. We see that, in this case, Equation (5.8) is strict. Therefore, when f is strictly convex, $\tilde{\boldsymbol{x}}$ necessarily satisfies $\sum_{e\in\delta(W_i)} \tilde{x}_e = d(U_i)$, and thus, if there is a i for which $\sum_{e\in\delta(W_i)} x_e \neq d(U_i)$, then the inequality in Equation (5.6) is strict. The conclusion is then immediate: let $\boldsymbol{x}^*$ be any optimal solution of Problem (Min-sum$^{f,\mathbb{R}_{\geqslant 0}}$); consider Equation (5.9) for $\boldsymbol{x} = \boldsymbol{x}^*$; all inequalities are then equalities; strict convexity implies that $\sum_{e\in\delta(w)} x_e^* = \frac{\sum_{e\in\delta(W_i)} x_e^*}{|W_i|}$ (from the first inequality in Equation (5.9), which can be in this case decomposed along the i's), and thus that $\sum_{e\in\delta(w)} x_e^* = \frac{d(U_i)}{|W_i|}$. □

5.2.4.2 Technical tools for the integer case

Lemma 34. *Let g be a convex function $\mathbb{R} \to \mathbb{R}$. Let $x_1, \ldots, x_m$ be integer numbers. Denote by s their sum and write $s = qm + r$ with $q, r \in \mathbb{Z}$ and $0 \leqslant r < s$ (Euclidean division). Then*

$$\sum_{i=1}^{m} g(x_i) \geqslant (m-r)g(\lfloor s/m \rfloor) + rg(\lceil s/m \rceil).$$

If g is strictly convex, then the previous inequality is an equality only in two cases:

- *either s/m is an integer and every x_i is equal to s/m,*
- *or s/m is not an integer, and the number of x_i that are equal to $\lceil s/m \rceil$ is r and the number of x_i that are equal to $\lfloor s/m \rfloor$ is $s - r$.*

Proof. By convexity, we have $g(z) + g(y) \geqslant g(z-1) + g(y+1)$ for any numbers $y < z$. We can transform the collection $x_1, \ldots, x_m$ into a collection of $m - r$ numbers equal to $\lfloor s/m \rfloor$ and r numbers equal to $\lceil s/m \rceil$, by subtracting repeatedly one to the largest number and adding one to the smallest number. The sum of the evaluations of g on the m numbers remains either constant or decreases at each such iteration.

We get the second part of the lemma by noticing that, when g is strictly convex, the inequality $g(z) + g(y) \geqslant g(z-1) + g(y+1)$ for $y < z$ is an equality only if $y + 1 = z$. To keep equality along the iterations implies that the smallest x_i and the largest x_i initially differ only by at most one, which is exactly the desired result. □

Lemma 35. *Suppose that $D^{\mathbb{Z}_{\geqslant 0}}$ is nonempty and that $d(u) > 0$ for all $u \in U$. Consider a partition $U_1, \ldots, U_s$ and an element $\bar{\boldsymbol{x}} \in D^{\mathbb{R}_{\geqslant 0}}$ as in Lemma 28.*

If an element $\boldsymbol{x}^* \in D^{\mathbb{Z}_{\geqslant 0}}$ satisfies for every $i \in [s]$ the relations

a) $\displaystyle\sum_{e\in\delta(W_i)} x_e^* = d(U_i)$ *and*

b) $\displaystyle\sum_{e\in\delta(w)} x_e^* \in \left\{\left\lfloor \frac{d(U_i)}{|W_i|} \right\rfloor, \left\lceil \frac{d(U_i)}{|W_i|} \right\rceil\right\}$ *for every $w \in W_i$,*

then $\boldsymbol{x}^*$ is an optimal solution of Problem (Min-sum$^{f,\mathbb{Z}_{\geqslant 0}}$). *Moreover, such an optimal solution exists.*

If f is strictly convex, then the converse holds: each optimal solution $\boldsymbol{x}^*$ of Problem (Min-sum$^{f,\mathbb{Z}_{\geqslant 0}}$)

satisfies Items a) *and* b) *for every* $i \in [s]$ *and every* $w \in W_i$.

Proof. We introduce some notation. Write $\rho_i = d(U_i) - |W_i| \left\lfloor \frac{d(U_i)}{|W_i|} \right\rfloor$. Set $a_i^{\boldsymbol{x}} = \sum_{e \in \delta(W_i)} x_e$ and $r_i^{\boldsymbol{x}} = a_i^{\boldsymbol{x}} - |W_i| \left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rfloor$. The number ρ_i (resp. $r_i^{\boldsymbol{x}}$) is thus the remainder of the Euclidean division of $d(U_i)$ (resp. $a_i^{\boldsymbol{x}}$) by $|W_i|$.

We start by proving that, for every $\boldsymbol{x} \in D^{\mathbb{Z}_{\geqslant 0}}$, we have

$$\sum_{i=1}^{s} r_i^{\boldsymbol{x}} f\left(\left\lceil \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rceil\right) + (|W_i| - r_i^{\boldsymbol{x}}) f\left(\left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rfloor\right) \geqslant \sum_{i=1}^{s} \rho_i f\left(\left\lceil \frac{d(U_i)}{|W_i|} \right\rceil\right) + (|W_i| - \rho_i) f\left(\left\lfloor \frac{d(U_i)}{|W_i|} \right\rfloor\right). \quad (5.10)$$

Let $\tilde{\boldsymbol{x}} \in D^{\mathbb{Z}_{\geqslant 0}}$ minimize $\sum_{i=1}^{s} r_i^{\boldsymbol{x}} f\left(\left\lceil \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rceil\right) + (|W_i| - r_i^{\boldsymbol{x}}) f\left(\left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rfloor\right)$. Choose $\tilde{\boldsymbol{x}}$ such that the quantity $\sum_{i \neq i'} \sum_{e \in \delta(U_i) \cap \delta(W_{i'})} \tilde{x}_e$ is as small as possible. Suppose for a contradiction that there exists i such that $a_i^{\tilde{\boldsymbol{x}}} \neq d(U_i)$. In particular, since $\sum_i a_i^{\tilde{\boldsymbol{x}}} = \sum_i d(U_i)$, there exists j such that $a_j^{\tilde{\boldsymbol{x}}} < d(U_j)$. Choose such a j as small as possible. There exists $k < j$ such that $\tilde{x}_{\bar{e}} > 0$ for some $\bar{e} \in \delta(U_j) \cap \delta(W_k)$. Pick any edge $\bar{e}'$ in $\delta(U_j) \cap \delta(W_j)$ (which is nonempty by definition) and define $\tilde{\boldsymbol{x}}' \in \mathbb{Z}_{\geqslant 0}^E$ as follows:

$$\tilde{x}'_e = \begin{cases} \tilde{x}_e & \text{if } e \notin \{\bar{e}, \bar{e}'\}, \\ \tilde{x}_{\bar{e}} - 1 & \text{if } e = \bar{e}, \\ \tilde{x}_{\bar{e}'} + 1 & \text{if } e = \bar{e}'. \end{cases}$$

The vector $\tilde{\boldsymbol{x}}'$ belongs to $D^{\mathbb{Z}_{\geqslant 0}}$. We have

$$\begin{aligned} r_j^{\tilde{\boldsymbol{x}}'} f\left(\left\lceil \frac{a_j^{\tilde{\boldsymbol{x}}'}}{|W_j|} \right\rceil\right) + (|W_j| - r_j^{\tilde{\boldsymbol{x}}'}) f\left(\left\lfloor \frac{a_j^{\tilde{\boldsymbol{x}}'}}{|W_j|} \right\rfloor\right) \\ = r_j^{\tilde{\boldsymbol{x}}} f\left(\left\lceil \frac{a_j^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rceil\right) + (|W_j| - r_j^{\tilde{\boldsymbol{x}}} - 1) f\left(\left\lfloor \frac{a_j^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rfloor\right) + f\left(\left\lfloor \frac{a_j^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rfloor + 1\right) \end{aligned}$$

and

$$\begin{aligned} r_k^{\tilde{\boldsymbol{x}}'} f\left(\left\lceil \frac{a_k^{\tilde{\boldsymbol{x}}'}}{|W_k|} \right\rceil\right) + (|W_k| - r_k^{\tilde{\boldsymbol{x}}'}) f\left(\left\lfloor \frac{a_k^{\tilde{\boldsymbol{x}}'}}{|W_k|} \right\rfloor\right) \\ = (r_k^{\tilde{\boldsymbol{x}}} - 1) f\left(\left\lceil \frac{a_k^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rceil\right) + (|W_k| - r_k^{\tilde{\boldsymbol{x}}} - 1) f\left(\left\lfloor \frac{a_k^{\tilde{\boldsymbol{x}}}}{|W_k|} \right\rfloor\right) + f\left(\left\lceil \frac{a_k^{\tilde{\boldsymbol{x}}}}{|W_k|} \right\rceil - 1\right). \end{aligned}$$

Since we have

$$\frac{a_j}{|W_j|} < \frac{d(U_j)}{|W_j|} \leqslant \frac{d(U_k)}{|W_k|} \leqslant \frac{a_k}{|W_k|},$$

the convexity of f implies that

$$f\left(\left\lfloor \frac{a_j^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rfloor + 1\right) + f\left(\left\lceil \frac{a_k^{\tilde{\boldsymbol{x}}}}{|W_k|} \right\rceil - 1\right) \leqslant f\left(\left\lfloor \frac{a_j^{\tilde{\boldsymbol{x}}}}{|W_j|} \right\rfloor\right) + f\left(\left\lfloor \frac{a_k^{\tilde{\boldsymbol{x}}}}{|W_k|} \right\rfloor\right), \quad (5.11)$$

which shows that $\tilde{\boldsymbol{x}}'$ is also an element of $D^{\mathbb{Z}_{\geqslant 0}}$ minimizing $\sum_{i=1}^{s} r_i^{\boldsymbol{x}} f\left(\left\lceil \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rceil\right) + (|W_i| - r_i^{\boldsymbol{x}}) f\left(\left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|} \right\rfloor\right)$.

But this contradicts the fact the minimality assumption on $\tilde{\boldsymbol{x}}$. Thus, $a_i^{\tilde{\boldsymbol{x}}} = d(U_i)$ for all $i \in [s]$ and Equation (5.10) is therefore satisfied for all elements $\boldsymbol{x}$ in $D^{\mathbb{Z}_{\geqslant 0}}$.

Consider $\boldsymbol{x}^*$ satisfying Items a) and b) for every $i \in [s]$. We get the first part of the lemma by noticing that, for every $\boldsymbol{x} \in D^{\mathbb{Z}_{\geqslant 0}}$, we have

$$\sum_{w\in W} f\Big(\sum_{e\in\delta(w)} x_e\Big) \geqslant \sum_{i=1}^{s} r_i^{\boldsymbol{x}} f\left(\left\lceil \frac{a_i^{\boldsymbol{x}}}{|W_i|}\right\rceil\right) + (|W_i| - r_i^{\boldsymbol{x}}) f\left(\left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|}\right\rfloor\right)$$
$$\geqslant \sum_{i=1}^{s} \rho_i f\left(\left\lceil \frac{d(U_i)}{|W_i|}\right\rceil\right) + (|W_i| - \rho_i) f\left(\left\lfloor \frac{d(U_i)}{|W_i|}\right\rfloor\right) = \sum_{w\in W} f\Big(\sum_{e\in\delta(w)} x_e^*\Big) \quad (5.12)$$

(the first inequality is a consequence of Lemma 34), and thus $\boldsymbol{x}^*$ is an optimal solution of Problem (Min-sum$^{f,\mathbb{Z}_{\geqslant 0}}$).

The existence of an optimal solution is obtained by a direct rounding flow argument applied on $\tilde{\boldsymbol{x}}$.

To finish the proof, consider now the case when f is strictly convex. We prove first that the inequality in Equation (5.10) is strict if there is a i for which $\sum_{e\in\delta(W_i)} x_e \neq d(U_i)$. Let again $\tilde{\boldsymbol{x}} \in D^{\mathbb{Z}_{\geqslant 0}}$ minimize $\sum_{i=1}^{s} r_i^{\boldsymbol{x}} f\left(\left\lceil \frac{a_i^{\boldsymbol{x}}}{|W_i|}\right\rceil\right) + (|W_i| - r_i^{\boldsymbol{x}}) f\left(\left\lfloor \frac{a_i^{\boldsymbol{x}}}{|W_i|}\right\rfloor\right)$. This time, do not make any other minimality assumption on $\tilde{\boldsymbol{x}}$. Define $\tilde{\boldsymbol{x}}'$ as before. We see that, in this case, Equation (5.8) is strict. Therefore, when f is strictly convex, $\tilde{\boldsymbol{x}}$ necessarily satisfies $\sum_{e\in\delta(W_i)} \tilde{x}_e = d(U_i)$, and thus, if there is a i for which $\sum_{e\in\delta(W_i)} x_e \neq d(U_i)$, then the inequality in Equation (5.10) is strict. The conclusion is then immediate: let $\boldsymbol{x}^*$ be any optimal solution of Problem (Min-sum$^{f,\mathbb{Z}_{\geqslant 0}}$); consider Equation (5.12) for $\boldsymbol{x} = \boldsymbol{x}^*$; all inequalities are then equalities; strict convexity implies that $\sum_{e\in\delta(w)} x_e^* \in \left\{\left\lfloor \frac{\sum_{e\in\delta(W_i)} x_e^*}{|W_i|}\right\rfloor, \left\lceil \frac{\sum_{e\in\delta(W_i)} x_e^*}{|W_i|}\right\rceil\right\}$ (from the first inequality in Equation (5.12), which can be in this case decomposed along the i's, and with the second part of Lemma 34), and thus that $\sum_{e\in\delta(w)} x_e^* \in \left\{\left\lfloor \frac{d(U_i)}{|W_i|}\right\rfloor, \left\lceil \frac{d(U_i)}{|W_i|}\right\rceil\right\}$ □

5.2.4.3 Proofs

Proof of Theorem 30. It is a direct consequence of Lemma 33 (when $X = \mathbb{R}_{\geqslant 0}$) and Lemma 35 (when $X = \mathbb{Z}_{\geqslant 0}$). □

Proof of Proposition 31.

