École doctorale Mathématiques et Sciences et Technologies de l'Information et de la Communication

Thèse de doctorat

Spécialité : Mathématiques

Présentée par

Laurent Daudet

Pour obtenir le grade de

Docteur de l'Université Paris-Est

Algorithms for train scheduling on a single line

Quelques algorithmes de planification ferroviaire sur voie unique

Soutenance le 22 décembre 2017 devant le jury composé de :

Mme Luce Brotcorne	*INRIA*	Rapporteur
M. Christophe Rapine	*Université de Lorraine*	Rapporteur
M. Michel Bierlaire	*EPFL*	Examinateur
M. Dominique Feillet	*Mines de Saint-Étienne*	Examinateur
M. Axel Parmentier	*École des Ponts ParisTech*	Examinateur
M. Frédéric Meunier	*École des Ponts ParisTech*	Directeur de thèse

À mon papa.

Remerciements

Je me retourne aujourd'hui sur ces trois belles années de thèse au cours desquelles j'ai grandi personnellement et professionnellement. Ce fut un enrichissement quotidien et avec un peu de recul, je réalise la chance et le bonheur que ce fut d'être entouré par chacun d'entre vous.

Je souhaiterais bien évidemment commencer par remercier mon directeur de thèse, Frédéric Meunier. Merci pour de m'avoir si bien accompagné pendant ces trois ans de recherche. J'ai énormément appris en travaillant à tes côtés, grâce à ta rigueur et tes connaissances. Les après-midis au tableau vont me manquer ! Merci d'être si disponible, même pendant ces quelques semaines à l'autre bout du monde, ton soutien et ton amitié sont précieux.

Mes remerciements vont ensuite à Luce Brotcorne et Christophe Rapine, pour avoir accepté de rapporter ma thèse. Vos remarques m'ont beaucoup servi dans l'écriture de mon manuscrit. Je remercie également Michel Bierlaire qui m'a accueilli à Lausanne pour quelques semaines qui ont été très enrichissantes et qui a effectué aujourd'hui le voyage dans l'autre sens, Dominique Feillet pour être venu depuis Gardanne et Axel Parmentier, qui ont accepté de siéger dans mon jury.

J'aimerais remercier, avec une attention toute particulière, Guillaume Rault et Jean-François Lenglen ainsi que toutes les personnes que j'ai pu côtoyer chez Eurotunnel. Travailler ensemble était un vrai plaisir et vos retours m'ont été très utiles dans l'avancée de mes recherches.

Un immense merci à tous les chercheurs du Cermics avec qui j'ai pu travailler. C'était toujours un plaisir d'échanger avec vous. Je pense aussi à mes partenaires de course à pied : des plus assidus à ceux plus occasionnels que j'ai pris plaisir à martyriser, merci pour ces échanges sportifs et ces moments partagés. Un immense merci à tous les doctorants du labo que j'ai pu rencontrer pendant ces trois ans. Discuter, rigoler et sortir avec vous ont rendu cette période d'autant plus agréable ! Je n'oublie pas la fée du Cermics, merci Isabelle de nous prendre par la main tous les jours. Enfin, un remerciement tout particulier aux doctorants de Frédéric que j'ai connus. Merci Thomas, Pauline, Axel et Étienne, nous formons une belle famille !

Je pense aussi à tous mes amis sur qui je peux compter sans faille depuis les bancs de l'école. C'est toujours un plaisir de vous retrouver pour un verre. Merci à tous !

Merci à ma famille pour tout leur soutien. Il est difficile d'exprimer en quelques lignes toute la gratitude que j'ai pour vous. Merci Maman d'être si présente et de toujours nous encourager. Ton soutien et ton amour indéfectible sont si précieux. Merci Hélène pour tout ce que tu m'as apporté depuis 26 ans. Merci François, c'est cool d'avoir un petit frère à maîtriser ! Merci Olivier et Alexandre d'avoir été là quand j'en avais besoin, c'est cool d'avoir des grands frères aussi ! Il n'en manque qu'un, qui aurait été si fier. Merci Papa d'avoir été un exemple, de m'avoir montré le chemin. Ta présence était si précieuse et ton absence si pesante. Je reprends le flambeau des docteurs Daudet !

Enfin, un immense merci à ma femme. Véro, que ferais-je sans toi ? Merci d'être là tous les jours. Ta présence est un cadeau.

Abstract

This thesis develops algorithms for rail transportation problems, conducted in relationship with the company Eurotunnel which operates the tunnel under the Channel. This partnership is a scientific chair with the École des Ponts et Chaussées, where this thesis was realized. We study three topics throughout the thesis: the first one is an operational problem faced by Eurotunnel, whereas the two other ones are prospective and theoretical problems inspired by their process.

The planning process for rail transportation can be divided into several phases (demand estimation, line planning, scheduling of the departure times, rolling stock and crew planning). In a first part, we focus on the scheduling phase on a time interval, applied to the specific case of Eurotunnel. The objective is to compute the departure times of the trains for each of the two stations (Calais in France and Folkestone in England), satisfying operation constraints (security, loading, ...) and commercial agreements with their partners (Eurostar, ...). Moreover, it is essential to take into account the delays in the scheduling phase to limit the propagation of the disturbances from train to train in the network. We develop scheduling algorithms for Eurotunnel that consider the operation and commercial constraints, and the random distributions of the delays for each train. These algorithms use standard tools of Operations Research to model and solve these optimization problems.

Pricing is a main issue for transportation companies. Many algorithms have been proposed to help airline companies to define optimized prices of the plane tickets for different classes of passengers. In a second part, we apply some standard pricing frameworks (discrete choice models) in order to optimize in a global way the prices and the departure times of the trains for rail transportation companies. Standard tools of stochastic optimization, discrete choice models, and some heuristics are used in our algorithms to compute the best possible solutions in a limited computation time.

We focus in a last part on a class of transportation problems inspired form Eurotunnel. We give efficient algorithms to solve exactly or to approximate the optimal solutions of these problems. These algorithms give an upper bound of the time complexity of this class of problems. The problems studied consist in scheduling the departure times of shuttles on a fixed trip, to transport passengers, arriving continuously at an initial station, to a given destination. The shuttles are potentially allowed to perform several rotations to transport several groups of passengers. The objective is to minimize the waiting time of the passengers before the depart of their shuttle. Custom combinations of convex optimization and graph theory (shortest path problems) are used in our algorithms.

Key words: rail transportation, scheduling, delays, stochastic optimization, pricing, heuristics, time complexity, convex optimization, shortest path problems.

Résumé

Cette thèse développe des algorithmes pour des problèmes de transport ferroviaire et est réalisée en partenariat avec l'entreprise Eurotunnel qui exploite le tunnel sous la Manche. Ce partenariat s'est établi sous la forme d'une chaire avec l'École des Ponts où cette thèse a été menée. Nous développons trois sujets dans cette thèse : le premier est un problème opérationnel rencontré par Eurotunnel, les deux autres sont plus prospectifs et théoriques, et sont inspirés des problèmes de transport ferroviaire d'Eurotunnel.

Le processus de création de grilles horaires pour le transport ferroviaire se découpe en plusieurs phases (estimation de la demande, détermination du réseau, planification des départs, affectation des trains et du personnel). Nous nous intéressons dans une première partie à la phase de planification des départs des trains sur un intervalle temporel, appliquée au cas spécifique d'Eurotunnel. L'objectif est de calculer les horaires des départs des trains depuis chacune des deux stations (Coquelles en France et Folkestone en Angleterre) en respectant des contraintes d'exploitation (sécurité, chargement, ...) et des accords commerciaux signés avec leurs partenaires (Eurostar, ...). De plus, la prise en compte des retards dès la planification des départs est primordiale pour limiter la propagation des perturbations de train en train sur le réseau. Nous avons développé des algorithmes de planification pour Eurotunnel tenant compte des contraintes du réseau et de la probabilité de retard pour chaque train. Ces algorithmes utilisent des outils standard de la Recherche Opérationnelle pour modéliser et résoudre ces problèmes d'optimisation.

La tarification des billets est un enjeu majeur pour les entreprises de transport. Pour les compagnies aériennes, de nombreux algorithmes ont été étudiés pour définir le prix optimal des billets pour différentes classes de passagers. Nous appliquons dans une deuxième partie des méthodes standard de tarification (modèles de choix discrets) afin d'optimiser de manière globale les prix et les horaires des départs pour des entreprises de transport ferroviaire. Des outils classiques de l'optimisation stochastique, des modèles de choix discrets et des heuristiques sont utilisés dans nos algorithmes pour donner les meilleures solutions possibles en un temps de calcul limité.

Nous nous intéressons dans une dernière partie à une classe de problèmes de transport, inspirés de ceux rencontrés par Eurotunnel, en donnant des algorithmes efficaces de résolution exacte ou approchée. Ces algorithmes permettent de donner une borne supérieure de la complexité temporelle de ces problèmes. La classe de problèmes étudiés consiste en la planification des départs de navettes sur une ligne fixe, pour transporter d'une station A vers une station B des usagers arrivant de manière continue. Les navettes sont éventuellement autorisées à faire de multiples rotations pour transporter plusieurs vagues d'usagers. L'objectif est de limiter le temps d'attente des passagers avant le départ de leur navette. Des combinaisons originales de l'optimisation convexe et de la théorie des graphes (problèmes de plus court chemin) sont utilisées dans nos algorithmes.

Mots clés : transport ferroviaire, grilles horaires, retards, optimisation stochastique, tarification, heuristiques, complexité algorithmique, optimisation convexe, plus court chemin.

Contents

List of Figures

List of Tables

1 Introduction

Three research topics of rail transportation are developed along this thesis. They were conducted with Eurotunnel, the company which operates the tunnel under the Channel, throughout an industrial partnership with the École des Ponts ParisTech. The aim is to improve the scientific knowledge on rail transportation. The objective is to develop methods to operate the Channel tunnel in an optimal way. The principal issue is to face the future significant increase of the freight and passengers demand. The first topic was conducted with a strong relationship with Eurotunnel. It deals with the computation of optimal schedules of departures which satisfy the commercial and operational constraints of operating the Channel tunnel. We use standard techniques of Operations Research to model and solve this optimization problem.

The second topic is more forward-looking and deals with a joint scheduling and pricing problem for a virtual transportation company. The objective is to compute optimal schedules of shuttle departures and to fix the price of the trips for customers belonging to different classes. We model this problem and propose various methods to compute good feasible solutions in a reasonable computation time.

The third research topic deals with a class of transportation problems inspired by Eurotunnel. We give efficient algorithms to solve exactly or approximately these problems, and estimate their running time. The algorithms are based on standard methods from Operations Research such as the shortest path computation.

The thesis has three main parts and each one develops one of these topics. The contributions are:

- A simple and efficient algorithm to compute feasible and optimized schedules for Eurotunnel.
- A method to deal with the delays for Eurotunnel.
- A proof that optimizing in a joint way the departures and the prices leads to an increase of the revenue.
- Algorithms to compute such departures and prices.
- Theoretical results and algorithms for transportation problems inspired by Euro-

tunnel.

Some of the work on the problem of scheduling and pricing was done at EPFL during a visit to Michel Bierlaire.

1.1 Cyclic scheduling at Eurotunnel

The planning process to compute schedules of departures in rail transportation can be divided into five successive phases, given in Figure 1.1.

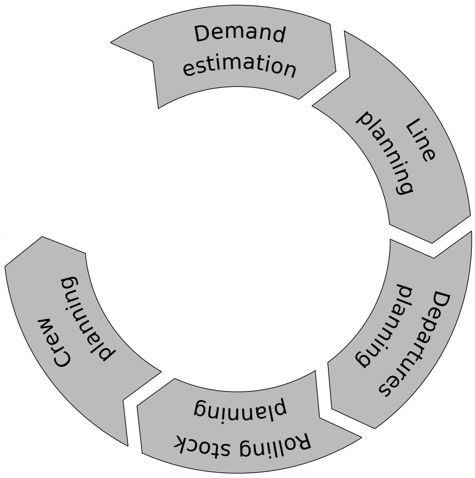

Figure 1.1 – Planning process in rail transportation

The demand estimation consists in estimating the number of passengers who want to travel from an origin to a destination on a time period, either in an aggregated way, or in a continuous way. The aggregated information is given in an Origin-Destination matrix. The line planning is generally done in a second phase. The aim is to decide the future transportation lines and connexions in order to face the estimated demand. The objective is to limit the operational costs and to minimize the number of connexions for the passengers. The scheduling phase comes next and it is about figuring out the departure times of the trains for each line which satisfy the operation constraints on the rail network. The objective is to minimize the trip time of the passengers or the connexion time between the lines in order to maximize their comfort. This phase is precisely studied in the first part of the thesis, in the case of Eurotunnel. Once the schedule is computed, it must be decided which train will cover each scheduled departure. The rolling stock planning has to satisfy several constraints (number of trains, capacity, engine power, maintenance, ...). Finally, on-board crew and dock crew have to be assigned to each train. As for airline transportation, the crew plan needs to satisfy several labour rules. The objective is to compute the crew plan which minimizes the costs (hotels for the crew, overtime, ...) and maximizes the crew satisfaction. Often, computing a feasible plan a complex task because of the huge combinatorics of the problem.

The planning process is usually done sequentially because of the algorithmic complexity of all the phases. A usual way to keep computation times acceptable and to improve the process is to create a feedback loop between the several planning phases (see Figure 1.1) and to iterate several times the process taking into account the previous phases.

In the particular case of Eurotunnel, the rail network is only composed of two stations with a line track for each direction. This rail track is one of the world's busiest commercial rail track, with Eurostars connecting London to the major European capitals, freight trains, and passengers and freight shuttles operated by Eurotunnel. As we already explained, we focus on the scheduling of the departure times in the first part. The objective is first to maximize the number of shuttles in a schedule, while satisfying operation constraints in the Channel tunnel and commercial constraints, with Eurostar mainly. This issue is presented in a first chapter and is dealt with a standard tool in Operations Research: *Mixed Integer Linear Programming*.

It considers optimization problems with a particular form. These problems are described by a linear objective function, linear constraints and integer variables. The canonical form of this class of problems is expressed as

$$\begin{array}{lrcl} \text{Minimize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & & \\ \text{Subject to} & \boldsymbol{A}\boldsymbol{x} & \leq & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \boldsymbol{0} \\ & \boldsymbol{x} & \in & \mathbb{Z}^n, \end{array}$$

where $\boldsymbol{b}$ and $\boldsymbol{c}$ are real vectors and $\boldsymbol{A}$ is a real matrix. Various resolution techniques are available for this class of problems and many different commercial solvers gather most of these techniques to compute optimal solutions in an efficient way. In the case of Eurotunnel, we get optimized schedules of departures on a given time period.