Case when $X = \mathbb{R}_{\geqslant 0}$. Lemma 28 implies that the optimal value of Problems (Min-max$^{\mathbb{R}_{\geqslant 0}}$) and (Max-min$^{\mathbb{R}_{\geqslant 0}}$) are respectively $\frac{d(U_1)}{|W_1|}$ and $\frac{d(U_s)}{|W_s|}$. It implies that the optimal value of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) is $\frac{d(U_1)}{|W_1|} - \frac{d(U_s)}{|W_s|}$. Lemma 33 shows that any optimal solution of Problem (Min-sum$^{f,\mathbb{R}_{\geqslant 0}}$) with f being strictly convex gives to the objective function of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) a value equal to $\frac{d(U_1)}{|W_1|} - \frac{d(U_s)}{|W_s|}$, which is the desired conclusion.

Case when $X = \mathbb{Z}_{\geqslant 0}$. Lemma 28 implies that the optimal value of Problems (Min-max$^{\mathbb{Z}_{\geqslant 0}}$) and (Max-min$^{\mathbb{Z}_{\geqslant 0}}$) are respectively $\left\lceil \frac{d(U_1)}{|W_1|}\right\rceil$ and $\left\lfloor \frac{d(U_s)}{|W_s|}\right\rfloor$. It implies that the optimal value of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) is $\left\lceil \frac{d(U_1)}{|W_1|}\right\rceil - \left\lfloor \frac{d(U_s)}{|W_s|}\right\rfloor$. Lemma 33 shows that any optimal solution of Problem (Min-sum$^{f,\mathbb{Z}_{\geqslant 0}}$) with f being strictly convex gives to the objective function of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) a value equal to $\left\lceil \frac{d(U_1)}{|W_1|}\right\rceil - \left\lfloor \frac{d(U_s)}{|W_s|}\right\rfloor$, which is the desired conclusion. □

Proof of Theorem 32. A partition $U_1, \dots, U_s$ as in Lemma 28 can be found in strongly polynomial time by following its proof: first, solve Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$) with Tardos's algorithm; second, greedily remove vertices from $W^{\max}(\boldsymbol{y})$, as in the proof, until the set U_1 is obtained; then recurse.

Lemma 33 (when $X = \mathbb{R}_{\geqslant 0}$) and Lemma 35 (when $X = \mathbb{Z}_{\geqslant 0}$) show that, once we have $U_1, \dots, U_s$, a simple flow algorithm can be used to find an optimal solution. □

5.2.5 Algorithms

The flow construction in the proof of Theorem 26 (in the case $X = \mathbb{Z}_{\geqslant 0}$) in Section 5.2.2 was used to get an integer solution from a continuous one. This "rounding" procedure can be performed efficiently by many flow algorithms, and this latter remark is essentially the proof of Proposition 27.

Proof of Proposition 27. An optimal solution $\boldsymbol{x}^*$ to Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) can be computed in strongly polynomial time by linear programming. With the help of $\boldsymbol{x}^*$, we build then a flow instance as in Section 5.2.2. An integer solution $\boldsymbol{z}^*$ as defined in that section can be computed in strongly polynomial time, by any efficient flow algorithm, e.g., the Edmonds–Karp algorithm. This solution $\boldsymbol{z}^*$ is an optimal solution of Problem (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) by Proposition 29. □

Remark 13. Notice that Theorem 32 provides an alternative proof of Proposition 27. Indeed, by Theorem 32, we have a strongly polynomial algorithm to generate a solution of Problem (Min-sum^{f,X}). Proposition 31 states that the generated optimal solution is also optimal for Problem (Min-diff^{X}), which can therefore be generated in strongly polynomial time.

5.3 Min-max relations

When $X = \mathbb{R}_{\geqslant 0}$, Problems (Min-max$^{\mathbb{R}_{\geqslant 0}}$), (Max-min$^{\mathbb{R}_{\geqslant 0}}$), and (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) enjoy min-max relations with problems of combinatorial nature. For a subset V' of V, we denote by $N(V')$ the set of vertices that are not in V' but have at least one neighbor in V'. For a subset U' of U, we denote by $d(U')$ the value of $\sum_{u \in U'} d(u)$.

Theorem 36. *When Problems* (Min-max$^{\mathbb{R}_{\geqslant 0}}$), (Max-min$^{\mathbb{R}_{\geqslant 0}}$), *and* (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) *are feasible, their optimal values are respectively equal to*

$$\max_{\substack{U' \subseteq U \\ U' \neq \varnothing}} \frac{d(U')}{|N(U')|}, \qquad \min_{\substack{W' \subseteq W \\ W' \neq \varnothing}} \frac{d(N(W'))}{|W'|}, \qquad \textit{and} \qquad \max_{\substack{U' \subseteq U, W' \subseteq W \\ U' \neq \varnothing, W' \neq \varnothing}} \frac{d(U')}{|N(U')|} - \frac{d(N(W'))}{|W'|}.$$

This theorem has a counterpart for $X = \mathbb{Z}_{\geqslant 0}$. It is exactly the same statement, except that the fractions are rounded.

Theorem 37. *When Problems* (Min-max$^{\mathbb{Z}_{\geqslant 0}}$), (Max-min$^{\mathbb{Z}_{\geqslant 0}}$), *and* (Min-diff$^{\mathbb{Z}_{\geqslant 0}}$) *are feasible, their optimal values are respectively equal to*

$$\max_{\substack{U' \subseteq U \\ U' \neq \varnothing}} \left\lceil \frac{d(U')}{|N(U')|} \right\rceil, \qquad \min_{\substack{W' \subseteq W \\ W' \neq \varnothing}} \left\lfloor \frac{d(N(W'))}{|W'|} \right\rfloor, \qquad \textit{and} \qquad \max_{\substack{U' \subseteq U, W' \subseteq W \\ U' \neq \varnothing, W' \neq \varnothing}} \left\lceil \frac{d(U')}{|N(U')|} \right\rceil - \left\lfloor \frac{d(N(W'))}{|W'|} \right\rfloor.$$

The proof of this theorem is immediate from the application of Theorem 36 and Proposition 29. We therefore concentrate on the proof of Theorem 36, where the following lemma is useful (notice that this lemma concerns all possible subsets of U and W).

Lemma 38. *Consider any feasible solution* $\boldsymbol{x}$ *of Problem* (Min-diff^X) *and any nonempty subsets* $U' \subseteq U$ *and* $W' \subseteq W$. *Then the following inequalities hold:*

$$\frac{d(U')}{|N(U')|} \leqslant \ell^{\max}(\boldsymbol{x}) \qquad \textit{and} \qquad \frac{d(N(W'))}{|W'|} \geqslant \ell^{\min}(\boldsymbol{x}).$$

Proof. We have $\sum_{w\in N(U')}\sum_{e\in\delta(w)} x_e \geqslant \sum_{u\in U'}\sum_{e\in\delta(u)} x_e = d(U')$. Since in a collection of numbers there is always one number non-smaller than the average, there is at least one w in $N(U')$ for which $\sum_{e\in\delta(w)} x_e$ is non-smaller than $\frac{d(U')}{|N(U')|}$, which proves the first part of the statement.

We also have $\sum_{w\in W'}\sum_{e\in\delta(w)} x_e \leqslant \sum_{u\in N(W')}\sum_{e\in\delta(u)} x_e = d(N(W'))$. Since in a collection of numbers there is always one number non-larger than the average, there is at least one w in W' for which $\sum_{e\in\delta(w)} x_e$ is non-larger than $\frac{d(N(W'))}{|W'|}$, which proves the second part of the statement. □

Proof of Theorem 36. Consider now any optimal solution $\boldsymbol{x}^*$ of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$). By Remark 12, the optimal value of Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$) is equal to $\ell^{\max}(\boldsymbol{x}^*)$, which is equal to $\frac{d(U_1)}{|N(U_1)|}$ for some nonempty subset U_1 of U. Lemma 38 shows then the min-max relation stated by Theorem 36 for Problem (Min-max$^{\mathbb{R}_{\geqslant 0}}$). The similar relation for Problem (Max-min$^{\mathbb{R}_{\geqslant 0}}$) is proved the same way.

Lemma 38 shows that $\ell^{\max}(\boldsymbol{x}^*) - \ell^{\min}(\boldsymbol{x}^*)$ is at least the quantity $\frac{d(U')}{|N(U')|} - \frac{d(N(W'))}{|W'|}$ for any pair of nonempty subsets U' and W' of U and W respectively. On the other hand, the difference $\ell^{\max}(\boldsymbol{x}^*) - \ell^{\min}(\boldsymbol{x}^*)$ is by definition the optimal value of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$), and is equal to $\frac{d(U^*)}{|N(U^*)|} - \frac{d(N(W^*))}{|W^*|}$, which shows the min-max relation stated by Theorem 36 for Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$). □

Remark 14. The three combinatorial problems stated in Theorem 36 enjoy a similar property as the one stated in Theorem 26: any optimal solution (U^*, W^*) of the right-hand side problem is such that U^* and W^* are optimal solutions of the left-hand side and middle problems. However, the proof of the property here is immediate since the right-hand side problem can be split independently in the two other problems.

5.4 A generalization and variations when $X = \mathbb{R}_{\geqslant 0}$

5.4.1 Statement

In this section, we consider a version of the problems where the form of the objective and the constraints is relaxed. We keep the same graph G and the same demand function d, but there are now maps $f_v\colon \mathbb{R}_{\geqslant 0}^{\delta(v)} \to \mathbb{R}$ attached to the vertices v of G.

Given a vector $\boldsymbol{x}$ and a subset A of its indices, we denote by $\boldsymbol{x}_A$ the vector $(x_i)_{i\in A}$. Here are the three problems.

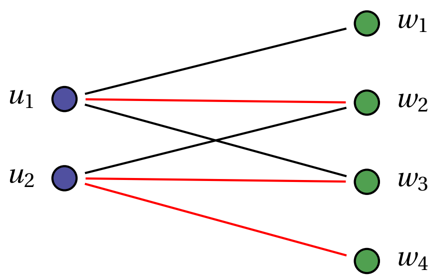

Figure 5.2 – Example that shows the gap must be nonzero in Theorem 39.

$$\begin{array}{lll} \text{Minimize} & \max\limits_{w\in W} f_w(\boldsymbol{x}_{\delta(w)}) & \\ \text{subject to} & f_u(\boldsymbol{x}_{\delta(u)}) = d(u) & \forall u \in U \\ & x_e \in \mathbb{R}_{\geqslant 0} & \forall e \in E\,, \end{array} \tag{Min-max^f}$$

$$\begin{array}{lll} \text{Maximize} & \min\limits_{w\in W} f_w(\boldsymbol{x}_{\delta(w)}) & \\ \text{subject to} & f_u(\boldsymbol{x}_{\delta(u)}) = d(u) & \forall u \in U \\ & x_e \in \mathbb{R}_{\geqslant 0} & \forall e \in E\,, \end{array} \tag{Max-min^f}$$

$$\begin{array}{lll} \text{Minimize} & \max\limits_{w,w'\in W} \left(f_w(\boldsymbol{x}_{\delta(w)}) - f_{w'}(\boldsymbol{x}_{\delta(w')})\right) & \\ \text{subject to} & f_u(\boldsymbol{x}_{\delta(u)}) = d(u) & \forall u \in U \\ & x_e \in \mathbb{R}_{\geqslant 0} & \forall e \in E\,. \end{array} \tag{Min-diff^f}$$

We have a result similar to the one stated in Theorem 26, which actually generalizes this latter theorem when $X = \mathbb{R}_{\geqslant 0}$ and the optimal value of Problem (Min-diff$^{\mathbb{R}_{\geqslant 0}}$) is nonzero. For any set A, we denote by $\mathcal{F}_A$ the set of continuous functions $f\colon \mathbb{R}^A_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ such that

- $f(\boldsymbol{x}) = 0$ if $x_i = 0$ for some $i \in A$, and
- $x \in \mathbb{R}_{\geqslant 0} \mapsto f(x, \boldsymbol{y}_{-i}) \in \mathbb{R}_{\geqslant 0}$ is bijective (and thus increasing) for every $i \in A$ and every $\boldsymbol{y} \in \mathbb{R}^A_{>0}$.

Here $(x, \boldsymbol{y}_{-i})$ is the element of $\mathbb{R}^A_{\geqslant 0}$ whose i-th component is x and whose j-th component for $j \neq i$ is y_j.

Theorem 39. *Suppose that either every f_v belongs to $\mathcal{F}_{\delta(v)}$, or every f_v writes as $\boldsymbol{x}_{\delta(v)} \mapsto \boldsymbol{a}^v \cdot \boldsymbol{x}_{\delta(v)}$ for some vector $\boldsymbol{a}^v \in \mathbb{R}^{\delta(v)}_{>0}$. If the optimal value of Problem* (Min-diff^f) *is nonzero, then every optimal solution of Problem* (Min-diff^f) *is simultaneously optimal for Problems* (Min-max^f) *and* (Max-min^f).

The proof of Theorem 39 shares similarities with that of Theorem 26 for $X = \mathbb{R}_{\geqslant 0}$, but it requires nontrivial preliminary results. These preliminary results and the proof are given in hereafter. Before that, we state an example showing that when the gap is zero, the conclusion of the theorem does not necessarily hold.

Consider the graph of Figure 5.2. Set $d(u_1) = \frac{21}{20}$ and $d(u_2) = \frac{7}{10}$, and $f_u(\boldsymbol{x}_{\delta(u)}) = \sum_{w\in\delta(u)} \boldsymbol{x}_{uw}$ for

$u \in \{u_1, u_2\}$ and $f_w(\boldsymbol{x}_{\delta(w)}) = \boldsymbol{a}^w \cdot \boldsymbol{x}_{\delta(w)}$ with

$$a_e^w = \begin{cases} 10 & \text{if } e \text{ is in red on the figure,} \\ 1 & \text{if } e \text{ is in black on the figure} \end{cases}$$

for $e \in \delta(w)$ and $w \in \{w_1, w_2, w_3, w_4\}$. The vector

$$x^*_{u_1w_1} = 1, \quad x^*_{u_1w_2} = \frac{1}{20}, \quad x^*_{u_1w_3} = 0, \quad x^*_{u_2w_2} = \frac{1}{2}, \quad x^*_{u_2w_3} = \frac{1}{10}, \quad x^*_{u_2w_4} = \frac{1}{10}$$

gives a value 0 to the objective function of Problem (Min-diff^f) and a maximum load of 1. It is thus an optimal solution of this latter problem. Yet, the feasible solution

$$x_{u_1w_1} = \frac{11}{20}, \quad x_{u_1w_2} = 0, \quad x_{u_1w_3} = \frac{1}{2}, \quad x_{u_2w_2} = \frac{7}{10}, \quad x_{u_2w_3} = 0, \quad x_{u_2w_4} = 0$$

gives a maximum load of $\frac{7}{10}$, which shows that $\boldsymbol{x}^*$ is not optimal for Problem (Min-max^f).

5.4.2 Preliminaries

We provide now a few preliminary results for the proof of Theorem 39, which are actually used only for the case when the f_v belong to $\mathcal{F}_{\delta(v)}$.

Lemma 40. *Let $k < n$ be nonnegative integers and A a subset of $[k]$ (this latter being empty if $k = 0$). Consider a map $g \in \mathcal{F}_{A\cup\{k+1\}}$. Suppose given a positive real number r and a continuous map $\boldsymbol{z}(\cdot)\colon [0,1] \to \mathbb{R}^n$ such that $z_i(t) \neq 0$ for all $t \in [0,1]$. Then there exists a continuous map $\phi\colon [0,1] \to \mathbb{R}$ such that $g(\boldsymbol{z}_A(t), \phi(t)) = r$ for all $t \in [0,1]$.*

Proof. Define $\phi(t)$ to be the unique x such that $g(\boldsymbol{z}_A(t), x) = r$. Such an x exists and is unique since by assumption $x \mapsto g(\boldsymbol{z}_A(t), x)$ is bijective.

The proof consists then in showing that such a ϕ is continuous. Consider a sequence of real number $(t_i)_{i\in\mathbb{Z}_{\geqslant 0}}$ converging to some $\bar{t}$. Let $\underline{z}_j$ (resp. $\overline{z}_j$) be the infimum (resp. supremum) of the $z_j(t_i)$. It is finite by continuity of $\boldsymbol{z}(\cdot)$. There exists $\underline{\ell} \in \mathbb{R}$ (resp. $\overline{\ell} \in \mathbb{R}$) such that $g(\overline{\boldsymbol{z}}, \underline{\ell}) = r$ (resp. $g(\underline{\boldsymbol{z}}, \overline{\ell}) = r$), again by the assumption on g. The quantity $\phi(t_i)$ belongs to $[\underline{\ell}, \overline{\ell}]$ for all $i \in \mathbb{Z}_{\geqslant 0}$. The sequence $\big(\phi(t_i)\big)$ admits limit points. Let $\tilde{\phi}$ be one of these limit points. By continuity of g, we have $g(\boldsymbol{z}_A(\bar{t}), \tilde{\phi}) = r$. Since $x \mapsto g(\boldsymbol{z}_A(\bar{t}), x)$ is a bijection, we have $\phi(\bar{t}) = \tilde{\phi}$, which means that the sequence $\big(\phi(t_i)\big)$ is actually converging to $\phi(\bar{t})$. □

Lemma 41. *Let $A_1, \ldots, A_q$ be subsets of $[n]$ and $g_1, \ldots, g_q$ be maps such that for every $j \in [q]$ the set $A_j \setminus (A_1 \cup \cdots \cup A_{j-1})$ is nonempty and $g_j \in \mathcal{F}_{A_j}$. Then, for every $\boldsymbol{r} \in \mathbb{R}^q_{>0}$, the set $\{\boldsymbol{x} \in \mathbb{R}^n\colon g_i(\boldsymbol{x}_{A_i}) = r_i \ \forall i \in [q]\}$ is nonempty and path-connected.*

Proof. The nonemptyness is obvious by definition of the $\mathcal{F}_{A_j}$. We prove the path-connectivity by induction on q. Let S_j be the set $\{\boldsymbol{x} \in \mathbb{R}^n\colon g_i(\boldsymbol{x}_{A_i}) = r_i \ \forall i \in [j]\}$.

Consider the case $q = 1$. Let $\boldsymbol{y}, \boldsymbol{y}'$ be two elements of S_1. They have no zero components. Choose any continuous map $\boldsymbol{z}(\cdot)\colon [0,1] \to \mathbb{R}^n$ such that $\boldsymbol{z}(0) = \boldsymbol{y}$ and $\boldsymbol{z}(1) = \boldsymbol{y}'$, and such that $z_i(t) \neq 0$ for all i and all $t \in [0,1]$. Lemma 40 (with $k = |A_1| - 1$) shows that there exists $\phi\colon [0,1] \to \mathbb{R}$ such that $g_1(\boldsymbol{z}_{A_1\setminus\{\max A_1\}}(t),\phi(t)) = r$ for all $t \in [0,1]$. Set $h_i(t) = z_i(t)$ for $i \neq \max A_1$ and $h_i(t) = \phi(t)$ for $i = \max A_1$. The map $\boldsymbol{h}\colon [0,1] \to \mathbb{R}^n$ is continuous and such that $\boldsymbol{h}(t) \in S_1$ for all $t \in [0,1]$. By the assumption on g_1, we necessarily have $\boldsymbol{h}(0) = \boldsymbol{y}$ and $\boldsymbol{h}(1) = \boldsymbol{y}'$. This ends the proof of the case $q = 1$.

Consider now the case $q > 1$. Let $\boldsymbol{y},\boldsymbol{y}'$ be two elements of S_q. By induction, there exists a continuous map $\boldsymbol{z}\colon [0,1] \to \mathbb{R}^n$ such that $\boldsymbol{z}(0) = \boldsymbol{y}$, $\boldsymbol{z}(1) = \boldsymbol{y}'$, and $\boldsymbol{z}(t) \in S_{q-1}$ for all $t \in [0,1]$. Lemma 40 (with $k = |A_1 \cup \cdots \cup A_q| - 1$) shows that there exists $\phi\colon [0,1] \to \mathbb{R}$ such that $g_q(\boldsymbol{z}_{A_q\setminus\{\max A_q\}}(t),\phi(t)) = r_q$ for all $t \in [0,1]$. Set $h_i(t) = z_i(t)$ for $i \neq \max A_q$ and $h_i(t) = \phi(t)$ for $i = \max A_q$. The map $\boldsymbol{h}\colon [0,1] \to \mathbb{R}^n$ is continuous and such that $\boldsymbol{h}(t) \in S_q$ for all $t \in [0,1]$. By the assumption on g_q, we necessarily have $\boldsymbol{h}(0) = \boldsymbol{y}$ and $\boldsymbol{h}(1) = \boldsymbol{y}'$. This ends the proof of the case $q > 1$. □

Let $D_f = \{\boldsymbol{x} \in \mathbb{R}^E_{\geqslant 0}\colon f_u(\boldsymbol{x}_{\delta(u)}) = d(u)\ \forall u \in U\}$. This is the set of feasible solutions of either of Problems (Max-min^f), (Min-max^f), and (Min-diff^f).

Lemma 42. *Let $\varepsilon > 0$. Suppose that f_v belongs to $\mathcal{F}_{\delta(v)}$ for all v. Consider two vectors $\boldsymbol{y},\boldsymbol{y}'$ in D_f. If there exists $w \in W$ such that $f_w(\boldsymbol{y}'_{\delta(w)}) \neq f_w(\boldsymbol{y}_{\delta(w)})$, then there exists $\boldsymbol{y}'' \in D_f$ such that*

- $f_w(\boldsymbol{y}_{\delta(w)}) - \varepsilon < f_w(\boldsymbol{y}''_{\delta(w)}) \leqslant f_w(\boldsymbol{y}_{\delta(w)})$ *for $w \in W$ with $f_w(\boldsymbol{y}'_{\delta(w)}) < f_w(\boldsymbol{y}_{\delta(w)})$.*
- $f_w(\boldsymbol{y}_{\delta(w)}) + \varepsilon > f_w(\boldsymbol{y}''_{\delta(w)}) \geqslant f_w(\boldsymbol{y}_{\delta(w)})$ *for $w \in W$ with $f_w(\boldsymbol{y}'_{\delta(w)}) > f_w(\boldsymbol{y}_{\delta(w)})$.*
- $f_w(\boldsymbol{y}''_{\delta(w)}) = f_w(\boldsymbol{y}_{\delta(w)})$ *for $w \in W$ with $f_w(\boldsymbol{y}'_{\delta(w)}) = f_w(\boldsymbol{y}_{\delta(w)})$.*
- $f_w(\boldsymbol{y}''_{\delta(w)}) \neq f_w(\boldsymbol{y}_{\delta(w)})$ *for at least one $w \in W$.*

Proof. Assume without loss of generality that G is connected.

The proof works by induction on the number of vertices w such that $f_w(\boldsymbol{y}'_{\delta(w)}) \neq f_w(\boldsymbol{y}_{\delta(w)})$. If this number is zero, then we are done (an implication is always true if its precedent is false). So, denote by W' the set $\{w \in W\colon f_w(\boldsymbol{y}'_{\delta(w)}) = f_w(\boldsymbol{y}_{\delta(w)})\}$ and suppose that $|W'| < |W|$.

There exists an ordering $v_1, v_2, \ldots$ of the vertices in V such that $\delta(v_j) \setminus (\delta(v_1) \cup \cdots \cup \delta(v_{j-1}))$ is nonempty for every v_j, except the last one. Indeed, consider a spanning tree T of G and remove one after the other the leaves of T. Note that we can even choose the last vertex of this ordering since a tree with more than one vertex has always at least two leaves. So, we fix the ordering so that the last vertex is in $W \setminus W'$. Then, by restricting this ordering to the vertices in $U \cup W'$, Lemma 41 implies that the set $L = \{\boldsymbol{x} \in D_f\colon f_w(\boldsymbol{x}_{\delta(w)}) = f_w(\boldsymbol{y}_{\delta(w)})\ \forall w \in W'\}$ is path-connected. Let $\boldsymbol{z}(\cdot)\colon [0,1] \to L$ be a continuous map such that $\boldsymbol{z}(0) = \boldsymbol{y}$ and $\boldsymbol{z}(1) = \boldsymbol{y}'$. Define t^* as the infimum of the t's such that

$$\left(f_w(\boldsymbol{z}_{\delta(w)}(t)) - f_w(\boldsymbol{y}_{\delta(w)})\right)\left(f_w(\boldsymbol{y}'_{\delta(w)}) - f_w(\boldsymbol{y}_{\delta(w)})\right) > 0 \quad \text{for all } w \notin W'.$$

Note that t^* exists by continuity, is smaller than 1, and that there necessarily exists $w_0 \notin W'$ such that $f_{w_0}(\boldsymbol{z}_{\delta(w_0)}(t^*)) = f_{w_0}(\boldsymbol{y}_{\delta(w_0)})$. If $f_w(\boldsymbol{z}_{\delta(w)}(t^*)) = f_w(\boldsymbol{y}_{\delta(w)})$ for all w, then defining $\boldsymbol{y}''$ as $\boldsymbol{z}(t_0)$ for a t_0 slightly larger than t^* makes the job. Hence, suppose $f_w(\boldsymbol{z}_{\delta(w)}(t^*)) \neq f_w(\boldsymbol{y}_{\delta(w)})$ for at least one w. We can apply induction on $\boldsymbol{y}, \boldsymbol{z}(t^*)$ to get $\boldsymbol{y}''$ that satisfies the required property. □

5.4.3 Proof

Consider a feasible solution $\boldsymbol{y}$ of Problem (Min-diff^f). We denote by $q^{\max}(\boldsymbol{y})$ the quantity $\max_{w\in W} f_w(\boldsymbol{y}_{\delta(w)})$ and by $q^{\min}(\boldsymbol{y})$ the quantity $\min_{w\in W} f_w(\boldsymbol{y}_{\delta(w)})$. We define $W^{\max}(\boldsymbol{y})$ (resp. $W^{\min}(\boldsymbol{y})$) to be the set of vertices w in W for which the maximum is attained in the definition of $q^{\max}$ (resp. $q^{\min}$). We define also $U^{\max}(\boldsymbol{y})$ to be the set of vertices $u \in U$ for which there exists $w \in W^{\max}(\boldsymbol{y})$ with $y_{uw} > 0$.