Our algorithms give the maximum number of freight shuttles that can be scheduled in a given period, with a fixed number of Eurostars, freight trains and passengers shuttles. For instance, for a one-hour cyclic schedule with 4 Eurostars, one freight train, and 5 passengers shuttles, the maximum number of freight shuttles that can be scheduled is 4 and an example of such a schedule is given in Figure 1.2. Moreover, we also show by our algorithms that relaxing some operational or commercial constraints may improve significantly the number of freight shuttles in the schedule. For example, relaxing a constraint on the loading platforms improves from 8 to 10 the maximum number of freight shuttles in a schedule with 2 Eurostars and 2 passengers shuttles.

The second chapter deals with the computation of schedules of departures minimizing the propagation of delays. It relies on *Stochastic Optimization* which studies optimization problems that involve random objective functions or random constraints. Indeed, randomness is everywhere and can occur at any time which leads to random delays on the schedule. These delays may propagate train after train and the objective is to compute feasible schedules minimizing the average total delay of each train. We model this

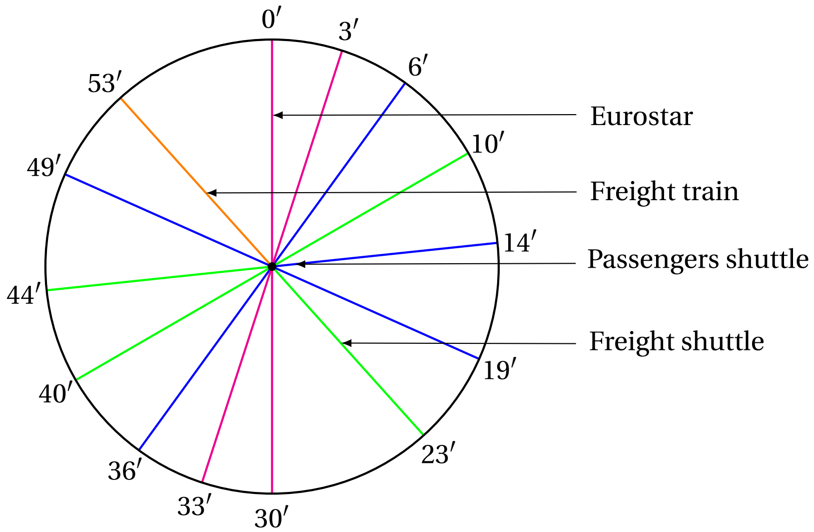

Figure 1.2 – An example of such a schedule

problem by a *recourse stochastic program* whose general formulation is of the form

$$\begin{array}{lrcl} \text{Minimize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} + \mathbb{E}_{\boldsymbol{\xi}}[Q(\boldsymbol{x},\boldsymbol{\xi})] & & \\ \text{Subject to} & \boldsymbol{A}\boldsymbol{x} & \leq & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \boldsymbol{0}, \end{array}$$

where $\mathbb{E}_{\boldsymbol{\xi}}[\cdot]$ is the expected value over $\boldsymbol{\xi}$ and $Q(\boldsymbol{x},\boldsymbol{\xi})$ is a recourse function.

The variables $\boldsymbol{x}$ are the first stage variables, $\boldsymbol{b}$ and $\boldsymbol{c}$ are deterministic real vectors and $\boldsymbol{A}$ a deterministic real matrix. The parameter $\boldsymbol{\xi}$ contains all the randomness in the second stage. In this problem, the variables $\boldsymbol{x}$ have to be optimized in a first stage before the randomness $\boldsymbol{\xi}$ has been revealed. These problems are much more difficult to solve that the deterministic ones. A method applied in the thesis to solve such problems consists in sampling the randomness $\boldsymbol{\xi}$ with scenarios and optimizing the problem on the scenarios with an estimator on the expected value. Under specific conditions, solving the problem with a large number of samples gives a very good approximation of the recourse function.

In this chapter, we compute schedules of departures taking into account the probability of a delay for each train in order to minimize their propagation. We propose algorithms for two versions of the problem, whether the order of the trains is fixed in input or not. The results show a significant improvement: for instance, for a schedule with 2 Eurostars, one freight train, 4 passengers shuttles and 7 freight shuttles per hour, the average delay of the trains in the current schedule in around 7'20" whereas it is less than 6' with our algorithm. It represents an improvement of almost 20%.

1.2 Joint pricing and scheduling

On the one hand, computing optimized schedules of departure times for rail transportation has been studied a lot. On the other hand, pricing problems have also been studied a lot, mainly for airline companies and the hotel industry. Often called yield management or revenue management, it deals with the study of the prices and the capacity management issues of products to offer to different categories of customers. The objective of such problems is to maximize the revenue of the company. The problem of computing schedules and fixing the prices is usually done sequentially: first an optimized schedule of departures is computed, and then a price and capacity policy is applied for each category of customers.

However, the customers willing to purchase a ticket for a trip are sensitive both to the departure times and the prices. This is the reason why the study of a joint scheduling and pricing problem could lead to higher revenues for the companies and a better customers' satisfaction. This is already the case for airline companies which combine these two subjects. To measure the attractiveness of a product for a customer, we use *discrete choice models*, known and developed originally for marketing. They are used when one wants to model and understand the customers' behaviour. They allow to identify key parameters among several ones and to quantify their influence on the customer's choices. This can be useful to optimize the strategic decisions of a company. Random behaviours can also be modelled this way. In our case, the price and the departure time of a train are key parameters in the choice of a customer. We model the problem by a stochastic recourse program and we sample the problem with scenarios of randomness as it is described in the previous section.

Moreover, heuristic algorithms are also presented in order to improve locally the solutions. In particular, we use a *Lagrangian heuristic* and a *Gauss-Seidel algorithm*, which are standard algorithms in Operations Research and Optimization.

With these techniques, we are able to compute good solutions for our joint scheduling and pricing problem. We propose two algorithms to compute the departures times and the prices and we compare the results with the natural sequential approach of first scheduling the departures, and then fixing the prices. Surprisingly, the sequential approach gives good results, but we are able with our algorithms to compute better feasible solutions. For instance, we get the following result. Consider 50 customers, belonging to three distinct economic classes with various sensitiveness to the prices and different preferred departure time. These customers want to buy a ticket for a trip, and the operator has 3 shuttles to schedule. The feasible solution computed with the sequential algorithm provides a revenue of 574, and our algorithms provide a revenue of 599. The improvement is almost 5% in this case.

Moreover, we also propose algorithms to compute upper bounds on the optimal value. One algorithm is based on *Lagrangian relaxation*. This standard technique of Operations Research consists in relaxing some "difficult" constraints of the problem and to set a penalization if they are not satisfied. We are thus able to assess the quality of the solutions.

1.3 Theoretical transportation problems

The third part of the thesis gives algorithms to solve a class of rail transportation problems. This class consists in scheduling the departure times of a fleet of S shuttles to transport passengers arriving continuously according to a cumulative demand $D(\cdot)$, on a fixed trip. The shuttles have a capacity C and are potentially allowed to perform several rotations and come back after a return time π. The loading speed of the passengers in the shuttles is ν. The objective is to minimize the maximum waiting time or the average waiting time of the passengers before the depart of their shuttle. This class of problems is inspired by our partnership with Eurotunnel but similar problems can also be found in chemistry and computer science, for example. The results are polynomial algorithms which compute optimal solutions or feasible solutions approximating the optimal value of the problem. Whereas the usual methods for such problems are mainly based on linear programming, we developed original methods dealing with convex optimization and shortest path computation.

We also give the complexity of the algorithms which is an important deal in the study of decision problems. It consists in determining the number of elementary operations performed during the execution of the algorithm. It is commonly expressed using the expression $O(\cdot)$ depending on the amount of information needed to encode the input parameters of the algorithm.

Consider the version in which the shuttles are not allowed to perform several rotations. We have the following result:

Theorem 1. *Assume that the demand $D(\cdot)$ is a step function defined with K discontinuities, supposed to be part of the input. Suppose moreover that the loading speed ν is equal to 0. Then one can compute a schedule minimizing the average waiting time in $O(K^2S)$. The same holds for problem minimizing the maximum waiting time.*

We also get another result regarding the case where all passengers are present from the beginning in terminal and where we have a single shuttle. To show the relevance of our result in other contexts, let us reformulate the problem. A given quantity D of some product is in a warehouse and has to be carried using a single shuttle. This shuttle is allowed to perform several rotations: when the shuttle leaves, it comes back after a return time $\pi > 0$. The shuttle has a limited capacity $C \geq 0$ and the loading speed of the product in the shuttle is $\nu \geq 0$. The objective is to schedule the departures d_j of the shuttle and the quantity of product x_j in each shuttle, in order to minimize the holding costs of the product in the warehouse. The naive strategy of loading the shuttle completely for each departure is not optimal. A tricky feature of the problem consists in showing that there exists an optimal solution and that it requires a finite number of trips.

Theorem 2. *The problem has an optimal solution and it is necessarily of the form*

$$
\begin{aligned}
x_0 &= 0 \\
x_j &= \begin{cases} C & \textit{if } j \le a, \\ \dfrac{D - aC}{\theta(a) - a} + \dfrac{\pi}{\nu}\left(\dfrac{a + \theta(a) + 1}{2} - j\right) & \textit{if } a + 1 \le j \le \theta(a), \\ 0 & \textit{otherwise,} \end{cases} \\
d_j &= \nu \sum_{i=1}^{j} x_i + (j-1)\pi
\end{aligned}
$$

with $a \in \mathbb{Z}_+$ *such that* $a \le \frac{D}{C}$ *and where*

$$
\theta(a) = a + \left\lceil \frac{-1 + \sqrt{1 + \frac{8\nu}{\pi}(D - aC)}}{2} \right\rceil .
$$

This theorem allows to compute very quickly the optimal solution by checking the $\lceil D/C \rceil + 1$ possible solutions.

1 Introduction (version française)

Trois thématiques de recherche liées au transport ferroviaire sont présentées dans cette thèse. Elles ont été réalisées en partenariat avec Eurotunnel, l'entreprise qui exploite le tunnel sous la Manche, via une chaire scientifique avec l'École Nationale des Ponts et Chaussées. Cette chaire a été créée dans le but d'améliorer les connaissances dans le domaine du transport ferroviaire et la thèse développée ici s'est effectuée dans ce cadre. L'objectif est la gestion optimisée du tunnel sous la Manche exploité par Eurotunnel et la principale problématique est d'anticiper la hausse significative de la demande de transport de fret et de passagers dans un futur proche. La première thématique a nécessité un dialogue soutenu avec Eurotunnel et l'objet est la création optimisée de grilles horaires satisfaisant les contraintes opérationnelles et commerciales liées à l'exploitation du tunnel sous la Manche. Des méthodes standard de la Recherche Opérationnelle sont utilisées pour modéliser et résoudre ce problème d'optimisation.

La deuxième thématique est plus prospective et consiste en un problème de planification et tarification jointes pour un opérateur de transport. L'objectif est d'établir une grille horaire de départs de navettes et de fixer le prix chaque navette pour un ensemble d'usagers appartenant à différentes classe économiques. Nous avons modélisé ce problème et donné des méthodes variées de résolution dans le but de produire de bonnes solutions en un temps de calcul raisonnable.

La troisième thématique de recherche traite d'une classe de problèmes de transport inspirés de ceux rencontrés par Eurotunnel. Elle présente des algorithmes efficaces de résolution ou d'approximation et donnent une estimation leur temps de calcul. Ils utilisent des méthodes originales de la Recherche Opérationnelle comme le calcul de plus courts chemins.

La thèse comporte trois parties, chacune développant une des thématiques. Les contributions sont:

- Un algorithme simple et efficace pour calculer des grilles horaires réalisables et optimisées pour Eurotunnel.
- Une méthode pour prendre en compte les retards pour Eurotunnel.

- Une démonstration qu'une optimisation jointe des départs et des prix mène à une augmentation des revenues.
- Des algorithmes pour calculer ces départs et ces prix.
- Des résultats théoriques et des algorithmes pour des problèmes de transport inspirés de ceux rencontrés par Eurotunnel.

Une partie du travail du problème de planification et tarification a été menée à l'EPFL durant un séjour avec l'équipe de Michel Bierlaire.

1.1 Grilles horaires chez Eurotunnel

Le processus de création de grilles horaires pour un opérateur de transport ferroviaire peut se découper en cinq décisions successives, illustrées dans la Figure 1.1.

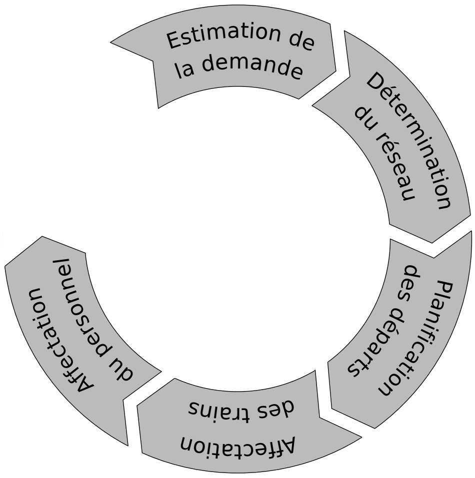

Figure 1.1 – Processus de création de grilles horaires en transport ferroviaire

L'estimation de la demande consiste à estimer le nombre d'usagers voulant voyager d'une station origine à une station destination, soit de manière agrégée, soit de manière continue, sur un intervalle temporel. Les informations agrégées sont généralement rassemblées dans une matrice pour chaque paire Origine-Destination. La détermination des lignes du réseau est généralement réalisée dans un deuxième temps. Il s'agit de décider des futures lignes de transport et de leurs connexions afin de répondre à la demande de déplacement des usagers, l'objectif étant de contrôler les coûts d'exploitation et de minimiser le nombre de connexions pour les usagers. La phase de planification des départs intervient ensuite et le but est de déterminer les horaires de départs des trains pour chaque ligne en respectant des contraintes d'exploitation du réseau. L'objectif est ici de minimiser les temps de transport moyen des usagers ou les temps de connexion entre deux trains afin de maximiser le confort des usagers. Nous nous intéressons à cette phase dans la première partie de la thèse, dans le cas spécifique d'Eurotunnel. Une fois les horaires des départs planifiés, l'affectation des trains pour chaque départ doit avoir lieu en respectant les contraintes de flotte (nombre, capacité, puissance, maintenance ...). Toutes ces contraintes doivent être prises en compte lors de la phase d'affection. Enfin, du personnel de bord (conducteur, contrôleurs, ...) et du personnel à quai doivent être affectés pour chaque train. À l'image du transport aérien, les horaires de travail du personnel sont très encadrés et soumis à de nombreuses réglementations. L'objectif de cette phase est de créer l'emploi de temps du chaque agent afin de minimiser les frais liés au découchage du personnel et de maximiser leur satisfaction (respect des demandes de congés, demandes pour certaines lignes, ...). Le simple fait de générer un emploi du temps respectant toutes les réglementations et affectant le personnel requis pour chaque train est souvent déjà un problème difficile.