The following lemma played actually already a role in the chapter: it was a key claim stated and proved within the proof of Theorem 26. For sake of readability, we state it as a separate lemma.

Lemma 43. *Suppose the condition of Theorem 39 is satisfied and that $d(u) > 0$ for at least one vertex u. Consider an optimal solution $\boldsymbol{y}$ of Problem* (Min-diff^f). *If $N(U^{\max}(\boldsymbol{y})) \neq W^{\max}(\boldsymbol{y})$, then there exists an optimal solution $\boldsymbol{y}'$ such that $W^{\max}(\boldsymbol{y}') \subsetneq W^{\max}(\boldsymbol{y})$ and $q^{\max}(\boldsymbol{y}') = q^{\max}(\boldsymbol{y})$.*

Proof. Suppose that $N(U^{\max}(\boldsymbol{y})) \neq W^{\max}(\boldsymbol{y})$. Since $W^{\max}(\boldsymbol{y}) \subseteq N(U^{\max}(\boldsymbol{y}))$ by definition ($d(u) > 0$ for at least one u), there exists $u' \in U^{\max}(\boldsymbol{y})$ and $w' \in W \setminus W^{\max}(\boldsymbol{y})$ such that $u'w' \in E$. Still by definition, there exists $w'' \in W^{\max}(\boldsymbol{y})$ such that $y_{u'w''} > 0$. In any case, there exists $\boldsymbol{y}' \in \mathbb{R}^E_{\geqslant 0}$ with

$$y'_e = y_e \text{ if } e \notin \{u'w'', u'w'\}, \quad y'_e < y_e \text{ if } e = u'w'', \quad \text{and} \quad y'_e \geqslant y_e \text{ if } e = u'w'$$

such that $f_u(\boldsymbol{y}'_{\delta(u)}) = f_u(\boldsymbol{y}_{\delta(u)})$ for every $u \in U$. (The last inequality is not strict because $d(u')$ might be equal to 0.)

The vector $\boldsymbol{y}'$ is thus still a feasible solution of Problem (Min-diff^f). The way $\boldsymbol{y}'$ is built implies that $q^{\max}(\boldsymbol{y}') \leqslant q^{\max}(\boldsymbol{y})$ and $q^{\min}(\boldsymbol{y}') \geqslant q^{\min}(\boldsymbol{y})$. Since $\boldsymbol{y}$ is an optimal solution of Problem (Min-diff^f), we have $q^{\max}(\boldsymbol{y}') = q^{\max}(\boldsymbol{y})$ and $q^{\min}(\boldsymbol{y}') = q^{\min}(\boldsymbol{y})$. The claim follows then from the fact that $W^{\max}(\boldsymbol{y}')$ is a strict subset of $W^{\max}(\boldsymbol{y})$: the optimal value being nonzero, we have in any case $f_{w''}(\boldsymbol{y}_{\delta(w'')}) > 0$ and thus $y'_{u'w''} < y_{u'w''}$ implies that $f_{w''}(\boldsymbol{y}'_{\delta(w'')}) < f_w(\boldsymbol{y}_{\delta(w'')})$. □

Proof of Theorem 39. Assume that Problem (Min-diff^f) admits feasible solutions and that $d(u) > 0$ for at least one vertex u (otherwise there is nothing to prove). We prove that all optimal solutions of Problem (Min-diff^f) are optimal solutions of Problem (Min-max^f). The proof that they are also optimal solutions of Problem (Max-min^f) is omitted since it follows exactly the same lines.

Consider any optimal solution $\boldsymbol{x}^*$ of Problem (Min-diff^f). By finiteness, Lemma 43 implies the existence of an optimal solution $\bar{\boldsymbol{x}}$ such that $N(U^{\max}(\bar{\boldsymbol{x}})) = W^{\max}(\bar{\boldsymbol{x}})$ and $q^{\max}(\bar{\boldsymbol{x}}) = q^{\max}(\boldsymbol{x}^*)$. Choose such an optimal solution such that $W^{\max}(\bar{\boldsymbol{x}})$ is as small as possible. Since the optimal value of Problem (Min-diff^f) is nonzero, the sets $W^{\max}(\bar{\boldsymbol{x}})$ and $W^{\min}(\bar{\boldsymbol{x}})$ are disjoint. Consider also an optimal solution $\tilde{\boldsymbol{x}}$ of Problem (Max-min^f).

Case when $f_v \in \mathcal{F}_{\delta(v)}$ for all v. Define the following solution:

$$z_e = \begin{cases} \tilde{x}_e & \text{if } e \in \delta(U^{\max}(\bar{\boldsymbol{x}})), \\ \bar{x}_e & \text{if } e \notin \delta(U^{\max}(\bar{\boldsymbol{x}})). \end{cases}$$

The vector $\boldsymbol{z}$ is a feasible solution of Problem (Min-diff^f). For $w \notin W^{\max}(\bar{\boldsymbol{x}})$, we have $f_{\delta(w)}(\boldsymbol{z}_{\delta(w)}) = f_{\delta(w)}(\bar{\boldsymbol{x}}_{\delta(w)})$ since there is no edge between $U^{\max}(\bar{\boldsymbol{x}})$ and $W \setminus W^{\max}(\bar{\boldsymbol{x}})$. The solution $\bar{\boldsymbol{x}}$ is also an optimal solution of Problem (Min-max^f) since otherwise Lemma 42 with $\boldsymbol{y} = \bar{\boldsymbol{x}}$ and $\boldsymbol{y}' = \boldsymbol{z}$ would then either contradict the optimality of $\bar{\boldsymbol{x}}$ for Problem (Min-diff^f) or the minimality of $W^{\max}(\bar{\boldsymbol{x}})$.

Case when f_v writes as $\boldsymbol{x}_{\delta(v)} \mapsto \boldsymbol{a}^v \cdot \boldsymbol{x}_{\delta(v)}$ for some vector $\boldsymbol{a}^v \in \mathbb{R}^{\delta(v)}_{>0}$. Define the following family of solutions:

$$z_e(t) = \begin{cases} t\tilde{x}_e + (1-t)\bar{x}_e & \text{if } e \in \delta(U^{\max}(\bar{\boldsymbol{x}})), \\ \bar{x}_e & \text{if } e \notin \delta(U^{\max}(\bar{\boldsymbol{x}})). \end{cases}$$

The vector $\boldsymbol{z}(t)$ is a feasible solution of Problem (Min-diff^f) for every $t \in [0,1]$. For $w \notin W^{\max}(\bar{\boldsymbol{x}})$, we have $\boldsymbol{a}^w \cdot \boldsymbol{z}_{\delta(w)}(t) = \boldsymbol{a}^w \cdot \bar{\boldsymbol{x}}_{\delta(w)}$ for every $t \in [0,1]$ since there is no edge between $U^{\max}(\bar{\boldsymbol{x}})$ and $W \setminus W^{\max}(\bar{\boldsymbol{x}})$. For $w \in W^{\max}(\bar{\boldsymbol{x}})$, we have

$$\boldsymbol{a}^w \cdot \boldsymbol{z}_{\delta(w)}(0) = \boldsymbol{a}^w \cdot \bar{\boldsymbol{x}}_{\delta(w)} \quad \text{and} \quad \boldsymbol{a}^w \cdot \boldsymbol{z}_{\delta(w)}(1) \leqslant \boldsymbol{a}^w \cdot \tilde{\boldsymbol{x}}_{\delta(w)}\,.$$

The second relation holds because $\tilde{x}_e = 0$ for $e \in \delta(W^{\max}(\bar{\boldsymbol{x}})) \setminus \delta(U^{\max}(\bar{\boldsymbol{x}}))$. It implies that $q^{\max}(\boldsymbol{z}(t_0)) \leqslant q^{\max}(\bar{\boldsymbol{x}})$ and $q^{\min}(\boldsymbol{z}(t_0)) = q^{\min}(\bar{\boldsymbol{x}})$ for some small enough $t_0 > 0$. By the optimality of $\bar{\boldsymbol{x}}$ for Problem (Min-diff^f), we have thus $q^{\max}(\boldsymbol{z}(t_0)) = q^{\max}(\bar{\boldsymbol{x}})$. The equality $q^{\max}(\boldsymbol{z}(t_0)) = q^{\max}(\bar{\boldsymbol{x}})$ implies (by linearity) that $\boldsymbol{a}^w \cdot \boldsymbol{z}_{\delta(w)}(1) = q^{\max}(\bar{\boldsymbol{x}})$ for at least one vertex $w \in W^{\max}(\bar{\boldsymbol{x}})$. Hence, $q^{\max}(\tilde{\boldsymbol{x}}) = q^{\max}(\bar{\boldsymbol{x}})$. The vector $\bar{\boldsymbol{x}}$ is therefore also optimal for Problem (Min-max^f), which in turn implies that the vector $\boldsymbol{x}^*$ is optimal for Problem (Min-max^f) as well. $\square$

Conclusion

As a conclusion, we summarize the main contributions of this thesis and give some possible future research directions. This thesis has been led in collaboration with Air France. This collaboration has motivated the research topic explored in Chapters 2, 3 and 4.

Chapter 2 focuses on a version of the aircraft routing problem with the introduction of constraints on the number of available gates in the airports and frequency constraints. The gate constraint has become limiting for Air France in the last years, especially in the hub of the company. In the chapter, we model the problem as a MILP and explore the numerical results of this model. We also prove the problem to be NP-complete which justifies that (unless P = NP) no polynomial algorithm is able to solve the problem exactly.

In Chapter 3, we study the evaluation of the revenue generated by a given schedule. The arrival dynamic we study is based on a multinomial logit choice model. Since the expected arrival number of customers in this atomic model is hard to estimate, we approximate those expectations by a fluid model. The revenue estimation of this model can be expressed as a problem known in the literature as the Sales Based Linear Program (SBLP). We show that for a fixed capacity, the atomic model converges to the fluid model and even prove that the convergence speed is exponential. Moreover, we develop a column generation method to solve the SBLP which has a better performance than the direct linear program implementation. This column generation is particularly effective because the slave problem can be solved quickly. We even prove that there exists an $O(n)$ algorithm that finds the solution of the slave problem.

The problem we investigate in Chapter 4 gathers the two problems studied in the previous chapters. This joint problem aims at finding the best possible schedule with an itinerary-based revenue model. The size of the problem being very important, the linear program that models our problem is hard to solve quickly. We therefore use a Benders decomposition to solve this problem. Since the decomposition method is not efficient enough to generate solutions quickly, we also present advanced enhancement methods based on the column generation introduced in the Chapter 3 and on non-trivial duality interpretations.

This thesis has also been the opportunity to study other problems. In particular, Chapter 5 focuses on a load balancing problem for which three objective functions are linked: minimizing the difference between the minimum and maximum workloads of the workers turns out to also minimize the maximum workload and maximize the minimum workload. It turns out that this property remains true for more general versions of the problem.

We now present some research directions that arose during our work on the problems studied during the thesis. A first question that we have been interested in but that we could not answer concerns Conjecture 20 presented in Chapter 3. The numerical experiments we led suggest that sorting the itineraries enables to make an easy distinction between the itineraries whose expected occupation are overestimated (and underestimated) by the fluid model (compared to the atomic model). If the conjecture revealed to be false, this would leave open the existence of a concrete and easy way to discriminate between itineraries the fluid model overestimate and underestimate.

The numerical results of the Benders decomposition methods tested to solve the joint problem do not have better performances than the direct linear problem implementation. A natural extension to our work on the Benders decomposition is therefore to design a practical heuristic to solve the joint problem quickly. A natural approach would be to generate a set of routes for each plane, and to find small modifications of the schedule that improve the revenue generated. However, the revenue estimation being quite time-consuming, it seems that finding guarantees of the good quality of the solution is difficult.

A type of heuristic that could give good results are machine learning based heuristics relying on the structured learning framework [17, 21]. This framework has been used recently to approximate hard problems by easier ones (see for instance in Parmentier [103]). In our context, it could be interesting to learn a function able to transform an instance of the joint problem into an instance of the extended aircraft routing problem.

A question that remains open is to find an efficient way to generate a solution of the joint problem and to have guarantees on the quality of the solution.