Ce processus de création de grilles horaires est effectué de manière séquentielle dans la plupart des cas à cause de la complexité algorithmique de chacune des phases. Une méthode standard pour garder des temps de calcul acceptables est de boucler le processus (voir Figure 1.1) et de recommencer plusieurs fois la boucle en tenant compte des étapes précédentes.

Dans le cas particulier d'Eurotunnel, le réseau ferroviaire se résume à deux stations reliées par une voie unique dans chaque sens. Ce réseau est un des plus empruntés au monde avec des Eurostars reliant Londres aux grandes capitales européennes, des trains de marchandises et des navettes de passagers et de frets exploitées par Eurotunnel. Comme nous l'avons dit, nous nous concentrons exclusivement à la phase de planification des départs dans la première partie. L'objectif est dans un premier temps de maximiser le nombre de départs de navettes dans une grille horaire satisfaisant des contraintes opérationnelles fortes liées au transport ferroviaire en tunnel, et des contraintes commerciales notamment avec Eurostar. Cette problématique est présentée dans un premier chapitre et est abordée avec un outil standard de la Recherche Opérationnelle : la *programmation linéaire en nombres entiers (PLNE)*.

Cette branche de la Recherche Opérationnelle considère des problèmes d'optimisation ayant une forme particulière. Ces problèmes sont décrits par une fonction objectif (ou fonction de coût) linéaire, des contraintes linéaires et des variables entières. La forme canonique de cette classe de problèmes est la suivante :

$$\begin{array}{rrcl} \text{Minimiser} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} & & \\ \text{Sous les contraintes} & \boldsymbol{A}\boldsymbol{x} & \leq & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \boldsymbol{0} \\ & \boldsymbol{x} & \in & \mathbb{Z}^n, \end{array}$$

où $\boldsymbol{A}$ est une matrice réelle, et $\boldsymbol{b}$ et $\boldsymbol{c}$ sont des vecteurs réels. De nombreuses techniques de résolution sont connues pour ces types de problèmes et un grand nombre de solvers commerciaux rassemblent la plupart de ces méthodes pour calculer les solutions optimales de manière efficace. Nous obtenons dans le cas d'Eurotunnel des grilles horaires optimisées planifiant les départs des trains sur une période et satisfaisant les contraintes spécifiques à un réseau en tunnel.

Nos algorithmes donnent le nombre maximal de navettes de fret pouvant être planifiées sur une période donnée, avec un nombre fixé d'Eurostars, de trains de marchandises et de navettes de passagers. Par exemple, pour une grille horaire cyclique d'une heure avec 4 Eurostars, un train de marchandises et 5 navettes de passagers, le nombre maximal de navettes de fret est 4 et un exemple d'une telle grille horaire est donné en Figure 1.2. De plus, nous montrons également avec nos algorithmes que la relaxation de certaines contraintes opérationnelles ou commerciales peuvent améliorer de manière significative le nombre de navettes de fret dans la grille horaire. Par exemple, relâcher une contrainte liée au terminal de chargement améliore le nombre maximal de navettes de fret de 8 à 10, pour une grille horaire avec 2 Eurostars et 2 navettes de passagers.

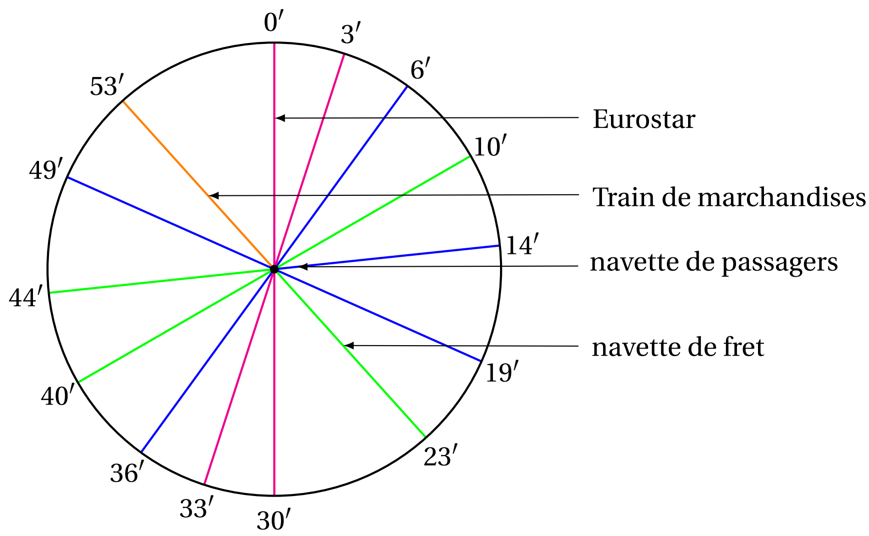

Figure 1.2 – Un exemple de grille horaire

Un second chapitre étudie la création de grilles horaires minimisant la propagation des retards. Ce chapitre rentre dans le cadre de l'*Optimisation Stochastique* qui étudie des problèmes d'optimisation avec des objectifs aléatoires et des contraintes aléatoires. En effet, l'aléa est présent partout et peut intervenir à tout moment ce qui implique des retards sur les départs dans la grille horaire. Ces retards peuvent se propager de train en train et l'objectif est de créer une grille horaire valide minimisant le retard moyen des trains. Nous modélisons ce problème par un *programme stochastique de recours* dont la forme générale est la suivante :

$$\begin{array}{rrcl} \text{Minimiser} & c^{\mathrm{T}}\boldsymbol{x} + \mathbb{E}_{\boldsymbol{\xi}}[Q(\boldsymbol{x},\boldsymbol{\xi})] & & \\ \text{Sous les contraintes} & \boldsymbol{A}\boldsymbol{x} & \leq & \boldsymbol{b} \\ & \boldsymbol{x} & \geq & \mathbf{0}, \end{array}$$

où $\mathbb{E}_{\boldsymbol{\xi}}[\cdot]$ est l'espérance mathématique par rapport à $\boldsymbol{\xi}$ et $Q(\boldsymbol{x}, \boldsymbol{\xi})$ est une fonction de recours.

Les variables $\boldsymbol{x}$ sont les variables de la première étape, $\boldsymbol{b}$ et $\boldsymbol{c}$ sont des vecteurs réels déterministes et $\boldsymbol{A}$ une matrice réelle déterministe. Le paramètre $\boldsymbol{\xi}$ contient les informations sur l'aléa dans le second niveau. Dans ce problème, les variables x doivent être optimisées dans un premier temps avant d'avoir l'information sur l'aléa. Ces problèmes sont beaucoup plus difficiles à résoudre que les problèmes déterministes. Une méthode employée dans la thèse consiste tirer au hasard un grand nombre de scénarios de l'aléa $\boldsymbol{\xi}$ et d'optimiser sur ces scénarios en donnant un estimateur de l'espérance mathématique. Sous certaines conditions, résoudre le problème avec un grand nombre de scénarios donc une très bonne approximation de la fonction de recours.

Nous calculons ici des grilles horaires réalisables qui prennent aussi en compte les probabilités de retards de chaque train pour en minimiser la propagation dans la grille. Nous proposons des algorithmes pour deux versions de ce problème, l'une pour laquelle l'ordre des trains est fixé dans l'input, l'ordre pour laquelle il est libre. Les résultats montrent une amélioration significative : par exemple, pour une grille horaire avec 2 Eurostars, un train de marchandises, 4 navettes de passagers et 7 navettes de fret par heure, le retard moyen des trains dans une grille horaire courante est d'environ 7'20" alors qu'il est inférieure à 6' avec nos algorithmes. Cela représente un gain de presque 20%.

1.2 Tarification et planification jointes

D'un côté, la création de grilles horaires optimisées pour le transport est un problème très étudié. De l'autre côté, les problèmes de tarification sont aussi très bien connus, développés notamment pour les compagnies aériennes ou l'industrie hôtelière. Souvent appelé "yield management", de l'anglais *yield* : rendement, ou bien "revenue management", il s'agit de l'étude des prix et de la gestion capacitaire de biens à offrir à différentes classes d'usagers. L'objectif de ce problème étant la maximisation du chiffre d'affaires. Les problèmes de création de grilles horaires et de tarification sont généralement étudiés de manière séquentielle : d'abord une grille horaire de départs optimisée est générée, puis une politique tarifaire et capacitaire y est appliquée donnant le prix de chaque départ, pour chaque classe d'usagers.

Cependant, les usagers voulant acheter un titre de transport sont sensibles de manière combinée aux horaires de départs des trains et à leur prix. C'est pourquoi une étudie jointe de la planification et de la tarification pourrait mener à des revenues plus importants pour les opérateurs et à une satisfaction supérieure pour les usagers. Ceci est déjà le cas pour les compagnies aériennes qui combinent de plus en plus ces deux aspects. Pour mesurer l'attrait d'un client vis-à-vis d'un produit, on utilise les *modèles de choix discrets*, connus et développés notamment en marketing. Ceux-ci sont utilisés lorsque l'on veut modéliser et comprendre le comportement des usagers. Ils permettent d'identifier les paramètres décisifs parmi un panel de choix proposés et de quantifier leur influence, ce qui peut être utile pour optimiser les décisions stratégiques d'une compagnie. Les événements et comportements aléatoires peuvent aussi être modélisés par ce biais. Dans notre cas, le prix et l'horaire de départ d'un train sont des paramètres décisifs dans le choix des usagers. Nous modélisons ce problème par un programme stochastique de recours et nous échantillonnons le problème avec des scénarios de l'aléa comme dans la section précédente.

De plus, des algorithmes heuristiques sont aussi présentés afin d'améliorer la qualité des solutions de manière locale. Ces algorithmes sont notamment basés sur des *heuristiques lagrangiennes* et des algorithmes de type *Gauss-Seidel*, qui sont des algorithmes standard de la Recherche Opérationnelle et de l'Optimisation.

Avec ces techniques, nous sommes capables de calculer de bonne solutions pour notre problème de planification et tarification jointes. Nous proposons deux algorithmes pour calculer les départs et les prix, et nous comparons les résultats avec une approche naturelle séquentielle qui consiste à planifier les horaires de départs dans un premier temps, et de fixer les prix dans un second temps. De manière surprenante, l'approche séquentielle donne de bons résultats, mais nous sommes capables avec nos algorithmes de calculer de meilleures solutions. Nous obtenons, par exemple, le résultat suivant. Considérons 50 clients, appartenant à trois classes économiques distinctes, ayant des sensibilités aux prix et des heures de départs privilégiées différentes. Ces clients veulent acheter un ticket pour un destination et l'opérateur de transport veut planifier 3 navettes. La solution calculée avec l'approche séquentielle donne un revenu de 574, alors que nos algorithmes donnent un revenu de 599. Le gain est d'environ 5% dans ce cas.

De plus, nous proposons des algorithmes pour calculer des bornes supérieures sur la valeur optimale. Ces algorithmes sont basés sur la *relaxation lagrangiennes*. Cette technique standard de la Recherche Opérationnelle consiste à relâcher des contraintes ”difficiles” du problème et à instaurer une pénalisation si elles ne sont pas satisfaites. Nous pouvons ainsi apprécier la qualité des solutions réalisables que nous trouvons.

1.3 Problèmes théoriques de transport

La troisième partie de la thèse donne des algorithmes de résolution d'une certaine classe de problèmes liés au transport ferroviaire. Cette classe de problèmes consiste en la planification des départs d'une flotte de S navettes sur un trajet donné. des usagers arrivent de manière continue suivant une demande cumulée $D(\cdot)$. Les navettes ont une capacité C et sont éventuellement autorisées à faire de multiples rotations et reviennent après un temps π. La vitesse de chargement des usagers dans les navettes est ν. L'objectif est de minimiser le temps d'attente maximal ou le temps d'attente moyen des usagers avant le départ de leur navette. Cette classe de problèmes est inspirée de notre partenaire Eurotunnel, mais on peut retrouver des problèmes similaires dans des domaines variés comme l'informatique ou la chimie. Les résultats sont des algorithmes polynomiaux donnant une solution optimale ou approchée du problème. Alors que les méthodes de résolution pour ces types de problèmes sont généralement basées sur de la programmation linéaire, nous avons développé dans cette thèse des méthodes originales de résolution qui font notamment appel à l'optimisation convexe calcul de plus court chemin.

De plus, nous donnons la complexité de ces algorithmes, qui est une donnée importante lors de l'étude algorithmique de problèmes de décisions. Elle consiste en la recherche du nombre d'instructions de base nécessaires lors de l'exécution de l'algorithme et utilise la notion de domination $O(\cdot)$ en fonction de la quantité d'informations nécessaire pour encoder les paramètres d'entrées de l'algorithme.

Considérons la version du problème pour laquelle les navettes ne font pas plusieurs rotations. Nous avons le résultat suivant.

Théorème 1. *Supposons que la demande $D(\cdot)$ soit une fonction constante par morceaux définie par K discontinuités faisant partie de l'input. Supposons aussi que la vitesse de chargement v soit nulle. Alors il existe un algorithme calculant une solution minimisant le temps d'attente moyen des usagers en $O(K^2 S)$. Il en est de même le problème minimisant le temps d'attente maximal.*

Nous avons également un résultat pour le cas où tous les usagers sont présents au début sur le terminal, et où nous ne disposons que d'une seule navette. Pour montrer la pertinence du résultat dans d'autres contextes, nous reformulons le problème. Une quantité D de produit est stockée dans un entrepôt et doit être transportée en utilisant une navette. Cette navette peut faire plusieurs rotations : quand elle quitte le terminal, elle revient après un temps $\pi > 0$. La navette à une capacité $C \geq 0$ et la vitesse de chargement du produit dans la navette est $v \geq 0$. L'objectif est planifier les départs d_j de la navette et la quantité de produit x_j transportée à chaque départ, afin de minimiser les coûts de stockage du produit dans l'entrepôt. La stratégie naïve consistant à remplir les navettes à chaque départ n'est pas optimale. Un point clé du problème consiste à montrer qu'il existe une solution optimale et que celle-ci nécessite un nombre fini de départs.