Appendix Part

A Technical tools

A.1 Decomposition methods for linear programs

In this appendix, we present the technical background for understanding the Dantzig–Wolfe and the Benders decomposition methods. We also present the links between the two approaches. Since the two methods are dual from one another, we consider a classical linear program and its dual on which we respectively apply the two decomposition methods. Let c, d, f and b be the constant vectors of our problem and E, F and A be the constraint matrices. The problem we consider can be written as follows.

$$\max c^\top x + d^\top y \tag{A.1a}$$
$$\text{st:} Ex + Fy = f \tag{A.1b}$$
$$Ax = b \tag{A.1c}$$
$$x \geqslant 0, \quad y \geqslant 0 \tag{A.1d}$$

The dual of this problem then has the following from:

$$\max f^\top u + b^\top v \tag{A.2a}$$
$$\text{st:} E^\top u + A^\top v \leqslant c \tag{A.2b}$$
$$F^\top u \leqslant d \tag{A.2c}$$

The two problems have a very similar structure and the two decomposition methods could be used on both. However, for the sake of the presentation, we consider that Problem (A.1) has a complicating set of constraints (Equations (A.1b)) and consequently, Problem (A.2) has a complicating set of variables (the variables u in that case). This distinction motivates the use of a Dantzig–Wolfe reformulation for Problem (A.1) and a Benders decomposition for Problem (A.2). In the following subsections, we start by presenting those decompositions for the two introduced problems. Then we explain why those methods are in reality dual approaches.

A.1.1 Dantzig–Wolfe decomposition

The Dantzig–Wolfe reformulation aims at providing an efficient way to solve a linear program with a linking constraints structure. To exploit this structure, we start by reformulating Problem (A.1) as an optimization problem over the set of convex combinations of the extreme points and rays of the polyhedron $X = \{x \mid Ax = b, x \geqslant 0\}$.

Let P_X be the index set of extreme points of X and R_X be the index set of extreme rays of X. With the Minkowski–Weyl theorem, we have that:

$$X = \left\{ \sum_{i \in P_X} u_i x^i + \sum_{j \in R_X} v_j r^j \mid \sum_{i \in P_X} u_i = 1, u_i \geqslant 0, \forall i \in P_X, v_j \geqslant 0, \forall j \in R_X \right\}.$$

In that context, Problem (A.1) can be reformulated as follows:

$$\begin{aligned}
\max \; & \sum_{s \in P_X} \left(c^\top x^s\right) u_s + \sum_{t \in R_X} \left(c^\top r^t\right) v_t + d^\top y & & \text{(A.3a)} \\
\text{st: } & \sum_{s \in P_X} u_s = 1 & & \text{(A.3b)} \\
& \sum_{s \in P_X} \left(E x^s\right)^\top u_s + \sum_{t \in R_X} \left(E x^s\right)^\top v_t + F y = f & & \text{(A.3c)} \\
& u_s \geqslant 0 & \forall s \in P_X & \quad \text{(A.3d)} \\
& v_t \geqslant 0 & \forall t \in R_X. & \quad \text{(A.3e)}
\end{aligned}$$

Usually, the Dantzig–Wolfe reformulation does not consider extreme rays since in most problems and applications, the constraint polyhedron of Problem (A.1) is non-empty and bounded. In that case, the reformulation has the following form:

$$\begin{aligned}
\max \; & \sum_{s \in P_X} \left(c^\top x^s\right) u_s + d^\top y & & \text{(A.4a)} \\
\text{st: } & \sum_{s \in P_X} u_s = 1 & & \text{(A.4b)} \\
& \sum_{s \in P_X} \left(E x^s\right)^\top u_s + F y = f & & \text{(A.4c)} \\
& u_s \geqslant 0 & \forall s \in P_X. & \quad \text{(A.4d)}
\end{aligned}$$

Problems (A.3) and (A.4) have a big number of variables (potentially exponentially many in the size of the initial problem). Solving those problems directly is therefore particularly difficult. In order to circumvent this difficulty, it is quite natural to use a column generation. In this context, a central question is to determine which pricing problem should be used in order to generate the column at each step of the algorithm. At iteration k, we denote $P_X(k)$ and $R_X(k)$ the set of indices associated with the columns already added at step k.

The master problem solved for the Dantzig–Wolfe decomposition method at iteration k is then:

$$\max \sum_{s\in P_X(k)} \left(c^\top x^s\right) u_s + \sum_{t\in R_X(k)} \left(c^\top r^t\right) v_t + d^\top y \tag{A.5a}$$

$$\text{st:} \sum_{s\in P_X(k)} u_s = 1 \tag{A.5b}$$

$$\sum_{s\in P_X(k)} \left(Ex^s\right)^\top u_s + \sum_{t\in R_X(k)} \left(Ex^s\right)^\top v_t + Fy = f \tag{A.5c}$$

$$u_s \geqslant 0 \qquad \forall s \in P_X(k) \tag{A.5d}$$

$$v_t \geqslant 0 \qquad \forall t \in R_X(k). \tag{A.5e}$$

We denote by λ^k the dual variables associated with constraint (A.5c). A natural pricing problem aims at finding the variable that has the biggest possible reduced cost. In that context, the slave problem that is solved is the following:

$$\max \left(c^\top - \lambda^{k\top} A\right) x_i \tag{A.6a}$$

$$\text{st:} Ax = b \tag{A.6b}$$

$$x \geqslant 0 \tag{A.6c}$$

Notice that the dual variables of constraints (A.5b) do not appear in the slave problem. This is possible because we search for the variable with maximal reduced cost, and this variable only appears as a constant term in Problem (A.6).

The efficiency of the Dantzig–Wolf reformulation is highly related to the efficiency of the slave problem resolution. When the slave problem is efficiently solved, it allows the quick generation of columns for the master problem. Coupled with the fact that the number of columns generated is very small compared to the total number of solutions in S, the use of the Dantzig–Wolfe reformulation is usually efficient for linear programs with complicating constraints.

Some good insights about the Dantzig–Wolfe decomposition and more general decomposition methods can be found in Conforti et al. [35] for a theoretical approach and the polyhedral point of view and in Conejo et al. [34] for a more practical and computational approach.

A.1.2 Benders decomposition

The Benders decomposition aims at exploiting the particular structure of a linear program when it has complicating variables. The variables are complicating in the sense that if they are fixed, the remaining problem becomes easy to solve (for instance because it can then be decomposed). It is the case of Problem A.2 which has u as complicating variables, the problem can be seen as:

$$\max f^\top u + \eta(u) \tag{A.7a}$$

$$\text{st: } F^\top u \leqslant d \tag{A.7b}$$

where

$$\eta(u) = \max b^\top v \tag{A.8a}$$
$$\text{st: } A^\top v \leqslant c - E^\top u \tag{A.8b}$$

By duality, we have

$$\eta(u) = \min x^\top \left(c - E^\top u\right) \tag{A.9a}$$
$$\text{st: } Ax = b \tag{A.9b}$$
$$x \geqslant 0 \tag{A.9c}$$

The dual expression of η reveals that it is in fact a piecewise linear function with each slope corresponding to a given extreme point or an extreme ray of the polyhedron $X = \{x | Ax = b\}$. Problem A.7 can therefore be reformulated as follows, by Minkowski–Weyl theorem:

$$\max f^\top u + \eta \tag{A.10a}$$
$$\text{st: } F^\top u \leqslant d \tag{A.10b}$$
$$x^{s\top}\left(c - E^\top u\right) \geqslant \eta \qquad \forall s \in P_X \tag{A.10c}$$
$$r^{t\top}\left(c - E^\top u\right) \geqslant 0 \qquad \forall t \in R_X, \tag{A.10d}$$

where P_X is the index set of extreme points of X and R_X be the index set of extreme rays of X.

The Benders decomposition is based on the previous reformulation, and consists in operating a constraint generation. This constraint generation starts with Problem (A.10) without the constraints (A.10c) and (A.10d). This problem is called the master problem. At each iteration k, the master problem is solved and provides a solution u^k. For this solution, solution u^k generates a problem of the form (A.9). If the solution x^k exists, then an optimality cut of the form (A.10d) can be generated. If the solution does not exist, then an extreme ray can be found and a feasibility cut of the form (A.10c) can be generated.

Notice that for a given solution u^k, there might exist several solutions of the slave problem (A.9). All those solutions do no necessarily provide equivalent cuts, as it has been pointed out by Magnanti and Wong [92]. In order to compare the different cuts that can be generated, we say a cut $x^{k\top}\left(c - E^\top u\right) \geqslant \eta$ dominates another cut $x'^{k\top}\left(c - E^\top u\right) \geqslant \eta$ if $x^{k\top}\left(c - E^\top u\right) \geqslant x'^{k\top}\left(c - E^\top u\right)$ for all feasible solution of the master problem u. A cut is the said to be Pareto optimal if no other cut dominates it. Several ways exist to generate Pareto-optimal cuts, we refer to Magnanti and Wong [92], Papadakos [101] and Sherali and Lunday [122] for an introduction of this kind of problems.

A.1.3 Links between decomposition methods

An important fact to point out is that the Benders decomposition and the Dantzig–Wolfe decomposition are in fact dual from one another. Indeed, Problem (A.1) and Problem (A.2) are dual versions of one another. Consequently, the complicating constraints from Problem (A.1) correspond to the complicating variables in Problem (A.2). When looking closely at the master problem of the Dantzig–Wolfe decomposition, we have that it is also the dual version of the master problem of the Benders decomposition. Moreover, the slave problems of the two methods are the same. Consequently, the new variables generated at each iteration of the Dantzig–Wolfe decomposition method and the constraints generated at the same corresponding step of the Benders decomposition are in fact the same extreme points (or rays) of the slave problem polyhedra. A deeper analysis of the links between the Dantzig–Wolfe decomposition, the Benders decomposition and even the Lagrangian relaxation can be found in [88].

A.2 The concentration inequalities

The concentration inequalities are an extension of the Markov inequality. They give exponential bounds over the difference between a random variable and it's mean.

The Markov inequality simply provides a way to bound the tail of a non-negative random variable:

Theorem 44 (Markov inequality)**.** *For X a non-negative random variable and $t > 0$ the following inequality is always true:*

$$\mathbb{P}(X \geqslant t) \leqslant \frac{\mathbb{E}[X]}{t}$$

This inequality is the base of most concentration inequalities. A well known consequence of Markov inequality is the Chebyshev inequality which is the simplest concentration inequality: it quantifies the concentration of a random variable around its mean. For a given random variable X with mean μ and variance σ^2 the Chebyshev inequality provides a bound over the difference between the random variable and its mean which depends on the variance:

Proposition 45 (Chebyshev inequality)**.** *For any $t > 0$, the following equality is true:*

$$\mathbb{P}\big(|X - \mu|\big) \leqslant \frac{\sigma^2}{t^2}$$

Concentration inequalities generally aim at finding an exponential bound. In that sense the Chebyshev inequality is weak, and some tighter transformations over the random variable in the Markov inequality, the bounds can be much stronger. In particular some particular cases are particularly useful.

The Hoeffding inequality focuses on a series of iid Bernoulli random variables:

Proposition 46 (Hoeffding inequality)**.** *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables and $a = (a_1, \ldots, a_n) \in \mathbb{R}^n$. For any $t \geqslant 0$, we have:*

$$\mathbb{P}\left(\sum_{i=1}^{n} a_i X_i \geqslant t\right) \leqslant \exp\left(-\frac{t^2}{2\|a\|_2^2}\right).$$

As using only Bernoulli random variables can be quite limiting, a generalization of the previous inequality has been developed in the case of sub-gaussian random variables.

Proposition 47 (General Hoeffding inequality)**.** *Let $X_1, \cdots, X_n$ be independent mean-zero sub-gaussian random variables. For any $t \geqslant 0$, we have:*

$$\mathbb{P}\left|\sum_{i=1}^{n} X_i \geqslant t\right| \leqslant 2\exp\left(-c\frac{t^2}{\sum_{i=1}^{2}\|X_i\|_{\psi_2}^2}\right).$$

Some other inequalities exist for other types of random variables. For instance, in the case of sub-exponential random variables, the following inequality holds:

Proposition 48 (Bernstein inequality)**.** *Let $X_1, \cdots, X_n$ be independent mean-zero sub-exponential random variables. For any $t \geqslant 0$, we have:*

$$\mathbb{P}\left|\sum_{i=1}^{n} X_i \geqslant t\right| \leqslant 2\exp\left(-c\min\left(\frac{t^2}{\sum_{i=1}^{2}\|X_i\|_{\psi_1}^2}, \frac{t}{\max_i \|X_i\|_{\psi_1}}\right)\right).$$

A general introduction and most of the classical concentration inequalities can be found in the following book by Vershynin [144].

B Bibliographical remarks on aircraft operations

B.1 Fleet assignment and crew pairing problems

B.1.1 Fleet assignment

Barnhart and Cohn [9] give a clear overview of the different challenges inherent to the airline schedule planning problem. They state that airline schedule planning can be split in four sub-problems: The schedule design, the fleet assignment, the aircraft maintenance routing and the crew scheduling. Then they present the aims and the challenges linked with each sub problems. The authors precise that advanced models usually take the form of network design models, where two sets of decision variables are in relation: one linking the resources to the network, and one assigning the demand to the network.

The fleet assignment problem aims at finding which fleet type should be used for each flight that will be operated by the company. Classically, the fleet assignment has been introduced with a leg-based estimated revenue. Abara [1] and Hane et al. [65] were among the first to present fleet assignment models. Their model is based on a time-space network representation. After this early work, this leg-based fleet assignment problem has been broadly studied in the literature, and many variations have been worked on. One of the main practical drawback of the leg-based model is that it does not take into account the spill and recover mechanic of the demand. It also has a leg-based estimate of the demand, which can be a big approximation.