Théorème 2. *Le problème possède une solution optimale et elle est nécessairement de la forme*

$$x_0 = 0$$

$$x_j = \begin{cases} C & \textit{si } j \leq a, \\ \dfrac{D - aC}{\theta(a) - a} + \dfrac{\pi}{v}\left(\dfrac{a + \theta(a) + 1}{2} - j\right) & \textit{if } a + 1 \leq j \leq \theta(a), \\ 0 & \textit{sinon,} \end{cases}$$

$$d_j = v \sum_{i=1}^{j} x_i + (j-1)\pi$$

avec $a \in \mathbb{Z}_+$ tel que $a \leq \frac{D}{C}$ et où

$$\theta(a) = a + \left\lceil \frac{-1 + \sqrt{1 + \frac{8v}{\pi}(D - aC)}}{2} \right\rceil.$$

Ce théorème permet de calculer très rapidement une solution optimale en testant les $\lceil D/C \rceil + 1$ solutions possibles.

2 The company Eurotunnel

2.1 History

Eurotunnel is a French and British company, which operates the Channel Tunnel, between France and the United Kingdom. It has its own shuttle services for freight and passengers, and earns revenue on other trains, such as Eurostars, or other freight trains.

Before the Tunnel as we know it today, architects suggested a lot a different ideas. The first one was proposed by the French mining engineer Albert Mathieu-Favier in 1802 who put forward the first design for a cross-Channel fixed link based on a bored two-level tunnel: a paved one for horse-drawn carriages and another one for groundwater flows. It included an artificial island halfway across for changing horses. In 1803, Henry Mottray proposed a submerged tunnel made of prefabricated iron sections. In the mid 19th century, the advent of steam machines and the construction of the first railway lines in the United Kingdom led to the first proposal for a rail tunnel by Aimé Thomé de Gamond. His proposal was approved by Queen Victoria and Napoleon III and was presented in the World Exposition in Paris in 1867. The boring started on both sides in 1880 but was stopped by the English for security reasons.

The project as it is today was finally launched in 1973 at Chequers (England) by Edward Heath, British Prime Minister, and Georges Pompidou, French President, when a Franco-British Channel Tunnel Treaty was signed. But in 1975, Harold Wilson, British Minister, announced that the project was stopped and withdrawn for financial reasons and in particular because of the oil crisis. The project has been stopped until 1984 when the British and French Governments reached an agreement on a consultation process with private promoters for the construction and operation of the tunnel, without public funding. On January 20th 1986, François Mitterrand and Margaret Thatcher announced the cooperation of France and the United Kingdom, for the construction of a tunnel between the two countries. The company Eurotunnel was founded on August 13th, 1986. Its missions were to finance, build, and operate the Tunnel. The tunnel cost 80% more than expected and the construction took 6 years. It was officially opened by Queen Elizabeth II and François Mitterrand on May 6th, 1994.

But the financial situation degraded very fast, mainly because of the strong increase of the construction and operational costs. A freight truck caused a huge fire in the tunnel in 1995 which also worsen the financial situation. The value of the share was divided by more than 20 between 1986 and 2003! In 2006, the company was placed into bankruptcy protection by a French court for six months. In 2007, a restructuring plan was approved by shareholders and the company made its first net profit in 2008, despite the costs associated with traffic losses from September 2008 to February 2009, following another fire in the tunnel.

Today, the company's profit is increasing: over 12 million passengers travel in the tunnel every year and its revenue was more than 1 billion Euros in 2016. Since the beginning, more than 390 million passengers (around 6 times the British population), 350 tonnes of goods, and more than 70 million vehicles crossed the Channel through the Tunnel. Around 100,000 trains travel in the Tunnel per year.

2.2 Technical characteristics

2.2.1 The Tunnel

The Channel Tunnel (also referred as "The Chunnel") is the Tunnel with the longest undersea portion in the world, with a section of 37 km. Its total length is about 50 km, between Coquelles (near Calais in France) and Folkestone (near Dover in England).

The tunnel was bored into the geological chalk marl section, which has the advantages of being impermeable and strong. On the British side, the chalk marl runs along the entire length of the tunnel, while on the French side, it crosses several geological layers which complicated the boring. The Tunnel is about 40 m under the sea floor (about 107 m under the sea level at its lowest point).

The cross-section of the Channel Tunnel is illustrated on Figure 2.1. It is composed of two major railway tunnels (A), one for each direction, and one service tunnel. The two major tunnels have 7.6 m diameter, and are 30 m apart. They are connected to a 4.8 m diameter service tunnel (B), for special service vehicles, by cross-passages (C) every 375 m, and with piston relief ducts (D) connecting the rail tunnels at 250 m spacing. The service tunnel allows access to maintenance, emergency rescue teams and serves as a safe haven if passengers need to be evacuated. The air pressure in the service gallery is sightly higher than in the major tunnels, in order to preserve it from smoke in case of fire. Two undersea crossovers allow trains to pass from one tunnel to the other during night maintenance periods to isolate a section of the tunnel, which brings flexibility on the operations. Two fire stations are set at the crossovers. These stations throw water on a train in case of a fire and help contain the fire.

The walls inside the tunnels are covered with concrete blocks, linked with large joints, which gives elasticity to the Tunnel in case of earthquakes. Indeed, the North of France is a seismic zone, and since Great Britain is an island, the length of the Tunnel may vary of a few centimetres.

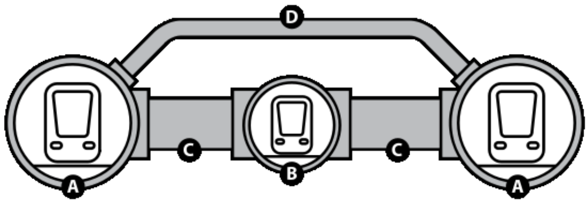

Figure 2.1 – Cross-section of the Tunnel

2.2.2 The terminals

On the French side, the Coquelles terminal, near Calais, is one of the biggest complexes in Europe. Covering 650 ha and having a 23 km long perimeter, it has the equivalent size of an international airport. It was built on a marshy soil, which had to be covered with a 50 cm thick layer of sand in order to ensure a good base for the foundations.

The Folkestone terminal is 8 km away from the Shakespeare Cliff. It covers about 150 ha, i.e. one third of the area of the French terminal.

Both terminals are loading and unloading points for vehicles travelling on the terminal. To travel through the Tunnel, all vehicles have to drive through tolls, and to pay fees depending on their size.

2.2.3 The different trains

Several types of trains may travel in the Tunnel. Eurotunnel has a shuttle service from Coquelles to Folkestone (CAFO) and from Folkestone to Calais (FOCA). These are the different trains travelling in the Tunnel:

- *HGV shuttles* which carry Heavy Gross Vehicles on open wagons, with a separate passenger wagon at the front of the train for the drivers. These shuttles travel at 140 km/h and are operated by Eurotunnel. They are Eurotunnel's major activity.
- *PAX shuttles* which carry passengers in closed wagons. Half of a typical shuttle (the rear rake) carries cars and other low vehicles in double-deck wagons. Coaches, buses and other high vehicles travel in the single-deck rake at the front of the shuttle. Passengers can stay in their vehicles or walk in their wagon during the trip, there is no special wagon for passengers is these shuttles. The PAX shuttles travel at 140 km/h as well and are also operated by Eurotunnel.
- *High-Speed Trains* (Eurostars) also travel through the tunnel. They are not operated by Eurotunnel itself, and they use the Tunnel in return for a fee. They travel at 160 km/h.
- *Freight trains* also use the Tunnel to travel through the Channel. They are not operated by Eurotunnel and are encouraged to travel at night, because the traffic is

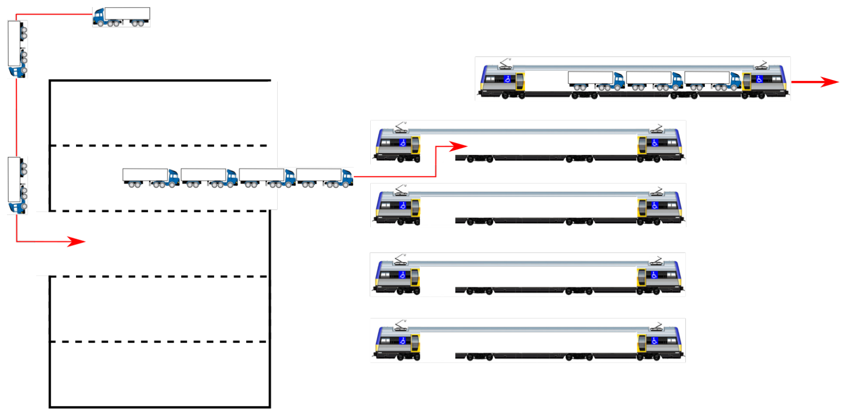

Figure 2.2 – The process of arrival, loading, and departure in the terminal.

lower. According to the types of goods they carry, these trains may travel at 100 km/h (MA100) or at 120 km/h (ME120).

2.2.4 The loading process

The loading process on the terminals has some specificities. There is no booking system for Heavy Gross Vehicles and their loading follows a FIFO policy. When a vehicle (usually a truck) arrives in the terminal, it drives through different tolls and enters a queue before loading in a shuttle. This queue closes when all vehicles which leave with the shuttle have arrived in the queue. The loading then starts in this shuttle and the next vehicles enter another queue. Vehicles can enter a new queue only if the previous one is closed.

The loading process is quite different for individual passengers. Passengers have to book in advance to travel in the Tunnel. When a passenger arrives, he waits until the operator calls him and enters a queue before loading in the shuttle. The same way as for HGV shuttles, the queue closes when all the passengers who will leave with the shuttle have arrived in the queue and then the loading starts.

The process is illustrated in Figure 2.2.

2.2.5 The Block Safety System

The Block Safety System is used to manage the Tunnel at an operation level. Each tunnel is divided into blocks of about 500 m long. The driver of a shuttle receives information on the required speed at the beginning of each block, depending on the speed and location of the other trains.

In the terminals and in the Tunnel, this information is given by the TVM (Transmission Voie-Machine, i.e. Track to Train Transmission) signalling. It is composed of two major systems:

- The rail system, which gives information on the location and on the speed of the other trains directly to the computer aboard the train.
- The system aboard which is autonomous and takes into account the information given by the rail to create orders that the driver has to follow. These orders are the maximum speed allowed for the train, at the end of the next block, or mandatory speed immediately requested.

The Rail Control Centre also ensures the security on the terminals and in the Tunnel. Controllers are in attendance 24 hours a day, ready to take manual control in the event of technical failure.

This block safety system allows to satisfy the security constraints between the different trains in the Tunnel. These constraints consist in a security headway between two consecutive trains at any point in the Tunnel.

2.3 Main objective

Eurotunnel's main objective is to face the future increase of the traffic in the Tunnel. This increase is due to three main phenomena:

- The natural growth of the traffic due to the global growth of exchanges between Paris and London, and between Brussels and London mainly. Based on the Gross Domestic Product (GDP) growth forecasts and the elasticity of rail traffic to the GDP growth, Eurotunnel expects a natural growth of 2.4% per year until 2020.
- The competitiveness of rail transportation compared to other means of transport. The ferry companies between France and the United Kingdom are collapsing and the major part of the freight traffic will turn towards Eurotunnel to cross the Chunnel.
- New direct lines and ease of transportation will also play a major role in the traffic increase. Eurotunnel expects 2.5 million additional passengers by 2020 generated by these new lines.

Part I

Cyclic schedules for trains and shuttles in the Tunnel

3 Scheduling at Eurotunnel

This chapter describes the problem of computing a schedule for all trains travelling in the Tunnel. The current method for computing such schedules at Eurotunnel is also described.

3.1 Objective

A natural way for Eurotunnel to face the increase of freight and passengers demand in the Tunnel is to compute schedules maximizing the number of freight and passengers shuttles. These objectives are conflicting, adding one HGV shuttle may decrease the number of PAX shuttles. Since the major increase will occur for freight, Eurotunnel focuses on HGV shuttles and maximizes their number in the schedule while the number of PAX shuttles is given as a constraint.

3.2 Rules to be satisfied by the schedules

This section describes the operational and commercial rules that the schedules have to satisfy.

3.2.1 Cyclic schedules

For operational reasons, the schedules are cyclic with a period of one hour. According to the variation of the demand, the global daily schedule is composed of repeated patterns of one-hour cyclic schedules.

3.2.2 Symmetric schedules

Since the fleet of shuttles is finite, the rotation of each shuttle has to be taken into account. To address this difficulty, Eurotunnel uses the same schedules for both directions France-United Kingdom and United Kingdom-France: the schedules are completely symmetric, the departures occur exactly at the same time for both stations. The interest of a symmetric schedule lies in regulating the number of shuttles on each side of the Tunnel at any time.

Moreover, combining this symmetry with the cyclic feature of the schedules also has a major benefit: a given shuttle leaves a same station exactly every two hours.

3.2.3 Discrete departure times

For operational reasons, Eurotunnel uses departure times on full minutes, or more rarely on half minutes. This is the case because a departure requires a lot of different people (driver, loading personnel, traffic regulator, ...) and it is impossible to guarantee a departure at a very precise instant. It is also easier for the customer to have a departure time on a full minute.

3.2.4 Security headways

A security headway between two trains has to be satisfied for security reasons. The block safety system (see Section 2.2.5) is used to guarantee this headway at any point in the Tunnel. The security headways are fixed by Eurotunnel.

Because of the differences between the speeds of the trains, the headway that has to be satisfied at the entrance gate of the Tunnel may be larger than the minimum security headway at any point of the Tunnel (see Figure 3.1). This figure shows *time-space diagrams* of 2 trains in the Tunnel. The x-axis represents the time and the y-axis represents the Tunnel. The lines represent the location of the trains in the Tunnel in according to the time.

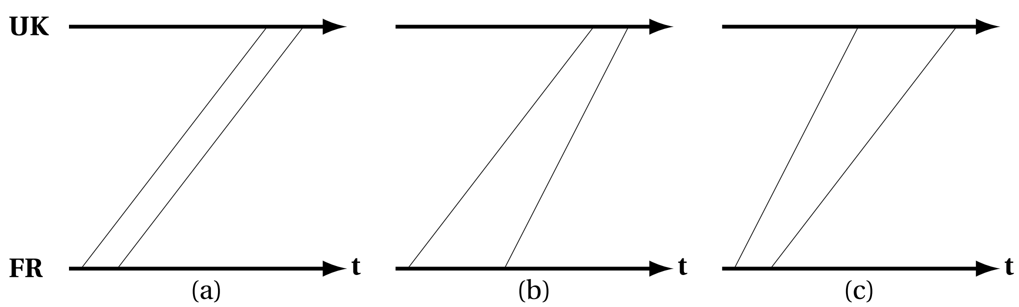

Figure 3.1 – Time-space diagrams representing the security headways

Computing these security headways for all ordered pairs of trains is also an important issue. Annexe A is devoted to the description of the mathematical model we propose to compute more accurate headways satisfying the security rules. They are given in Tables 3.1 and 3.2. The leading train is indicated in the first column and the following one in the first line.

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0’53”	0’48”	0’48”	0’48”	0’48”
PAX shuttle	6’22”	2’49”	2’49”	1’21”	1’21”
HGV shuttle	6’22”	2’49”	2’49”	1’21”	1’21”
MA100	12’21”	7’34”	7’34”	1’30”	5’59”
ME120	7’48”	3’02”	3’02”	1’15”	1’18”

Table 3.1 – Security headways for direction CAFO

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0’53”	0’31”	0’31”	0’32”	0’40”
PAX shuttle	6’07”	2’35”	2’35”	1’42”	1’50”
HGV shuttle	6’07”	2’35”	2’35”	1’42”	1’50”
MA100	13’02”	8’14”	8’14”	1’57”	6’43”
ME120	7’48”	3’02”	3’02”	1’15”	1’18”

Table 3.2 – Security headways for direction FOCA

Buffer times

Moreover, the schedules have to be designed in such a way that it limits the impact of the perturbations (delays, incidents, problems with a train, ...). To cope with this problem, a schedule is usually designed with buffer times between two consecutive train departures, and the travel times for all the trains are slightly over-evaluated.

Delays occur when a real-time operation (a departure for instance) is subject to an external or internal disturbance. There can be of two types: the first one is called *primary delay*. It corresponds to a delay directly implied by a disturbance. For instance, if there is an incident during the loading of a shuttle, the delay implied is a primary delay since it is directly caused by the incident. The second type is the *secondary delay*. It corresponds to a delay indirectly implied by a disturbance by propagation in the schedule. For instance, the previous incident on the loading of a shuttle may imply a delay on the following shuttle since its loading will start later. This delay is not implied by a disturbance on this shuttle, but on another train which impacts this shuttle. Adding buffer times between the departures cannot reduce the primary delays since they are directly implied by the disturbance, but may reduce the secondary delays.

To deal with the delays and design more robust timetables, arbitrarily buffer times are added to the security headway between any two ordered trains. For instance, if the security headway between two trains is theoretically 2’22”, Eurotunnel may add a 38-second buffer time in order to have a 3-minute minimum headway between these trains. These minimum headways contain the security headways and the buffer times. Adding these buffer times decreases the secondary delays. It is also a way for Eurotunnel to get minimum headways in full minutes which eases the design of their timetables. Moreover, they systematically add a 5-percent increase of the travel time of every train in order to prevent delays during the trip. Buffer times are useful to minimize the impact of a delay on the following trains, but they decrease the capacity of the network by allowing fewer trains to be scheduled.

The minimum headways used by Eurotunnel for both directions are given in Table 3.3.

	Eurostar	PAX	HGV	MA100	ME120
Eurostar	3'00"	2'30"	2'30"	5'00"	5'00"
PAX	7'00"	3'30"	3'30"	4'00"	4'00"
HGV	7'00"	3'30"	3'30"	4'00"	4'00"
MA100	15'00"	8'30"	8'30"	3'00"	7'00"
ME120	10'00"	4'00"	4'00"	5'00"	5'00"

Table 3.3 – Minimum headways used by Eurotunnel

3.2.5 Commercial agreements with Eurostar

Some commercial agreements between Eurotunnel and Eurostar constraint the scheduling. The agreements specify that for some periods of time in a day, the Eurostars have to be somehow evenly scheduled in the cyclic hour.

If there is an even number of Eurostars to schedule in the one-hour cyclic period, they have to be grouped by pairs of twin Eurostars whose departure times are separated by exactly 30 min. It means that if a Eurostar leaves a station at time t, another has to leave 30 min later at time $t+30$ (see Figure 3.2(a)). If there is an odd number of Eurostars to schedule, they are also grouped by pairs of twin Eurostars whose departure times are separated by exactly 30 min and there is thus a remaining Eurostar without any twin (see Figure 3.2(b)) whose departure time can be anywhere in the schedule. These agreements only apply during peak periods, mainly in the morning and in the evening.

Figure 3.2 represents feasible one-hour cyclic timetables satisfying the commercial agreements with Eurostar, for the case of 4 Eurostars (a) and 3 Eurostars (b). In both cases, the Eurostar represented with solid lines have their departure times separated by exactly 30 min. For case (b), the second dashed Eurostar is the remaining one.

3.2.6 Loading platforms

Because of a defect in the layout of the loading platforms, it is impossible to load three HGV shuttles or three PAX shuttles at the same time. Eurotunnel schedules at most two HGV shuttles in any 12-minute time window, and at most two PAX shuttles in any 12-minute time window.

Figure 3.3 describes this constraint for the HGV shuttles. After the first departure, there is a buffer time represented by the hatched zone of 12 min where only one additional

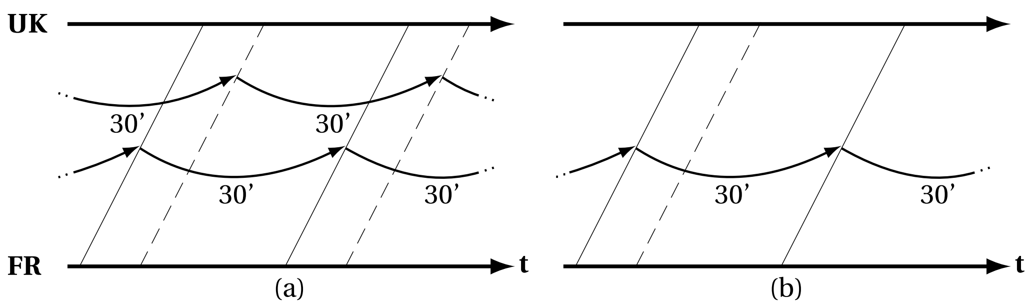

Figure 3.2 – Time-space diagrams representing the commercial agreement with Eurostar

departure can occur.

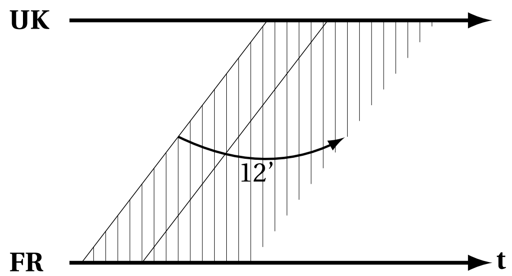

Figure 3.3 – Time-space diagram representing the platform constraint for the HGV shuttles

3.2.7 Equal distribution of PAX shuttles within the cycle

In order to guarantee a good quality of service to its passengers, Eurotunnel tries to schedule the PAX shuttles as evenly distributed as possible in the cyclic period. Thus, the time between two consecutive PAX shuttle departures must not be too long. This constraint is not clearly formalized by Eurotunnel and we propose the following rule: for any PAX shuttle, the following PAX shuttle has to depart at most $T/N^{\text{PAX}} \times 1.5$ min later. For instance, in a one-hour cyclic schedule with 2 PAX shuttles, their departures are separated by at most 45 min. In a schedule with 5 PAX shuttles, for any PAX shuttle departure, the next PAX shuttle has to depart at most 18 min later.

3.3 Current scheduling technique at Eurotunnel

The scheduling of trains and shuttles is currently done without any mathematical model at Eurotunnel. A few pre-calculated symmetric one-hour cycles are designed and combined into a global schedule for one day.

3.3.1 Standard train path and Tunnel capacity

A *train path* is the infrastructure capacity needed to run a train between two places over a given time period. The *capacity* of a tunnel is the maximum number of trains that can travel during a time lapse. The initial objective of Eurotunnel was to have 20 trains travelling through the tunnel per hour, and for each direction, with identical speeds, and spaced by 3 min. Indeed, 3 min was considered at first as the minimum headway between two shuttles.

A shuttle travels through the Tunnel at 140 km/h, and the total travel time, from gate to gate is around 26 min. Eurotunnel considers that a 3 min inter-time between 2 shuttles is needed for security reasons. Based on these considerations, Eurotunnel designed its own *standard train path*, corresponding to the train path of a shuttle through the Tunnel (26 min), and on an inter-time of 3 min between two trains. Only considering shuttles, the capacity of the Tunnel would be 20 shuttles per hour, i.e. 20 standard train paths per hour. These are shown on Figure 3.4. This time-space diagram would represent the *standard schedule* designed by Eurotunnel if there were only shuttles.

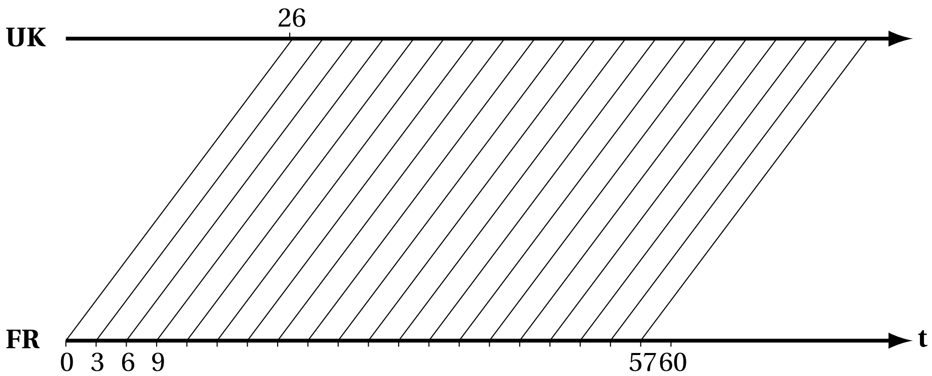

Figure 3.4 – Time-space diagram representing the standard schedule for the 20 standard train paths

3.3.2 Dealing with trains

When taking into account the different trains, the design of the timetable becomes harder because of the different speeds. To build the schedule, the train path corresponding to the different trains are inserted in the standard schedule. Since the speeds of the Eurostars and the shuttles are nonidentical, it crosses more than one standard train path in the total capacity of the Tunnel (see Figure 3.5).

For each train which travels through the Tunnel, the part of the total capacity used by the

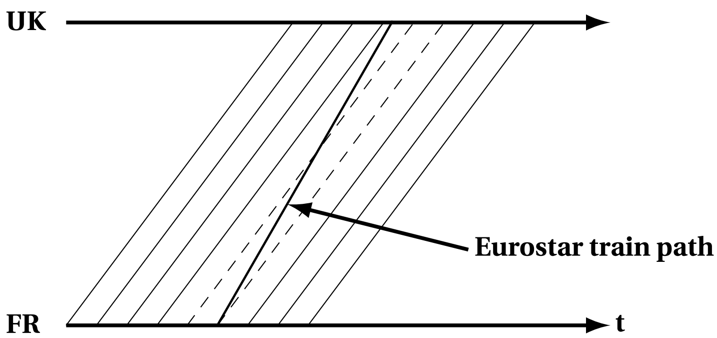

Figure 3.5 – Time-space diagram for the Eurostar train path

train is measured by the standard train path. For all the different trains, these values are summed up in the following Table 3.4.

Trains	Eurostar 160 km/h	Shuttles 140 km/h	Freight 120 km/h	Freight 100 km/h
Travel time (min)	21	26	27	32
Standard train paths	8/3	1	4/3	3

Table 3.4 – Travel times of all the different trains

Eurotunnel uses the standard train path and the capacity used by each train to compute by hand their cyclic schedules. By the way, the capacity used by Eurostars and freight trains determines the price they pay to travel in the Tunnel.

An example of a feasible schedule in given on Figure 3.6. This schedule represents a cyclic timetable of one hour, where the departure time of each train is indicated.

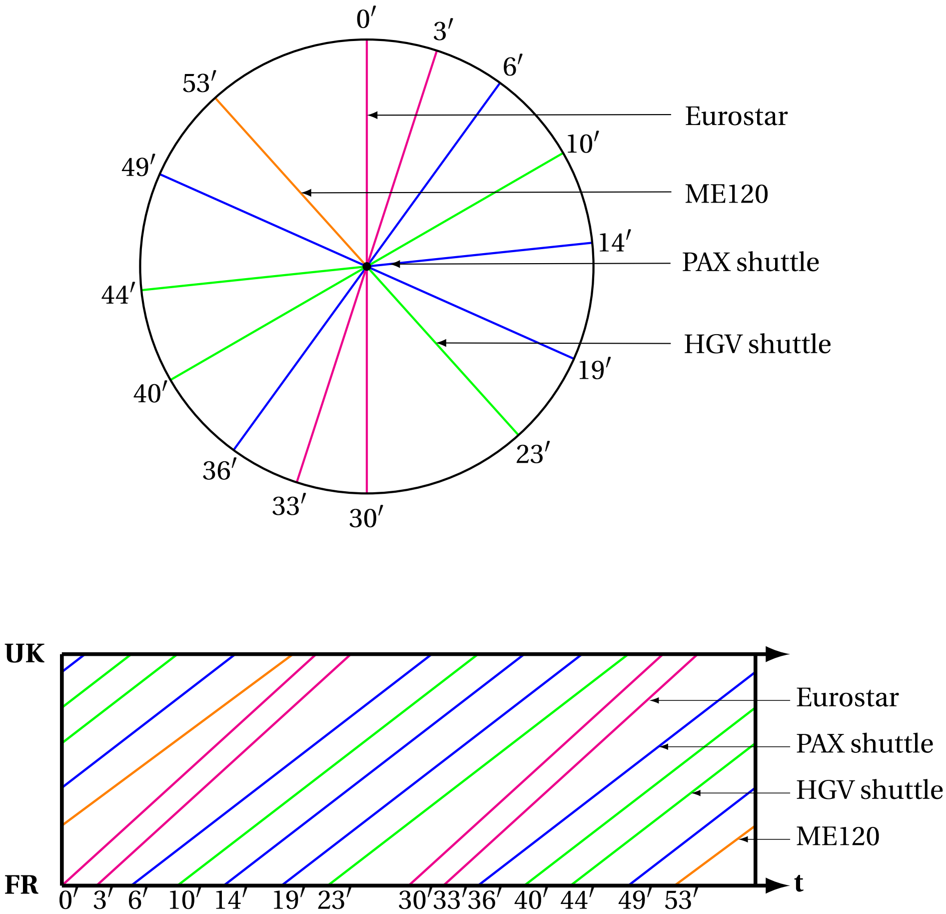

Figure 3.6 – An example of a feasible schedule designed by Eurotunnel

4 Cyclic scheduling deterministic problem

This chapter describes a model for the scheduling problem faced by Eurotunnel and introduced in Chapter 3. A general model is presented and some numerical experiments have been performed to test the model. We present in the last section some schedule improvements that can be achieved by relaxing some constraints.