A way to circumvent these problems can be to add an estimation of the spill on to the model. Indeed, assigning a small aircraft to a flight with high potential demand results in a spill phenomenon i.e. people will buy itineraries on other flights due to the insufficient capacity. On the contrary, assigning a big aircraft to a flight with low potential demand will result in unsold seats. Evaluating the transfer of passengers among similar itineraries is called spill and recapture, and it has been an important topic of research to improve the fleet assignment models. Barnhart et al. [11] introduce a spill and recapture effect in the fleet assignment problem with an itinerary base dynamic for the demand. Lohatepanont and Barnhart [90] add leg selection in the model introduced in [11]. Since the size of the problem is too important to provide a good solution, they also provide an approximation method to get a solution to the problem. Barnhart et al. [10] present a framework to include general revenue estimation in the fleet assignment problem. Some work has also been dedicated to a more short terms fleet assignment problem. This problem is called re-fleeting and aims at changing the fleets of some flights in order to fit to the estimated demand. For instance, Sherali and Zhu [123] propose a two stage stochastic fleet assignment model, getting flexibility and resilience of the fleet affectation.

Sherali et al. [121] provide a general overview of the fleet assignment problem models and algorithms that can be found in the literature.

B.1.2 Crew pairing

The last problem we want to present here is the crew pairing problem. Crew pairing consists in finding the list of flights that each crew will operate. The main difficulty of this problem lies in the many and particular rules that the planning has to follow. Those rules are very often non-linear, and vary following many parameters. For instance, they enforce the rest time of the crew, asking for a certain amount of it, depending on the last flights they operated and on the places they are staying in. Early on, Lavoie et al. [83] have proposed a model to solve the problem. They formulate the crew pairing as a large scale set covering problem. They generate a variable per path in the graph that represents the sequence of flights operated by the crews and solve the set covering corresponding integer program with a column generation method. State-of-the-art resolution methods to solve crew pairing are column generations (see for instance [5, 12, 31]). The complexity of the problem usually lies in the pricing subproblem which is highly non-linear due to the particular constraints that need to be respected by the crew affectations in order to provide an admissible solution.

In the last two decades, the main focus on crew pairing has been its integration with aircraft routing (see Subsection B.2.2 for more details on that topic). Only a few works has focused uniquely on crew scheduling during this time. Subramanian and Sherali [127] present an advanced method of pricing in a column generation scheme, applied to the crew pairing problem. They identify that the dual solution generated with the column generation have dual instabilities, therefore they develop an adapted quadratic pricing subproblem to quicken the resolution. Zeren and Özkol [155] have recently proposed a branch and price algorithm with mixed heuristic and exact column generation approach to solve the problem. Gopalakrishnan and Johnson [60] review the main models and resolution techniques used to solve crew pairing.

B.2 Integrated approaches

One of the main area of research for the last twenty years has been the integration of the different presented problems with one another. We have already presented the links that have been made between the aircraft routing and the revenue management. We now look at the other development that have been led in the aircraft scheduling more generally.

B.2.1 Fleet assignment and aircraft routing

The two first steps of a global schedule optimization are fleet assignment and aircraft routing. Those problems gather a lot of research in the operations research for airlines . Naturally, in the last decades, integrating the two problems at the same time has been a subject of interest.

Sherali et al. [118] work on a MILP model that gather the fleet assignment and flight scheduling. Their model includes two types of flight legs, mandatory and optional ones, and includes the classical fleet assignment model. The model contains variables that indicate the number of passengers from a designated fare that are accepted on a certain path. This number is limited by the expected demand of each fare class on each path (it is basically a fluid demand approximation on each itinerary without spill and recover). In the article, they propose several improvements to the MILP formulation, through the inclusion of intermediary variable, the tightening of some cuts, and the introduction of some valid cuts. In order to find solutions for their model, they use a Benders decomposition. Sherali et al. [119] further develop this model including flight retiming and a recapture mechanic of the demand. The retiming consists in adding possible legs with some time shifts around the planned flight. In order to model the recapture, they model the flow of people spilled from a path and recaptured in another one. They penalize this value in the objective and linearize the natural non-linear constraints that model the recapture phenomena. They also solve this model with a Benders decomposition. It is reused and modified in Sherali et al. [120], where the aircraft maintenance routing is added to the integrated problem. The model is slightly different, as the graph used for the model is a flight network graph, rather than a state-time graph. This formulation of the problem has several advantages: it is more compact and allows the introduction of maintenance constraints. Once again, the model is solved using a Benders decomposition.

B.2.2 Aircraft routing and crew pairing

In the last decades, a lot of work has been done to join crew scheduling and aircraft routing. Since the two problems state-of-the-art resolution methods use a set partitioning formulation with a column generation approach, joining those two problems has been a natural line of research.

Cordeau et al. [36] work on a joined model which is based on a state-time network with nodes corresponding to flights, with each node needing to be covered. They formulate the problem as an integer linear program, with two types of variables: one for the paths of the aircraft and one for the paths of the crews. The objective minimizes the sum of the costs, and the model has one linking type of constraints, which forbids the crew to change of aircraft when the stop time is too short. They try two strategies to solve the problem. A first strategy uses a column generation (Dantzig-Wolfe decomposition) with a branch and bound. The second strategy uses a Benders decomposition to split the problem and then solves the master and the slave problems with column generation (Dantzig-Wolfe decomposition) and branch and bound to get an integer optimal solution. The second strategy seems to work better than the first.

Klabjan et al. [78] partially integrate the fleet assignment problem with the crew scheduling problem using a sequential approach. They use the ground values of the fleet assignment problem, and solve the crew pairing problem adding plane count constraints derived from ground arc variables. To solve the crew pairing problem, they don't generate all possible pairing, but only a part of them, and then solve a set covering ILP with the generated pairings, and the plane-count constraint.

Mercier et al. [98] further develop the model used by Cordeau et al. [36], using the concept of restricted connection, a concept that is useful for propagation delay modelling. Their model adds a constraint for restricted connection, associated with a penalty variable, which is activated every time a restricted connection is used. They also include a constraint that enforces the cyclic behaviour of the planning over a week. They use a Benders decomposition methods, with the crew pairing problem as a master problem. This differs from the resolution in [36] where the master problem is the aircraft routing problem. The generation of Pareto optimal cuts is used to quicken the Benders decomposition and gives decent results. Another extension is added in Mercier and Soumis [99] where the authors consider flight retiming on top of the integrated problems. As the number of constraints limiting the changes of the crews gets huge as the number of possible flights operated increases, they reformulate this constraint in a compact way. They use the same method as Mercier et al. [98] to solve the problem.

Ruther et al. [112] focus on solving an integrated aircraft routing, crew pairing and tail assignment problems a few days before the operation day. This is motivated by the increase in reliability of the data some days before the actual operations allows a much more precise optimization. Their model uses route variables for the planes and the crews. Routes respecting the planning rules are generated through column generation. They use a drive and price approach to generate a global heuristic integer solution. In their paper, they focus on the fine-tuning of the pricing subproblems. These problems are shortest path problems with resource constraints, and they mixed exact and heuristic approached to solve them. As many pricing problems are similar during the column generation, they only focus on solving the most changing one. They introduce the concept of superimposed pricing problems which are aggregation of the original pricing problem.

Parmentier and Meunier [104] have recently proposed an integrated method that combines a compact formulation (presented previously) of the aircraft routing problem and a column generation for the crew pairing problem, where the pricing subproblem structure is precisely studied and used. The pricing of the crew pairing problem is seen as a monoid resource constrained short path problem which has the advantage of having an exploitable ordered structure over the paths. This structure is used to quicken the resolution of the pricing problem and though it is in theory NP, it has very good practical effectiveness.

B.2.3 Multi-integrated problems

Some other works have developed general methods that integrate more than two operational problems at once. We end this review by presenting some examples of those multi-integrated problems.

Sandhu and Klabjan [115] work on a model integrating fleet assignment, crew pairing and some aspects of aircraft routing. To be sure that the aircraft routing problem is feasible, they include plane count constraints in their model (similar to Klabjan et al. [78]) which ensure that the maintenance slots will have enough room to be included in the aircraft planning. They compared Benders decomposition and Lagrangian decomposition to solve their problem. They conclude that Benders decomposition gives better results during the early stages of the resolution, while the Lagrangian method takes more time to better the initial solution, but obtains better results in the long run.

Cacchiani and Salazar-Gonzalez [26] have developed a multi-objective problem that incorporates crew pairing, fleet assignment and aircraft routing. This program is formulated as a MILP and solves to optimality on instances of a regional carrier (relatively small instances). To solve the problem to optimality, they solve the LP relaxation with column generation, then, use heuristics to derive a good bound, and a branch and price to find the optimal solution. They also use cuts to speed up the process. Their model is based on two acyclic graphs that model the aircraft feasible routes and the crew feasible routes. Their graph has two types of nodes (base nodes and classical nodes) with base node allowing the inclusion of maintenance constraints and crew social constraints. They use variables that model routes between bases, which motivates the use of a column generation to solve it.

Dunbar et al. [42] work on an integrated aircraft routing and crew pairing problem, that limits the delay propagation. To do that, they use a column generation approach, and estimated dynamically the combined propagated delay (crew and plane delays) to solve the pricing problem.

Papadakos [102] fully integrate fleet assignment, crew pairing and aircraft routing. He uses a route formulation for the crew pairing and the aircraft routing problems, with copies of the different variables for each fleet assignment. He uses a Benders decomposition to solve the problem. Contrary to what is presented in [98], he takes the crew pairing problem as the slave problem for computational results. The master and the slave problems have a very big number of variables, and thus a column generation is used. More precisely, he uses an approximate branch and price to get integer good solutions. He also accelerates the Benders decomposition with an enhanced Magnanti-Wong method, which finds cuts that are Pareto optimal. He enhances the method with a three phases decomposition to quicken the resolution.

Shao et al. [117] present a model that integrates fleet assignment, aircraft routing and crew scheduling. Their initial model is highly non-linear (constraint wise) but they use a method presented in [66] to linearize the non-linear constraints. Furthermore, they add valid inequalities on the lower bound on the number of required aircraft. To solve the problem, they use a Benders decomposition which generates two types of sub-problems: an itinerary-based passenger mix subproblem and a crew pairing subproblem. As the crew pairing subproblem is hard to solve, they use a branch and price heuristic, generating the columns through a perturbed Lagrangian dual of the crew pairing master problem (presented in [127]). To accelerate the generation of cuts and the convergence of the decomposition, they generate maximal-non dominated Benders cuts by adding a perturbation term in the constraints of the master problem.

C Proof of Lemma 14

In this appendix, we focus on the proof of Lemma 14. More precisely, we prove that $Z(\overline{T}_\theta(t), \theta y)_{t\in\mathbb{Z}}$ is a Markov chain with the same law as $Q(\theta t, \theta y)_{t\in\mathbb{Z}}$. For simplicity in this appendix, we will denote the two processes by Z_t and Q_t respectively

Proof. Let $\varsigma_t = \min\{t' \mid \sum_i W^i_{t'} \wedge y_i = t\}$. Then we have:

$$\begin{aligned}
&\mathbb{P}(Z_{t+1} = x_{t+1} \mid Z_0 = x_0, \dots, Z_t = x_t) \\
&= \sum_{s_0<\dots<s_{t+1}} \mathbb{P}(\varsigma_0 = s_0, \dots, \varsigma_{t+1} = s_{t+1}) \mathbb{P}(Z_{t+1} = x_{t+1} \mid Z_0 = x_0, \dots, Z_t = x_t, \varsigma_0 = s_0, \dots, \varsigma_{t+1} = s_{t+1})
\end{aligned}$$

where

$$\mathbb{P}(Z_{t+1} = x_{t+1} \mid Z_0 = x_0, \dots, Z_t = x_t, \varsigma_0 = s_0, \dots, \varsigma_{t+1} = s_{t+1}) \tag{C.1}$$
$$= \mathbb{P}\big(W_{s_{t+1}} = x_{t+1} \mid W_{s_0} = s_0, \dots, W_{s_t} = x_t, \varsigma_0 = s_0, \dots, \varsigma_{t+1} = s_{t+1}\big) \tag{C.2}$$
$$= \mathbb{P}\big(W_{s_{t+1}} = x_{t+1} \mid W_{s_t} = x_t, W_{s_t} \wedge y = x_t \wedge y, W_{s_0} = s_0, \dots, W_{s_t} = x_t, \varsigma_0 = s_0, \dots, \varsigma_{t+1} = s_{t+1}\big) \tag{C.3}$$
$$= \mathbb{P}\big(W_{s_{t+1}} = x_{t+1} \mid W_{s_t} = x_t, W_{s_t} \wedge y = x_t \wedge y\big) \tag{C.4}$$
$$= \begin{cases} \frac{\gamma_j \mathbb{1}(x_t^j < y_j)}{\sum_{k\in I} \gamma_k \mathbb{1}(x_t^k < y_k)} & \text{if } x_{t+1} = x_t + e_j \\ 0 & \text{otherwise,} \end{cases} \tag{C.5}$$

□

where from (C.2) to (C.3), we use the equality of the events $\{W_{s_t} = x_t, \varsigma_{t+1} = s_{t+1}, \varsigma_t = s_t\}$ and $\{W_{s_t} = x_t, W_{s_t} \wedge y = x_t \wedge y, \varsigma_t = s_t\}$. Step (C.4) follows from the fact that W is a Markov chain, and step (C.5) is a consequence of the definitions of W and ς.