4.1 Problem

The aim is to compute a cyclic various trains: Eurostars, freight trains MA100 and ME120, HGV and PAX shuttles in a single direction. Since the schedules are symmetric (see Section 3.2.2), they are the same for both directions. There is thus no mention of FOCA and CAFO directions in this chapter. The objective is to maximize the number of HGV shuttles under the constraints defined in Section 3.2:

- Cyclicity: the schedule has to be cyclic with a period of length T.
- Discrete departure times: the trains can only be scheduled on discrete instants of step η in the cyclic period. The parameter η is a divisor of T, which means that T can be written as $a\eta$ where a is an integer.
- Security and buffer times: a minimum headway has to be scheduled between each pair of trains. The minimum headway for each ordered pair (A,A') being a train of type A followed by a train of type A' is denoted $K^{\mathrm{A,A'}}$. These quantities are positive.
- Commercial agreements with Eurostar: we try to figure out what could be a generalization of the rule given in Section 3.2.5 for two Eurostars. During peak periods, the Eurostars are gathered into groups of T/C^{Eur} Eurostars, whose departure times separated by C^{Eur} during the whole cyclic period. The remaining ones, which cannot form a group of T/C^{Eur} Eurostars, are scheduled in a row, with their departure times separated by C^{Eur}, but not during the whole cycle. Note that in the real case faced by Eurotunnel, Eurostars are gathered into pairs ($T/C^{\mathrm{Eur}} = 2$), and there is possibly one remaining Eurostar. In order to have a feasible cyclic schedule, we need C^{Eur} to be a divisor of T and a multiple of η. If the schedule is not in a peak period, the parameter C^{Eur} will be set to T.
- Loading platforms: At most two HGV shuttles can be scheduled in any time window

of length L^{HGV}, and at most two PAX shuttles can be scheduled in any time window of length L^{PAX}, these parameters being given in input.
- Equal distribution of PAX shuttles among the cycle: for any PAX shuttle, the following PAX shuttle (potentially using the cyclicity of the schedule) has to depart at most $T/N^{\mathrm{PAX}} \times 1.5$ min later.

Moreover, the number of trains of each type (except the number of HGV shuttles that we want to maximize) is fixed. The following Table 4.1 provides the parameters given in input of the problem.

T	Length of the cyclic period
η	Discretization of the departure times
$\mathscr{M}$	Set of train types: $\{$Eur, MA100, ME120, PAX, HGV$\}$
N^{A}	Number of trains of type $\mathrm{A} \in \mathscr{M} \setminus \{\mathrm{HGV}\}$
L^{A}	Size of the time window in which 3 shuttles of type $\mathrm{A} \in \{\mathrm{PAX, HGV}\}$ cannot be scheduled
C^{Eur}	Time between twin Eurostars in the commercial agreements
$K^{\mathrm{A,A'}}$	Minimum headway between departures of a train of type $\mathrm{A} \in \mathscr{M}$ followed by a train of type $\mathrm{A'} \in \mathscr{M}$

Table 4.1 – Sets and parameters

4.2 Model

Let N^{HGV} be any upper bound on the objective function (e.g. $\min\left(\left\lfloor \frac{T}{K^{\mathrm{HGV,HGV}}} \right\rfloor, 2\left\lfloor \frac{T}{L^{\mathrm{HGV}}} \right\rfloor\right)$).

The problem is modelled as a Mixed Integer Linear Program.

$$
\begin{array}{rrcll}
\text{Max} & \multicolumn{4}{l}{\displaystyle\sum_{j=1}^{N^{\text{HGV}}} x_j} \\
\text{s.t.} & d^{\text{Eur}}_{j+n_{FS}+1} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{1, \ldots, (n_{FS}+1)(N^{\text{Eur}} - n_{FS} n) - 1\right\} \quad \text{(i)} \\
& d^{\text{Eur}}_{j+n_{FS}} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{(n_{FS}+1)(N^{\text{Eur}} - n_{FS} n) + 1, \ldots, N^{\text{Eur}} - n_{FS}\right\} \quad \text{(ii)} \\
& d^{\text{A}}_{j} - d^{\text{A'}}_{j'} & \in & \left[K^{\text{A',A}}, T - K^{\text{A, A'}}\right]_T & \text{A, A'} \in \mathcal{M} \setminus \left\{\text{HGV}\right\},\ \text{A} \neq \text{A'}, \\
& & & & j \in \left\{1, \ldots, N^{\text{A}}\right\}, j' \in \left\{1, \ldots, N^{\text{A'}}\right\} \quad \text{(iii)} \\
& x_j & \leq & x_{j-1} & j \in \left\{2, \ldots, N^{\text{HGV}}\right\} \quad \text{(iv)} \\
& d^{\text{HGV}}_{j} - d^{\text{A}}_{j'} & \in & \left[K^{\text{A,HGV}} x_j, T - K^{\text{HGV,A}} x_j\right]_T & \text{A} \in \mathcal{M} \setminus \left\{\text{HGV}\right\}, j \in \left\{1, \ldots, N^{\text{HGV}}\right\}, \\
& & & & j' \in \left\{1, \ldots, N^{\text{A}}\right\} \quad \text{(v)} \\
& d^{\text{A}}_{j} - d^{\text{A}}_{j-1} & \geq & K^{\text{A,A}} & \text{A} \in \mathcal{M} \setminus \left\{\text{HGV}\right\}, \\
& & & & j \in \left\{2, \ldots, N^{\text{A}}\right\} \quad \text{(vi)} \\
& d^{\text{HGV}}_{j} - d^{\text{HGV}}_{j-1} & \geq & K^{\text{HGV,HGV}} x_j & j \in \left\{2, \ldots, N^{\text{HGV}}\right\} \quad \text{(vii)} \\
& d^{\text{A}}_{1} + T - d^{\text{A}}_{N^{\text{A}}} & \geq & K^{\text{A,A}} & \text{A} \in \mathcal{M} \quad \text{(viii)} \\
& d^{\text{PAX}}_{j} - d^{\text{PAX}}_{j-2} & \geq & L^{\text{PAX}} & j \in \left\{3, \ldots, N^{\text{PAX}}\right\} \quad \text{(ix)} \\
& d^{\text{HGV}}_{j} - d^{\text{HGV}}_{j-2} & \geq & L^{\text{HGV}} x_j & j \in \left\{3, \ldots, N^{\text{HGV}}\right\} \quad \text{(x)} \\
& d^{\text{A}}_{1} + T - d^{\text{A}}_{N^{\text{A}}-1} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \quad \text{(xi)} \\
& d^{\text{A}}_{2} + T - d^{\text{A}}_{N^{\text{A}}} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \quad \text{(xii)} \\
& d^{\text{PAX}}_{j} - d^{\text{PAX}}_{j-1} & \leq & T / N^{\text{PAX}} \times 1.5 & j \in \left\{2, \ldots, N^{\text{PAX}}\right\} \quad \text{(xiii)} \\
& d^{\text{PAX}}_{1} + T - d^{\text{PAX}}_{N^{\text{PAX}}} & \leq & T / N^{\text{PAX}} \times 1.5 & \quad \text{(xiv)} \\
& d^{\text{A}}_{j} & \in & \left\{0, \eta, \ldots, T - \eta\right\} & \text{A} \in \mathcal{M}, j \in \left\{1, \ldots, N^{\text{A}}\right\} \\
& x_j & \in & \{0, 1\} & j \in \left\{1, \ldots, N^{\text{HGV}}\right\},
\end{array}
$$

where $n = T / C^{\text{Eur}}$, $n_{FS} = \left\lfloor N^{\text{Eur}} / n \right\rfloor$, and $[\ell, u]_T$ is a cyclic time-window (see p. 36).

The model uses the following decision variables:

$$
\begin{array}{lcll}
d^{\text{A}}_{j} & \in & \left\{0, \eta, \ldots, T - \eta\right\} & \text{for all A} \in \mathcal{M} \text{ and } j \in \left\{1, \ldots, N^{\text{A}}\right\} \\
x_j & = & \begin{cases} 1 & \text{if HGV shuttle } j \text{ is scheduled} \\ 0 & \text{otherwise} \end{cases} & \text{for all } j \in \left\{1, \ldots, N^{\text{HGV}}\right\}.
\end{array}
$$

The variable d^{A}_{j} represents the discrete value of the departure time of the jth train of type A, regarding the discretization η.

The objective of the problem is to maximize the number of HGV shuttles in the timetable. The variables x_j are binary and are set to 1 if the jth HGV shuttle is really scheduled. This objective function can thus be written as

$$\text{Max} \sum_{j=1}^{N^{\text{HGV}}} x_j.$$

This Mixed Integer Linear Program correctly models the problem describes in Chapter 3 and all the constraints are obvious except the commercial agreements ((i) and (ii)) and the minimum headways constraints between different types of trains ((iii) and (v)) .

Commercial agreements constraints

The constant $n = T/C^{\text{Eur}}$ represents the number of Eurostars in a group with all their departures separated by C^{Eur} during the whole cycle. The constant $n_{FS} = \lfloor N^{\text{Eur}}/n \rfloor$ represents the number of such groups of Eurostars. Since the schedule is cyclic, if the set of optimal solutions is nonempty, then there exists an optimal solution for which the $N^{\text{Eur}} - n_{FS}n$ remaining Eurostars are scheduled in a row at the beginning of the period, after the first departures of the full sequences (see Figure 4.1).

The constraints can be expressed as follows:

$$\begin{aligned} d^{\text{Eur}}_{j+n_{FS}+1} - d^{\text{Eur}}_j &= C^{\text{Eur}} && \text{for all } j \in \{1,\dots,(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) - 1\} \\ d^{\text{Eur}}_{j+n_{FS}} - d^{\text{Eur}}_j &= C^{\text{Eur}} && \text{for all } j \in \{(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) + 1,\dots,N^{\text{Eur}} - n_{FS}\}. \end{aligned}$$

When C^{Eur} is set to T, the sets $\{1,\dots,(n_{FS}+1)(N^{\text{Eur}}-n_{FS}n)-1\}$ and $\{(n_{FS}+1)(N^{\text{Eur}}-n_{FS}n)+1,\dots,N^{\text{Eur}} - n_{FS}\}$ are empty and there is no more constraint.

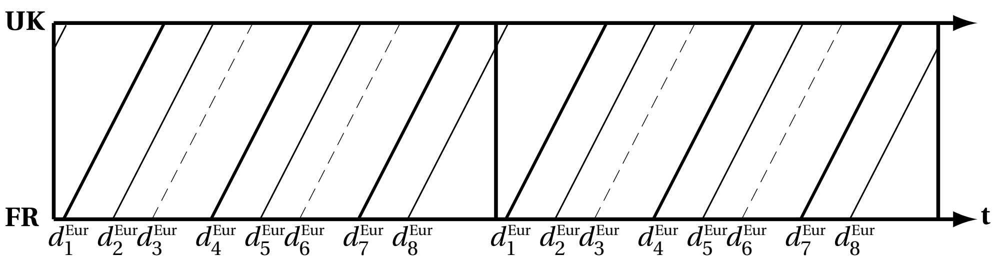

Figure 4.1 – Time-space diagram representing a feasible schedule for the commercial agreement with Eurostar

Minimum headways constraints

We define a *cyclic time-window* as a time-window repeated periodically. The notation $[\ell,u]_T$ is used to represent a time-window $[\ell,u]$ repeated with a period T. The constraint $x \in [\ell,u]_T$ means $x \mod T \in [\ell,u]$.

Serafini and Ukovich [62], and Voorhoeve[69] showed that the constraint $d_j^{\text{A}} - d_{j'}^{\text{A}'} \in \left[K^{\text{A}',\text{A}}, T - K^{\text{A},\text{A}'}\right]_T$ can be linearised as follows:

$$K^{\text{A}',\text{A}} \leq d_j^{\text{A}} - d_{j'}^{\text{A}'} + T p_{jj'}^{\text{AA}'} \leq T - K^{\text{A},\text{A}'}$$

where $p_{jj'}^{\text{AA}'}$ is a binary variable. The same linearisation is used for $d_j^{\text{HGV}} - d_{j'}^{\text{A}} \in \left[K^{\text{A,HGV}} x_j, T - K^{\text{HGV,A}} x_j\right]_T$. These two constraints model the minimum headways constraints between different types of trains. If the jth HGV shuttle is not scheduled, $x_j = 0$ and the last constraint is always satisfied.

4.3 Numerical results

We tested the model on several real instances of Eurotunnel. This section reports the numerical results of these experiments. For solving the Mixed Integer Linear Program, we used the MIP solver CPLEX 12.6 with its standard settings. The computation time is limited to 3600 s. All the experiments were performed on a MacBook Pro of 2014 with four 2.2 Ghz processors and 16 Gb of ram.

4.3.1 Instances

We fix the input parameters of the model. The length of the period is one hour ($T = 3600$ s) and the discretization is one minute ($\eta = 60$ s). The loading platforms constraints give $L^{\text{PAX}} = L^{\text{HGV}} = 720$ s and the commercial agreements impose $C^{\text{Eur}} = 1800$ s if these agreements should apply (peak periods) and $C^{\text{Eur}} = T = 3600$ s if they do not. The minimum headways $K^{\text{A},\text{A}'}$ for A,A' $\in \mathscr{M}$ are given in Table 3.3. The number of Eurostars, ME120, MA100, and PAX shuttles may vary.

4.3.2 Results

Table 4.2 gives the numerical results for real instances of Eurotunnel. The first 4 columns give respectively the number of Eurostars, ME120 trains, MA100 trains, and PAX shuttles. The fifth column gives the value of C^{Eur} (which varies weather the commercial agreements apply or not). The next two columns give respectively the lower bound (value of the best feasible solution) and the upper bound. The next column is the optimality gap. The last column provides the CPU time spent solving the problem, given in seconds.

4.3.3 Comments

The results are convincing. The real instances faced by Eurotunnel are solved in less than one second for all the cases. Moreover, the optimal value is the same as the one computed by Eurotunnel "by hand" without any optimization technique. The second instance of the table with 4 Eurostars, 1 ME120 trains and 5 PAX shuttles leads to a maximum of 4 HGV shuttles. Eurotunnel schedules this instance with the schedule given in Figure 3.6 as an example. This schedule also has 4 HGV shuttles.

N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	C^{Eur}	LB	UB	Gap(%)	CPU (s)
4	1	0	5	1800	4	4	0.0	0.2
4	1	0	4	1800	5	5	0.0	0.1
4	1	0	3	1800	6	6	0.0	0.2
4	1	0	2	1800	7	7	0.0	0.1
4	1	0	1	1800	8	8	0.0	0.1
4	1	0	0	1800	8	8	0.0	0.0
3	1	1	3	1800	5	5	0.0	0.1
2	1	0	4	3600	7	7	0.0	0.3
1	0	0	5	3600	8	8	0.0	0.0

Table 4.2 – Numerical results for real instances of Eurotunnel

The first 6 instances of the table have 4 Eurostars, 1 ME120, and a number of PAX shuttles varying from 0 to 5. Note that it allows to build a Pareto frontier of the maximum number of PAX and HGV shuttles, which can be an interest for Eurotunnel.

This model allows Eurotunnel to compute very quickly one hour cyclic schedules for any instance.

4.3.4 Schedule improvements

Some constraints are very limiting while computing an optimal cyclic schedule. This section compares the optimal value of the real problem with the problem where some of these constraints have been relaxed. It is useful for Eurotunnel to "price" these constraints in order to have objective arguments for future investments or negotiations.

Cyclic period

Eurotunnel computes one-hour cyclic schedules which are repeated hour after hour. But this may not be optimal and computing longer cycles may improve the number of HGV shuttles in the period. Table 4.3 compares the number of HGV shuttles in duplicated one-hour cyclic schedules with the optimal value of a cyclic schedule with a longer cyclic period. The parameters are exactly the same as the ones described in Section 4.3.1 except for T which varies here.

The first four columns give respectively the number of Eurostars, ME120 trains, MA100 trains, and PAX shuttles, and the fifth column gives the value of C^{Eur}. The sixth column gives the length of the period in hours. The next column is the value of the feasible solution computed with a duplicated one-hour schedule. Columns 8 and 9 give respectively the lower bound (best feasible solution) and the upper bound computed with the longer cyclic period. The next column is the optimality gap and the next one provides the CPU time spent solving the problem, given in seconds. The last column gives the improvement.

N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	C^{Eur}	T (h)	SOL	LB	UB	Gap(%)	CPU (s)	Imp(%)
4	2	2	10	1800	2	8	9	9	0.0	194.6	12.5
8	4	4	8	1800	4	24	30	30	0.0	1.2	25.0
6	2	2	8	1800	2	8	9	9	0.0	46.5	12.5
6	6	6	24	3600	6	36	42	53	26.2	3600	16.7

Table 4.3 – Numerical results for longer cyclic periods

The improvement is very significant for the few instances given in the table. This is mainly the case when there are several MA100 or ME120 freight trains which can be scheduled consecutively in the longer schedule. The computational time is much higher than for the one-hour cycle but since the schedules are computed days before, this time remains acceptable.

Loading platforms

Enlarging the terminal to allow 3 simultaneous loadings for shuttles (see Section 3.2.6) is possible but costs a lot of money. Relaxing this constraint gives the potential gain of having such a larger terminal in terms of number of shuttles. Table 4.4 compares the values of one-hour cyclic schedules with and without the loading platforms constraints. The parameters are exactly the same as the ones described in Section 4.3.1 except for L^{PAX} and L^{HGV} which are equal to 0 here.

The first four columns give respectively the number of Eurostars, ME120 trains, MA100 trains, and PAX shuttles and the fifth column gives the value of C^{Eur}. The next column is the value of the one-hour cyclic schedule with the loading platforms contraints. Columns 7 and 8 give respectively the lower bound (best feasible solution) and the upper bound computed without the loading platforms constraints. The next column is the optimality gap and the next one provides the CPU time spent solving the problem, given in seconds. The last column gives the improvement.

N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	C^{Eur}	SOL	LB	UB	Gap(%)	CPU (s)	Imp(%)
0	0	0	0	1800	10	15	15	0.0	0.0	50.0
0	0	0	3	1800	10	12	12	0.0	0.0	20.0
2	0	0	2	3600	8	10	10	0.0	0.1	25.0
1	0	0	3	1800	9	10	10	0.0	0.0	11.1

Table 4.4 – Numerical results for relaxed loading platforms constraints

Relaxing the loading platforms constraints improves significantly the maximal number of HGV shuttles in the cycle in comparison to the real problem. This is not surprising since this constraint is very limiting while computing the schedules. See for example the first instance with only HGV shuttles!

Discretization of the departure times

Eurotunnel computes schedules with departure times on full minutes (see Section 3.2.3). Relaxing this constraint by scheduling continuous departure times may thus improve their schedules. Note that relaxing this constraint could be still applicable in practice: since the scheduled departures times are at the entrance gate of the Tunnel, adapting the speed of the trains between loading platforms and the entrance gate would lead to departure times on full minutes at the loading platforms, which would be acceptable for the passengers and the staff. Table 4.5 compares the value of a schedule with departure times on full minutes with the optimal value of a one-hour cyclic schedule with continuous departure times. The parameters are exactly the same as the ones described in Section 4.3.1 and the variables d_j^A are continuous in the set $[0, T]$.

The first four columns give respectively the number of Eurostars, ME120 trains, MA100 trains, and PAX shuttles and the fifth column gives the value of C^{Eur}. The next column gives the value of the schedule with departure times on full minutes. Columns 7 and 8 give respectively the lower bound (best feasible solution) and the upper bound computed with continuous departures. The next column is the optimality gap and the next one provides the CPU time spent solving the problem, given in seconds. The last column gives the improvement.

N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	C^{Eur}	SOL	LB	UB	Gap(%)	CPU (s)	Imp(%)
4	0	0	3	1800	7	8	8	0.0	0.0	14.3
2	1	0	5	3600	6	7	7	0.0	0.1	16.7
1	1	1	3	1800	7	8	8	0.0	0.1	14.3
4	0	1	5	1800	3	4	4	0.0	0.1	33.3

Table 4.5 – Numerical results for continuous departure times

Relaxing the discretization of the departure times on full minutes to continuous departure times allows to schedule one HGV shuttle more than for the real problem for some instances.

Commercial agreements with Eurostar

The commercial agreements with Eurostar specify that Eurostars must be scheduled in groups where the departure times are separated by exactly 30 min. Relaxing totally this constraint is not a option for Eurostar, but relaxing the strict equality of 30 min by a time-window [27 min, 33 min] for instance may be possible. It gives Eurotunnel objective arguments for future negotiations. The constraints modelling the commercial agreements in the model can easily be adapted. Table 4.6 compares the value of a schedule with the current commercial agremments with the optimal value of a one-hour cyclic schedule with the relaxed commercial agreements of [27 min, 33 min]. The parameters are exactly the same as the ones described in Section 4.3.1 with the value of C^{Eur} set to 1800.

The first four columns give respectively the number of Eurostars, ME120 trains, MA100 trains, and PAX shuttles. The next column gives the value of the schedule with the current commercial agreements. Columns 7 and 8 give respectively the lower bound (best feasible solution) and the upper bound computed without the relaxed commercial agreements. The next column is the optimality gap and the next one provides the CPU time spent solving the problem, given in seconds. The last column gives the improvement.

N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	SOL	LB	UB	Gap(%)	CPU (s)	Imp(%)
2	0	1	0	7	8	8	0.0	0.0	14.3
2	0	1	1	7	8	8	0.0	0.0	14.3
3	1	1	0	7	8	8	0.0	0.0	14.3
3	0	1	1	7	8	8	0.0	0.0	14.3

Table 4.6 – Numerical results for the relaxed commercial agreements

Relaxing these constraints improve the schedules for some particular instances for which one more HGV shuttle can be scheduled in one hour.

Conclusion

Increasing the length of the cyclic period or relaxing the discretization of the departure times are operational levers to improve the schedules. Changing these rules and reshaping the operational constraints would allow Eurotunnel to increase the number of shuttles in the Tunnel.

Enlarging the terminal and the platforms would lead to a very important gain (up to 50% for some instances). The current constraint of the loading platforms is technical and relaxing it would cost money to the company. Changing may be more complicated than for the previous constraints but it is a decision within Eurotunnel and remains feasible.

Relaxing the commercial agreements with Eurostar leads to a gain of one shuttle per hour for some instances. The constraint is maybe the more difficult one to set up in practice since it implies negotiations.

Relaxing these constraints allows to schedule more HGV shuttles with an important gain (more than 10% for the instances presented here). But combining them would lead to an even more important gain for Eurotunnel.

Literature review. This section provides a review of the research on train scheduling and cyclic timetabling. Apart from [62] and [69], our work is not directly related to this review.

The train planning process can be decomposed into network design, timetabling, rolling stock and personnel planning. Assad [1] and Cordeau et al. [18] give a general overview of mathematical models for train transportation. For more contributions on train scheduling in general, we refer to the contributions by Caprara et al. [16], Guihaire et Hao [29], Huisman et al. [33], Kroon et al. [38], and Larson and Odoni [43]. The timetabling part consists in giving the departure and arrival times for each train and for each station in the railway network.

We do not review the literature on noncyclic train scheduling, but the important contributions are referred in the reviews from Assad [1] and Cordeau et al. [18]. Cacchiani et al. [13] and Zhou and Zhong [73] provide interesting methods to solve these problems.

In a cyclic schedule, train departure and arrival times are operated regularly with respect to a cyclic period. In the general case, this period can last any time, it can be for example one hour, two hours, or a whole day. For other companies, cyclic timetables are mainly used for passenger trains, thought freight railway schedules can also be built this way. For instance, Swiss, Dutch and German companies usually design such timetables with a cyclic period of one hour. This kind of timetable is very convenient for customers since the departure times for the trains at any station are the same every hour. They do not have to memorize complex departure times for their regular trains, but only the minutes of the hour at which they leave. Moreover, these timetables also have many advantages for the operator himself. It makes easier the computation of the total daily schedule since one only has to focus on a single cyclic period. Rush hours and low-demand hours still need some adjustments but the total computation is reduced. It also makes easier the management of the fixed rolling stock of all the trains. In the particular case of Eurotunnel, a train leaves the same station every two hours. The operator thus has an easy control on the rolling stock during the period. These timetables are also less complex for the drivers and the staff since their working hours remain the same period after period. However, cyclic timetables may imply higher operational costs since the service is globally the same whereas the demand varies. Although it can be adjusted, it yields to a lower occupation of the trains for some departures. For a classical railway operator, this issue can be avoided using longer trains when the demand is high and shorter ones when the demand is low.

The cyclic train scheduling was introduced by Serafini and Ukovich [62] who presented the Periodic Event Scheduling Problem (PESP). They considered the problem of scheduling periodic events and activities which are repeated at a constant rate. The relative position of pairs of events is affected by time-window constraints and the objective is to find a feasible timetable which gives the instants at which the events take place.

Voorhoeve [69] was one of the first to consider the PESP model for the train scheduling problem. He considered the Cyclic Railway Timetable Problem (CRTP) which modelled the timetabling problem with cyclic time window constraints for each pair of trains. Schrijver and Steenbeek [61] used the CRTP model by Voorhoeve and improved it using constraint programming algorithms. They performed a lot of tests on the real Dutch railway network.

Nachtigall [50, 52] also used the PESP model for a problem of computing a cyclic train schedule with periodic constraints on pairs of events. He added an objective function of minimizing the passenger waiting times in the network and the passenger total transit time. This objective function was also studied with Voget in [53]. They created heuristics to build feasible timetables which were improved with genetic algorithms. Nachtigall also proposed bi-criteria objective functions minimizing both the passengers waiting times and the rolling stock costs, or minimizing the cost of the infrastructure while maximizing the total revenue for the operator. He improved the PESP model in [51] considering several trains lines with different cyclic periods.

5 Buffer time allocation

This chapter describes the problem of buffer time allocation met by Eurotunnel and already introduced in Section 3.2. We give our model for two versions of the problem and a method to approximate an optimal solution. We performed numerical experiments to test our methods. In a last section, we describe a related problem maximizing the number of HGV shuttles and minimizing the expected average waiting time.

5.1 Problems

The aim is to compute a cyclic schedule with a period of length T and a fixed number N^{A} of trains of every type A. The cyclic schedule is then evaluated on H consecutive periods and all the departures are subject to a primary disturbance from a probability distribution depending on the type of train. All the departures are subject to these primary disturbances, but they are also impacted by secondary disturbances implied by the primary ones.

The objective is to minimize the expected average delay of each departure for H consecutive periods of the cyclic schedule, the delay being the difference between the scheduled departure time and the effective departure time taking into account the disturbances. The problem is thus to allocate buffer times between the departures of two consecutive trains. Naturally, the cyclic schedule has to satisfy the constraints defined in Section 3.2: cyclicity, discrete scheduled departure times, commercial agreements with Eurostar, loading platforms, and even distribution of PAX shuttles among the cycle. Only the minimum headways constraints are different: since the buffer times are allocated by the optimization process, we must consider only the security headways between two consecutive departures and not the minimum headways (which contain the security headways and the buffer times, see Section 3.2.4). We denote by $S^{\mathrm{A,A'}}$ the security headways between two consecutive trains respectively of types A and A'. Moreover, the effective departure times for the H consecutive periods only have to satisfy the security and the loading platforms constraints. They do not need to satisfy the discretization constraint, the commercial agreements with Eurostars and the even distribution of PAX shuttles since these constraints are operational constraints for the scheduling , but can be violated by the effective departures. The objective is to compute a schedule minimizing the expected value of the average delay of each departure for H consecutive periods.

The parameters given in input of the problem are the same as the ones for the previous chapter provided in Table 4.1, except, of course, the security headways which are given in Tables 3.1 and 3.2. We are also given in input the number N^{HGV} of HGV shuttles, the number H of consecutive periods, and of each type of train, the probability distribution Δ^{A} for the primary disturbances. We define the total number of trains in the schedule $N^{\text{all}} = N^{\text{Eur}} + N^{\text{PAX}} + N^{\text{HGV}} + N^{\text{MA100}} + N^{\text{ME120}}$. We set $n = T/C^{\text{Eur}}$ and $n_{FS} = \lfloor N^{\text{Eur}}/n \rfloor$ respectively the number of Eurostars in a group with all their departures separated by C^{Eur} and the number of such groups of Eurostars.

We consider two versions of this problem:

- In a first version, we consider that the relative order of the departures in the schedule is given in input.
- In a second version, we consider that only the number of trains of each type is given in input, but without the order of the departures in the schedule. This version may give a better objective value than the first version.

5.2 First version

For this version, the number of trains of each type and the order in which they are scheduled is given. We denote by A_f the type of the last train of the schedule.