Therefore, we get that Z is a Markov chain with the same initial conditions and transition matrix as Q, which gives the result.

Bibliography

[1] Abara, J. (1989). Applying Integer Linear Programming to the Fleet Assignment Problem. *Interfaces*, 19(4):20–28.

[2] Abdelghany, A., Abdelghany, K., and Azadian, F. (2017). Airline flight schedule planning under competition. *Computers & Operations Research*, 87:20–39.

[3] Adelman, D. (2007). Dynamic Bid Prices in Revenue Management. *Operations Research*, 55(4):647–661.

[4] Aloulou, M. A., Haouari, M., and Zeghal Mansour, F. (2013). A model for enhancing robustness of aircraft and passenger connections. *Transportation Research Part C: Emerging Technologies*, 32:48–60.

[5] Anbil, R., Tanga, R., and Johnson, E. L. (1992). A global approach to crew-pairing optimization. *IBM Systems Journal*, 31(1):71–78.

[6] Aouad, A., Levi, R., and Segev, D. (2018). Greedy-Like Algorithms for Dynamic Assortment Planning Under Multinomial Logit Preferences. *Operations Research*, 66(5):1321–1345.

[7] Aouad, A. and Segev, D. (2019). The Stability of MNL-Based Demand under Dynamic Customer Substitution and its Algorithmic Implications. *SSRN Electronic Journal*.

[8] Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L., and Shenoi, R. G. (1998). Flight String Models for Aircraft Fleeting and Routing. *Transportation Science*, 32(3):208–220.

[9] Barnhart, C. and Cohn, A. (2004). Airline Schedule Planning: Accomplishments and Opportunities. *Manufacturing & Service Operations Management*, 6(1):3–22.

[10] Barnhart, C., Farahat, A., and Lohatepanont, M. (2009). Airline Fleet Assignment with Enhanced Revenue Modeling. *Operations Research*, 57(1):231–244.

[11] Barnhart, C., Kniker, T. S., and Lohatepanont, M. (2002). Itinerary-Based Airline Fleet Assignment. *Transportation Science*, 36(2):199–217.

[12] Barnhart, C. and Shenoi, R. G. (1998). An Approximate Model and Solution Approach for the Long-Haul Crew Pairing Problem. *Transportation Science*, 32(3):221–231.

[13] Belobaba, P. P. (1989). OR Practice—Application of a Probabilistic Decision Model to Airline Seat Inventory Control. *Operations Research*, 37(2):183–197.

[14] Ben-Akiva, M. E., Lerman, S. R., and Lerman, S. R. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*, volume 9. MIT Press.

[15] Ben-Ameur, W. and Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17.

[16] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252.

[17] Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J.-P., and Bach, F. (2020). Learning with Differentiable Perturbed Optimizers. *arXiv:2002.08676 [cs, math, stat]*. arXiv: 2002.08676.

[18] Bertsimas, D., Lulli, G., and Odoni, A. (2011). An Integer Optimization Approach to Large-Scale Air Traffic Flow Management. *Operations Research*, 59(1):211–227.

[19] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

[20] Blanchet, J., Gallego, G., and Goyal, V. (2016). A Markov Chain Approximation to Choice Modeling. *Operations Research*, 64(4):886–905.

[21] Blondel, M., Martins, A. F. T., and Niculae, V. (2020). Learning with Fenchel-Young Losses. page 69.

[22] Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L., and Tarjan, R. E. (1973). Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461.

[23] Bront, J. J. M., Mendez-Diaz, I., and Vulcano, G. (2009). A Column Generation Algorithm for Choice-Based Network Revenue Management. *Operations Research*, 57(3):769–784.

[24] Brumelle, S. L. and McGill, J. I. (1993). Airline Seat Allocation with Multiple Nested Fare Classes. *Operations Research*, 41(1):127–137.

[25] Burke, E. K., De Causmaecker, P., De Maere, G., Mulder, J., Paelinck, M., and Vanden Berghe, G. (2010). A multi-objective approach for robust airline scheduling. *Computers & Operations Research*, 37(5):822–832.

[26] Cacchiani, V. and Salazar-Gonzalez, J.-J. (2017). Optimal Solutions to a Real-World Integrated Airline Scheduling Problem. *Transportation Science*, 51(1):250–268.

[27] Cacchiani, V. and Salazar-Gonzalez, J.-J. (2020). Heuristic approaches for flight retiming in an integrated airline scheduling problem of a regional carrier. *Omega*, 91:102028.

[28] Cadarso, L., Vaze, V., Barnhart, C., and Main, A. (2017). Integrated Airline Scheduling: Considering Competition Effects and the Entry of the High Speed Rail. *Transportation Science*, 51(1):132–154.

[29] Cao, Y., Rusmevichientong, P., and Topaloglu, H. (2020). Revenue Management under a Mixture of Multinomial Logit and Independent Demand Models. Technical report.

[30] Castelli, L., Pellegrini, P., and Pesenti, R. (2012). Airport slot allocation in Europe: economic efficiency and fairness. *International Journal of Revenue Management*, 6(1/2):28.

[31] Chu, H. D., Gelman, E., and Johnson, E. L. (1997). Solving large scale crew scheduling problems. *European Journal of Operational Research*, 97(2):260–268.

[32] Clarke, L., Johnson, E., Nemhauser, G., and Zhu, Z. (1997). The aircraft rotation problem. *Annals of Operations Research*, 69:33–46.

[33] Coldren, G. M. (2005). *Modeling the competitive dynamic among air-travel itineraries with generalized extreme value models*. PhD thesis, Citeseer.

[34] Conejo, A. J., Castillo, E., Minguez, R., and Garcia-Bertrand, R. (2006). *Decomposition techniques in mathematical programming: engineering and science applications*. Springer Science & Business Media.

[35] Conforti, M., Cornuéjols, G., Zambelli, G., et al. (2014). *Integer programming*, volume 271. Springer.

[36] Cordeau, J.-F., Stojkovic, G., Soumis, F., and Desrosiers, J. (2001). Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling. *Transportation Science*, 35(4):375–388.

[37] Corolli, L., Lulli, G., and Ntaimo, L. (2014). The time slot allocation problem under uncertain capacity. *Transportation Research Part C: Emerging Technologies*, 46:16–29.

[38] Davis, J., Gallego, G., and Topaloglu, H. (2013). Assortment Planning under the Multinomial Logit Model with Totally Unimodular Constraint Structures. *Work in progress*, page 22.

[39] Davis, J. M., Gallego, G., and Topaloglu, H. (2014). Assortment Optimization Under Variants of the Nested Logit Model. *Operations Research*, 62(2):250–273.

[40] Demange, M., Monnot, J., Pop, P., and Ries, B. (2012). Selective Graph Coloring in Some Special Classes of Graphs. In *Combinatorial Optimization*, volume 7422, pages 320–331. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

[41] Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. (1997). Daily Aircraft Routing and Scheduling. *Management Science*, 43(6):841–855.

[42] Dunbar, M., Froyland, G., and Wu, C.-L. (2012). Robust Airline Schedule Planning: Minimizing Propagated Delay in an Integrated Routing and Crewing Framework. *Transportation Science*, 46(2):204–216.

[43] Dunbar, M., Froyland, G., and Wu, C.-L. (2014). An integrated scenario-based approach for robust aircraft routing, crew pairing and re-timing. *Computers & Operations Research*, 45:68–86.

[44] Dunning, I., Huchette, J., and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM review*, 59(2):295–320.

[45] Désir, A. and Goyal, V. (2014). Near-Optimal Algorithms for Capacity Constrained Assortment Optimization. *SSRN Electronic Journal*.

[46] Désir, A., Goyal, V., Segev, D., and Ye, C. (2020). Constrained Assortment Optimization Under the Markov Chain–based Choice Model. *Management Science*, 66(2):698–721.

[47] Etschmaier, M. and Mathaisel, D. F. X. (1984). Aircraft scheduling: The state of the art. *AGIFORS*, 24:181–225.

[48] Feldman, J. B. and Topaloglu, H. (2017). Revenue Management Under the Markov Chain Choice Model. *Operations Research*, 65(5):1322–1342.

[49] Feng, G., Li, X., and Wang, Z. (2017). Technical Note-On the Relation Between Several Discrete Choice Models. *Operations Research*, 65(6):1516–1525.

[50] Feo, T. A. and Bard, J. F. (1989). Flight Scheduling and Maintenance Base Planning. *Management Science*, 35(12):1415–1432.

[51] Frank, A. and Murota, K. (2020a). Decreasing Minimization on M-convex Sets. *arXiv:2007.09616 [math]*. arXiv: 2007.09616.

[52] Frank, A. and Murota, K. (2020b). Decreasing Minimization on M-convex Sets: Algorithms and Applications. *arXiv:2007.09618 [math]*. arXiv: 2007.09618.

[53] Froyland, G., Maher, S. J., and Wu, C.-L. (2014). The Recoverable Robust Tail Assignment Problem. *Transportation Science*, 48(3):351–372.

[54] Gabteni, S. and Grönkvist, M. (2009). Combining column generation and constraint programming to solve the tail assignment problem. *Annals of Operations Research*, 171(1):61–76.

[55] Gallego, G., Iyengar, G., Phillips, R., and Dubey, A. (2004). Managing Flexible Products on a Network. *Available at SSRN 3567371*.

[56] Gallego, G., Ratliff, R., and Shebalov, S. (2015). A General Attraction Model and Sales-Based Linear Program for Network Revenue Management Under Customer Choice. *Operations Research*, 63(1):212–232.

[57] Gallego, G. and Topaloglu, H. (2019). *Assortment Optimization*, pages 129–160. Springer New York, New York, NY.

[58] Garay, M. and Johnson, D. (1979). Computers and intractability-a guide to the theory of NP-completeness.

[59] Garey, M. R., Johnson, D. S., Miller, G. L., and Papadimitriou, C. H. (1980). The Complexity of Coloring Circular Arcs and Chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227.

[60] Gopalakrishnan, B. and Johnson, E. L. (2005). Airline Crew Scheduling: State-of-the-Art. *Annals of Operations Research*, 140(1):305–337.

[61] Gopalan, R. and Talluri, K. T. (1998). The Aircraft Maintenance Routing Problem. *Operations Research*, 46(2):260–271.

[62] Goyal, V., Levi, R., and Segev, D. (2016). Near-Optimal Algorithms for the Assortment Planning Problem Under Dynamic Substitution and Stochastic Demand. *Operations Research*, 64(1):219–235.

[63] Grani, G., Leo, G., Palagi, L., Piacentini, M., and Toyoglu, H. (1977). The Sales Based Integer Program for Post-Departure Analysis in Airline Revenue Management: model and solution. Technical report.

[64] Gurobi Optimization, L. (2021). Gurobi optimizer reference manual.

[65] Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., and Sigismondi, G. (1995). The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming*, 70(1-3):211–232.

[66] Haouari, M., Shao, S., and Sherali, H. D. (2013). A Lifted Compact Formulation for the Daily Aircraft Maintenance Routing Problem. *Transportation Science*, 47(4):508–525.

[67] Harsha, P. (2009). *Mitigating Airport Congestion: Market Mechanisms and Airline Response Models*. PhD thesis, Massachusetts Institute of Technology.

[68] Harvey, N. J., Ladner, R. E., Lovász, L., and Tamir, T. (2006). Semi-matchings for bipartite graphs and load balancing. *Journal of Algorithms*, 59(1):53–78.

[69] Hofbauer, J. and Sandholm, W. H. (2002). On the global convergence of stochastic fictitious play. *Econometrica*, 70(6):2265–2294.

[70] Honhon, D., Gaur, V., and Seshadri, S. (2010). Assortment Planning and Inventory Decisions Under Stockout-Based Substitution. *Operations Research*, 58(5):1364–1379.

[71] Jacobs, T. L., Smith, B. C., and Johnson, E. L. (2008). Incorporating Network Flow Effects into the Airline Fleet Assignment Process. *Transportation Science*, 42(4,):514–529.

[72] Jacquillat, A. and Odoni, A. R. (2015). An Integrated Scheduling and Operations Approach to Airport Congestion Mitigation. *Operations Research*, 63(6):1390–1410.

[73] Jagabathula, S. (2014). Assortment Optimization Under General Choice. *SSRN Electronic Journal*.

[74] Jamili, A. (2017). A robust mathematical model and heuristic algorithms for integrated aircraft routing and scheduling, with consideration of fleet assignment problem. *Journal of Air Transport Management*, 58:21–30.

[75] Jiang, H. (2006). *Dynamic airline scheduling and robust airline schedule de-peaking*. PhD thesis, Massachusetts Institute of Technology.

[76] Khaled, O., Minoux, M., Mousseau, V., Michel, S., and Ceugniet, X. (2018a). A compact optimization model for the tail assignment problem. *European Journal of Operational Research*, 264(2):548–557.

[77] Khaled, O., Minoux, M., Mousseau, V., Michel, S., and Ceugniet, X. (2018b). A multi-criteria repair/recovery framework for the tail assignment problem in airlines. *Journal of Air Transport Management*, 68:137–151.

[78] Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E., and Ramaswamy, S. (2002). Airline Crew Scheduling with Time Windows and Plane-Count Constraints. *Transportation Science*, 36(3):337–348.

[79] Kunnumkal, S. and Talluri, K. (2019). A strong Lagrangian relaxation for general discrete-choice network revenue management. *Computational Optimization and Applications*, 73(1):275–310.

[80] Kunnumkal, S. and Topaloglu, H. (2009). A New Dynamic Programming Decomposition Method for the Network Revenue Management Problem with Customer Choice Behavior. *Production and Operations Management*, page 16.

[81] Kök, A. G., Fisher, M. L., and Vaidyanathan, R. (2008). Assortment Planning: Review of Literature and Industry Practice. In Agrawal, N. and Smith, S. A., editors, *Retail Supply Chain Management*, volume 122, pages 99–153. Springer US, Boston, MA. Series Title: International Series in Operations Research & Management Science.

[82] Lan, S., Clarke, J.-P., and Barnhart, C. (2006). Planning for Robust Airline Operations: Optimizing Aircraft Routings and Flight Departure Times to Minimize Passenger Disruptions. *Transportation Science*, 40(1):15–28.

[83] Lavoie, S., Minoux, M., and Odier, E. (1988). A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35(1):45–58.

[84] Levin, A. (1971). Scheduling and Fleet Routing Models for Transportation Systems. *Transportation Science*, 5(3):232–255.

[85] Liang, D., Ratliff, R., and Remenyi, N. (2017). Robust revenue opportunity modeling with quadratic programming. *Journal of Revenue and Pricing Management*, 16(6):569–579.

[86] Liang, Z. and Chaovalitwongse, W. A. (2013). A Network-Based Model for the Integrated Weekly Aircraft Maintenance Routing and Fleet Assignment Problem. *Transportation Science*, 47(4):493–507.

[87] Liang, Z., Chaovalitwongse, W. A., Huang, H. C., and Johnson, E. L. (2011). On a New Rotation Tour Network Model for Aircraft Maintenance Routing Problem. *Transportation Science*, 45(1):109–120.

[88] Lim, C. (2011). Relationship among Benders, Dantzig-Wolfe, and Lagrangian Optimization. In *Wiley Encyclopedia of Operations Research and Management Science*, page eorms0717. John Wiley & Sons, Inc., Hoboken, NJ, USA.

[89] Liu, Q. and van Ryzin, G. (2008). On the Choice-Based Linear Programming Model for Network Revenue Management. *Manufacturing & Service Operations Management*, 10(2):288–310.

[90] Lohatepanont, M. and Barnhart, C. (2004). Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation science*, 38(1):19–32.

[91] Luce, R. D. (1959). Individual choice behavior.

[92] Magnanti, T. L. and Wong, R. T. (1981). Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29(3):464–484. ZSCC: 0000672.

[93] Mahajan, S. and van Ryzin, G. (2001). Stocking Retail Assortments Under Dynamic Consumer Substitution. *Operations Research*, 49(3):334–351.

[94] McFadden, D. et al. (1973). Conditional logit analysis of qualitative choice behavior.

[95] McGill, J. I. and van Ryzin, G. J. (1999). Revenue Management: Research Overview and Prospects. *Transportation Science*, 33(2):233–256.

[96] Meissner, J. and Strauss, A. (2012). Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research*, 216(2):459–468.

[97] Meissner, J., Strauss, A., and Talluri, K. (2013). An Enhanced Concave Program Relaxation for Choice Network Revenue Management. *Production and Operations Management*, 22(1):71–87.

[98] Mercier, A., Cordeau, J.-F., and Soumis, F. (2005). A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476.

[99] Mercier, A. and Soumis, F. (2007). An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34(8):2251–2265.

[100] Natarajan, K., Song, M., and Teo, C.-P. (2009). Persistency Model and Its Applications in Choice Modeling. *Management Science*, 55(3):453–469.

[101] Papadakos, N. (2008). Practical enhancements to the Magnanti–Wong method. *Operations Research Letters*, 36(4):444–449. ZSCC: 0000187.

[102] Papadakos, N. (2009). Integrated airline scheduling. *Computers & Operations Research*, 36(1):176–195.

[103] Parmentier, A. (2021). Learning to Approximate Industrial Problems by Operations Research Classic Problems. *Operations Research*, page opre.2020.2094.

[104] Parmentier, A. and Meunier (2019). Aircraft routing and crew pairing: Updated algorithms at Air France. *Omega*, page 102073.

[105] Pita, J. P., Barnhart, C., and Antunes, A. P. (2013). Integrated Flight Scheduling and Fleet Assignment Under Airport Congestion. *Transportation Science*, 47(4):477–492.

[106] PODS-Simulator (2021). Passenger origin-destination simulator website. Accessed: 2021-07-27.

[107] Pyrgiotis, N. and Odoni, A. (2016). On the Impact of Scheduling Limits: A Case Study at Newark Liberty International Airport. *Transportation Science*, 50(1):150–165.

[108] Rahmaniani, R., Crainic, T. G., Gendreau, M., and Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817.

[109] Ribeiro, N. A., Jacquillat, A., Antunes, A. P., Odoni, A. R., and Pita, J. P. (2018). An optimization approach for airport slot allocation under IATA guidelines. *Transportation Research Part B: Methodological*, 112:132–156.

[110] Rusmevichientong, P., Shen, Z.-J. M., and Shmoys, D. B. (2010). Dynamic Assortment Optimization with a Multinomial Logit Choice Model and Capacity Constraint. *Operations Research*, 58(6):1666–1680.

[111] Rusmevichientong, P., Shmoys, D., Tong, C., and Topaloglu, H. (2014). Assortment Optimization under the Multinomial Logit Model with Random Choice Parameters. *Production and Operations Management*, 23(11):2023–2039.

[112] Ruther, S., Boland, N., Engineer, F. G., and Evans, I. (2017). Integrated Aircraft Routing, Crew Pairing, and Tail Assignment: Branch-and-Price with Many Pricing Problems. *Transportation Science*, 51(1):177–195.

[113] Safaei, N. and Jardine, A. K. (2018). Aircraft routing with generalized maintenance constraints. *Omega*, 80:111–122.

[114] Salazar-González, J.-J. (2014). Approaches to solve the fleet-assignment, aircraft-routing, crew-pairing and crew-rostering problems of a regional carrier. *Omega*, 43:71–82.

[115] Sandhu, R. and Klabjan, D. (2007). Integrated Airline Fleeting and Crew-Pairing Decisions. *Operations Research*, 55(3):439–456.

[116] Sarac, A., Batta, R., and Rump, C. M. (2006). A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operational Research*, 175(3):1850–1869.

[117] Shao, S., Sherali, H. D., and Haouari, M. (2017). A Novel Model and Decomposition Approach for the Integrated Airline Fleet Assignment, Aircraft Routing, and Crew Pairing Problem. *Transportation Science*, 51(1):233–249.

[118] Sherali, H. D., Bae, K.-H., and Haouari, M. (2010). Integrated Airline Schedule Design and Fleet Assignment: Polyhedral Analysis and Benders' Decomposition Approach. *INFORMS Journal on Computing*, 22(4):500–513.

[119] Sherali, H. D., Bae, K.-H., and Haouari, M. (2013a). A benders decomposition approach for an integrated airline schedule design and fleet assignment problem with flight retiming, schedule balance, and demand recapture. *Annals of Operations Research*, 210(1):213–244.

[120] Sherali, H. D., Bae, K.-H., and Haouari, M. (2013b). An Integrated Approach for Airline Flight Selection and Timing, Fleet Assignment, and Aircraft Routing. *Transportation Science*, 47(4):455–476.

[121] Sherali, H. D., Bish, E. K., and Zhu, X. (2006). Airline fleet assignment concepts, models, and algorithms. *European Journal of Operational Research*, 172(1):1–30.

[122] Sherali, H. D. and Lunday, B. J. (2013). On generating maximal nondominated Benders cuts. *Annals of Operations Research*, 210(1):57–72. ZSCC: 0000072.

[123] Sherali, H. D. and Zhu, X. (2008). Two-Stage Fleet Assignment Model Considering Stochastic Passenger Demands. *Operations Research*, 56(2):383–399.

[124] Spieksma, F. C. R. (1999). On the approximability of an interval scheduling problem. *J. Sched.*, page 13.

[125] Sriram, C. and Haghani, A. (2003). An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*, 37(1):29–48.

[126] Strauss, A. K., Klein, R., and Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*, 271(2):375–387.

[127] Subramanian, S. and Sherali, H. D. (2008). An Effective Deflected Subgradient Optimization Scheme for Implementing Column Generation for Large-Scale Airline Crew Scheduling Problems. *INFORMS Journal on Computing*, 20(4):565–578.

[128] Talluri, K. (2014). New Formulations for Choice Network Revenue Management. *INFORMS Journal on Computing*, 26(2):401–413.

[129] Talluri, K. and van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Manage. Sci.*, 50(1):15–33.

[130] Talluri, K. T. (1998). The Four-Day Aircraft Maintenance Routing Problem. *Transportation Science*, 32(1):43–53.

[131] Talluri, K. T. and Van Ryzin, G. J. (2006). *The theory and practice of revenue management*, volume 68. Springer Science & Business Media.

[132] Tardos, E. (1986). A Strongly Polynomial Algorithm to Solve Combinatorial Linear Programs. *Operations Research*, 34(2):250–256.

[133] Thurstone, L. L. (1927). A law of comparative judgment. *Psychological review*, 34(4):273.

[134] Topaloglu, H. (2009). Using Lagrangian Relaxation to Compute Capacity-Dependent Bid Prices in Network Revenue Management. *Operations Research*, 57(3):637–649.

[135] Topaloglu, H. (2013). Joint Stocking and Product Offer Decisions Under the Multinomial Logit Model. *Production and Operations Management*.

[136] Topaloglu, H., Birbil, S. I., Frenk, J. B. G., and Noyan, N. (2012). Tractable Open Loop Policies for Joint Overbooking and Capacity Control Over a Single Flight Leg with Multiple Fare Classes. *Transportation Science*, 46(4):460–481.

[137] Tucker, A. (1975). Coloring a family of circular arcs. *SIAM Journal on Applied Mathematics*, 29(3):493–502.

[138] Valencia-Pabon, M. (2003). Revisiting tucker's algorithm to color circular arc graphs. *SIAM Journal on Computing*, 32(4):1067–1072.

[139] Van Bevern, R., Mnich, M., Niedermeier, R., and Weller, M. (2015). Interval scheduling and colorful independent sets. *Journal of Scheduling*, 18(5):449–469. arXiv: 1402.0851.

[140] van Ryzin, G. and McGill, J. (2000). Revenue Management Without Forecasting or Optimization: An Adaptive Algorithm for Determining Airline Seat Protection Levels. *Management Science*, 46(6):760–775.

[141] van Ryzin, G. and Vulcano, G. (2017). Technical Note—An Expectation-Maximization Method to Estimate a Rank-Based Choice Model of Demand. *Operations Research*, 65(2):396–407.

[142] Vaze, V. and Barnhart, C. (2012). Modeling Airline Frequency Competition for Airport Congestion Mitigation. *Transportation Science*, 46(4):512–535.

[143] Vershynin, R. (2018a). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press. ZSCC: 0000625 Google-Books-ID: TahxDwAAQBAJ.

[144] Vershynin, R. (2018b). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

[145] Vulcano, G., van Ryzin, G., and Chaar, W. (2010). OM Practice —Choice-Based Revenue Management: An Empirical Study of Estimation and Optimization. *Manufacturing & Service Operations Management*, 12(3):371–392.

[146] Vulcano, G., van Ryzin, G., and Ratliff, R. (2012). Estimating Primary Demand for Substitutable Products from Sales Transaction Data. *Operations Research*, 60(2):313–334.

[147] Wang, R. (2012). Capacitated assortment and price optimization under the multinomial logit model. *Operations Research Letters*, 40(6):492–497.

[148] Warburg, V., Gotsæd Hansen, T., Larsen, A., Norman, H., and Andersson, E. (2008). Dynamic airline scheduling: An analysis of the potentials of refleeting and retiming. *Journal of Air Transport Management*, 14(4):163–167.

[149] Wei, K., Vaze, V., and Jacquillat, A. (2019). Airline Timetable Development and Fleet Assignment Incorporating Passenger Choice. *Transportation Science*, pages 381–394.

[150] Weide, O., Ryan, D., and Ehrgott, M. (2010). An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 37(5):833–844.

[151] Yan, S., Tang, C., and Lee, M. (2007). A flight scheduling model for Taiwan airlines under market competitions. *Omega*, 35(1):61–74.

[152] Yan, S., Tang, C.-H., and Fu, T.-C. (2008). An airline scheduling model and solution algorithms under stochastic demands. *European Journal of Operational Research*, 190(1):22–39.

[153] Yan, S. and Tseng, C.-H. (2002). A passenger demand model for airline fight scheduling and fleet routing. *Operations Research*, page 23.

[154] Zaourar, S. and Malick, J. (2014). Quadratic stabilization of Benders decomposition. page 28.

[155] Zeren, B. and Özkol, I. (2016). A novel column generation strategy for large scale airline crew pairing problems. *Expert Systems with Applications*, 55:133–144.

[156] Zhang, D. and Adelman, D. (2009). An Approximate Dynamic Programming Approach to Network Revenue Management with Customer Choice. *Transportation Science*, 43(3):381–394.

[157] Zhang, D. and Cooper, W. L. (2005). Revenue Management for Parallel Flights with Customer-Choice Behavior. *Operations Research*, page 18.