5.2.1 Model

The first version of the problem can be modelled with a two-stage recourse stochastic model:

$$
\begin{array}{lrcll}
\text{Min}_{\boldsymbol{d}} & f(\boldsymbol{d}) := \mathbb{E}\left[Q_1(\boldsymbol{d},\boldsymbol{\delta})\right] & & & \\
\text{s.t.} & d^{\text{Eur}}_{j+n_{FS}+1} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{1,\dots,(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) - 1\right\} \\
& d^{\text{Eur}}_{j+n_{FS}} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) + 1, \dots, N^{\text{Eur}} - n_{FS}\right\} \\
& d^{t(j,\text{A})}_{i(j,\text{A})} - d^{\text{A}}_{j} & \geq & S^{\text{A},t(j,\text{A})} & (\text{A}, j) \in \mathcal{M} \times \left\{1,\dots,N^{\text{A}}\right\} \setminus \left\{\left(N^{\text{A}_f}, \text{A}_f\right)\right\} \\
& d^{t(N^{\text{A}_f},\text{A}_f)}_{i(N^{\text{A}_f},\text{A}_f)} + T - d^{\text{A}_f}_{N^{\text{A}_f}} & \geq & S^{\text{A}_f,t(N^{\text{A}_f},\text{A}_f)} & \\
& d^{\text{A}}_{j} - d^{\text{A}}_{j-2} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\}, j \in \left\{3,\dots,N^{\text{A}}\right\} \\
& d^{\text{A}}_{1} + T - d^{\text{A}}_{N^{\text{A}}-1} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \\
& d^{\text{A}}_{2} + T - d^{\text{A}}_{N^{\text{A}}} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \\
& d^{\text{PAX}}_{j} - d^{\text{PAX}}_{j-1} & \leq & T/N^{\text{PAX}} \times 1.5 & j \in \left\{2,\dots,N^{\text{PAX}}\right\} \\
& d^{\text{PAX}}_{1} + T - d^{\text{PAX}}_{N^{\text{PAX}}} & \leq & T/N^{\text{PAX}} \times 1.5 & \\
& d^{\text{A}}_{j} & \in & \left\{0,\eta,\dots,T-\eta\right\} & \text{A} \in \mathcal{M}, j \in \left\{1,\dots,N^{\text{A}}\right\},
\end{array}
\tag{5.1}
$$

where $Q_1(\boldsymbol{d},\boldsymbol{\delta})$ is the recourse function defined hereafter, the d^{A}_{j}'s are variables representing the departure times of trains of type A, and $\boldsymbol{\delta}$ are random variables modelling the primary disturbances.

$$
\begin{array}{lrl}
Q_1(\boldsymbol{d},\boldsymbol{\delta}) = \text{Min}_{\boldsymbol{e}} & \multicolumn{2}{l}{\dfrac{1}{HN^{\text{all}}} \displaystyle\sum_{\text{A}\in\mathcal{M}} \sum_{j=1}^{N^{\text{A}}} \sum_{h=0}^{H-1} \left(e^{\text{A}}_{jh} - d^{\text{A}}_{j} - hT\right)} \\
\text{s.t.} & e^{t(j,\text{A})}_{i(j,\text{A})h} - e^{\text{A}}_{jh} \geq S^{\text{A},t(j,\text{A})} & (j,A) \in \left\{1,\dots,N^{\text{A}}\right\} \times \mathcal{M} \setminus \left\{\left(N^{\text{A}_f}, \text{A}_f\right)\right\}, h \in \left\{0,\dots,H-1\right\} \\
& e^{t(N^{\text{A}_f},\text{A}_f)}_{i(N^{\text{A}_f},\text{A}_f)h+1} - e^{\text{A}_f}_{N^{\text{A}_f}h} \geq S^{\text{A}_f,t(N^{\text{A}_f},\text{A}_f)} & h \in \left\{0,\dots,H-2\right\} \\
& e^{\text{A}}_{jh} - e^{\text{A}}_{j-2,h} \geq L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\}, j \in \left\{3,\dots,N^{\text{A}}\right\}, h \in \left\{0,\dots,H-1\right\} \\
& e^{\text{A}}_{1,h+1} - e^{\text{A}}_{N^{\text{A}}-1,h} \geq L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\}, h \in \left\{0,\dots,H-2\right\} \\
& e^{\text{A}}_{2,h+1} - e^{\text{A}}_{N^{\text{A}}h} \geq L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\}, h \in \left\{0,\dots,H-2\right\} \\
& e^{\text{A}}_{jh} - d^{\text{A}}_{j} - hT \geq \delta^{\text{A}}_{jh} & \text{A} \in \mathcal{M}, j \in \left\{1,\dots,N^{\text{A}}\right\}, h \in \left\{0,\dots,H-1\right\}, \\
& e^{\text{A}}_{jh} \in \mathbb{R} & \text{A} \in \mathcal{M}, j \in \left\{1,\dots,N^{\text{A}}\right\}, h \in \left\{0,\dots,H-1\right\},
\end{array}
\tag{5.2}
$$

where $i(j,\text{A})$ and $t(j,\text{A})$ are two functions which respectively give the index and the type of the train immediately following the jth train of type A, δ^{A}_{jh} is the primary disturbance of the jth train of type A in period h, and e^{A}_{jh} is the effective departure time of the jth train of type A in period h.

In the recourse function (5.2), the variables $\boldsymbol{d}$ are fixed, the random disturbances $\boldsymbol{\delta}$ have been revealed and the recourse variables $\boldsymbol{e}$ are optimized.

5.2.2 Method

Computing an optimal solution of (5.1) is difficult. As Kroon et al. [41] did in their paper, we use the Sample Average Approximation which is a Monte-Carlo method to sample the problem. Suppose that a random sample $(\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_R)$ of R independent realizations of the random variable is generated, where R is fixed independently of the problem. Denoting by X the set of feasible solutions $\boldsymbol{d}$ of (5.1) and replacing $f(\cdot)$ by an approximation give the following program:

$$\mathrm{Min}_{\boldsymbol{d}\in X} \left\{ \hat{f}_R(\boldsymbol{d}) := \frac{1}{R}\sum_{r=1}^{R} Q_1(\boldsymbol{d}, \boldsymbol{\delta}_r) \right\}, \tag{5.3}$$

where $Q_1(\cdot,\cdot)$ is given in (5.2). Note that $\mathbb{E}\left[\hat{f}_R(\boldsymbol{d})\right] = f(\boldsymbol{d})$. We denote by v^* and the random variable $\hat{v}_R$ the optimal values of respectively program (5.1) and program (5.3).

Proposition 1. *We have* $\mathbb{E}[\hat{v}_R] \le v^*$ *and* $\hat{v}_R$ *converges to* v^* *with probability 1.*

Proof. We denote by X^* the set of optimal solutions of (5.1). We have

$$\mathbb{E}[\hat{v}_R] = \mathbb{E}\left[\min_{\boldsymbol{d}\in X} \hat{f}_R(\boldsymbol{d})\right] \le \mathbb{E}\left[\min_{\boldsymbol{d}\in X^*} \hat{f}_R(\boldsymbol{d})\right] \le \min_{\boldsymbol{d}\in X^*} \mathbb{E}\left[\hat{f}_R(\boldsymbol{d})\right] = v^*,$$

which proves the first inequality. The convergence properties of such problems have also been studied (see King and Rockafellar [36], Shapiro and Homem-de-Mello [64], Kleywegt et al. [37], and Shapiro et al. [63]). In our case, the proof is immediate with [37, Proposition 2.1] since X is finite (the scheduled departure times are discrete), $\mathbb{E}\left[Q_1(\boldsymbol{d},\boldsymbol{\delta})\right]$ and $Q_1(\boldsymbol{d},\boldsymbol{\delta}_r)$ are measurable for all $\boldsymbol{d} \in X$, and the probability distributions Δ^{A} are bounded (see Section 5.4.1). □

Moreover, we define for all $\varepsilon > 0$, the sets of ε-optimality solutions

$$\begin{aligned} X^*_{\varepsilon} &= \left\{\bar{\boldsymbol{d}} \in X:\ \mathbb{E}\left[Q_1(\bar{\boldsymbol{d}},\boldsymbol{\delta})\right] \le \min_{\boldsymbol{d}\in X} \mathbb{E}\left[Q_1(\boldsymbol{d},\boldsymbol{\delta})\right] + \varepsilon\right\} \\ X^*_{R\varepsilon} &= \left\{\bar{\boldsymbol{d}} \in X:\ \tfrac{1}{R}\textstyle\sum_{r=1}^{R} Q_1(\bar{\boldsymbol{d}},\boldsymbol{\delta}_r) \le \min_{\boldsymbol{d}\in X} \tfrac{1}{R}\textstyle\sum_{r=1}^{R} Q_1(\boldsymbol{d},\boldsymbol{\delta}_r) + \varepsilon\right\}. \end{aligned}$$

The set $X^*_{R\varepsilon}$ is a random set. We also have with [37, Proposition 2.1] that for R large enough and for all $\varepsilon > 0$, the probability $\mathbb{P}\left(X^*_{R\varepsilon} \subset X^*_{\varepsilon}\right) = 1$. Thus, the set of optimal scheduled departure times for the sampled problem converges with probability 1 to the set of optimal scheduled departure times of (5.1). In particular, if there is a unique optimal schedule, the optimal schedule of the sampled problem converges to this solution with probability 1.

Lower bound estimation

The expected value $\mathbb{E}[\hat{v}_R] \leq v^*$ can be estimated. We generate Ω independent samples of $(\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_R)$. The sample $\boldsymbol{\delta}_r^\omega$ corresponds to the ωth sample of the rth realization with $\omega \in \{1, \ldots, \Omega\}$ and $r \in \{1, \ldots, R\}$. We solve the approximated program (5.3) for all ω and denote by $\hat{v}_R^\omega$ and $\hat{\boldsymbol{d}}_R^\omega$ respectively its optimal value and an optimal solution. We compute the average $\bar{v}_{R\Omega}$ of these objective values and an estimator $\sigma_{R\Omega}^2$ of its variance:

$$\bar{v}_{R\Omega} = \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \hat{v}_R^\omega \qquad \text{and} \qquad \sigma_{R\Omega}^2 = \frac{1}{\Omega - 1} \sum_{\omega=1}^{\Omega} \left(\bar{v}_{R\Omega} - \hat{v}_R^\omega \right)^2.$$

The average objective value $\bar{v}_{R\Omega}$ and the estimator of the variance $\sigma_{R\Omega}^2$ are unbiased estimators of respectively $\mathbb{E}[\hat{v}_R]$ and $\mathrm{Var}(\hat{v}_R)$, and $\bar{v}_{R\Omega}$ provides an estimator of a lower bound on v^*. We have with the Central Limit Theorem

$$\sqrt{\Omega}\,(\bar{v}_{R\Omega} - \mathbb{E}[\hat{v}_R]) \xrightarrow[R \to +\infty]{\mathscr{L}} \mathcal{N}\left(0, \sigma_{R\Omega}^2\right)$$

and an approximate $(1-\alpha)$-confidence interval for $\mathbb{E}[\hat{v}_R]$ which is a lower bound of our real problem:

$$\left[\bar{v}_{R\Omega} - \frac{z_{\alpha/2}\sigma_{R\Omega}}{\sqrt{\Omega}}, \bar{v}_{R\Omega} + \frac{z_{\alpha/2}\sigma_{R\Omega}}{\sqrt{\Omega}} \right],$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)$-quantile of the standard normal distribution.

Upper bound estimation

Picking any feasible solution $\hat{\boldsymbol{d}} \in X$ of (5.1) provides an upper bound $f(\hat{\boldsymbol{d}})$ on the optimal value. This upper bound can be estimated. We generate an independent sample $(\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_\Lambda)$ of the random variables and we define

$$\bar{f}_\Lambda(\hat{\boldsymbol{d}}) = \frac{1}{\Lambda} \sum_{\lambda=1}^{\Lambda} Q_1(\hat{\boldsymbol{d}}, \boldsymbol{\delta}_\lambda)$$

which is an unbiased estimator of $f(\hat{\boldsymbol{d}})$. The computation of $Q_1(\hat{\boldsymbol{d}}, \boldsymbol{\delta}_\lambda)$ is done via simulation and is almost immediate. However, the variance of this estimator is quite large since the values $Q_1(\hat{\boldsymbol{d}}, \boldsymbol{\delta}_\lambda)$ may vary a lot. A classical way to reduce the variance is to generate Z independent samples of $(\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_\Lambda)$, the same way as it is done for the estimation of the lower bound. The sample $\boldsymbol{\delta}_\lambda^\zeta$ corresponds to the ζth sample of the λth realization with $\zeta \in \{1, \ldots, Z\}$ and $\lambda \in \{1, \ldots, \Lambda\}$. We define

$$\bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}) = \frac{1}{Z} \sum_{\zeta=1}^{Z} \bar{f}_\Lambda^\zeta(\hat{\boldsymbol{d}}),$$

where $\bar{f}_\Lambda^\zeta(\hat{\boldsymbol{d}})$ is the value $\bar{f}_\Lambda(\hat{\boldsymbol{d}})$ for the ζth sample of $\boldsymbol{\delta}$.

The method to compute a good feasible solution is to pick the feasible solution $\hat{\boldsymbol{d}}_R^\omega$ among the set $\left\{\hat{\boldsymbol{d}}_R^\omega,\ \omega \in \{1,\dots,\Omega\}\right\}$ of feasible solutions computed for the lower bound estimation, for which $\bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}_R^\omega)$ is minimal. We also have with the Central Limit Theorem

$$\sqrt{Z}\left(\bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}_R) - f(\hat{\boldsymbol{d}}_R)\right) \xrightarrow[R\to+\infty]{\mathscr{L}} \mathcal{N}\left(0, \sigma^2_{\Lambda Z}\right),$$

where $\sigma^2_{\Lambda Z} = \frac{1}{Z-1}\sum_{\zeta=1}^{Z}\left(\bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}_R) - \bar{f}^{\zeta}_{\Lambda}(\hat{\boldsymbol{d}})\right)^2$ is an estimator of the variance. We have an approximate $(1-\alpha)$-confidence interval for $f(\hat{\boldsymbol{d}}_R)$ which is an estimator of an upper bound of v^*:

$$\left[\bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}_R) - \frac{z_{\alpha/2}\sigma_{\Lambda Z}}{\sqrt{Z}}, \bar{f}_{\Lambda Z}(\hat{\boldsymbol{d}}_R) + \frac{z_{\alpha/2}\sigma_{\Lambda Z}}{\sqrt{Z}}\right].$$

5.3 Second version

For this version, only the number of trains of each type to schedule is given.

5.3.1 Model and Method

The second version of the problem can almost be modelled with the same two-stage recourse stochastic model as for the first version. Only the security constraints change since the order of the departures is not given here. The model is the following one:

$$\begin{array}{llcll}
\text{Min}_{\boldsymbol{d}} & \mathbb{E}\left[Q_2(\boldsymbol{d},\boldsymbol{\delta})\right] & & & \\
\text{s.t.} & d^{\text{Eur}}_{j+n_{FS}+1} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{1,\dots,(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) - 1\right\} \\
& d^{\text{Eur}}_{j+n_{FS}} - d^{\text{Eur}}_{j} & = & C^{\text{Eur}} & j \in \left\{(n_{FS}+1)(N^{\text{Eur}} - n_{FS}n) + 1,\dots, N^{\text{Eur}} - n_{FS}\right\} \\
& d^{\text{A}}_{j} - d^{\text{A'}}_{j'} & \in & \left[S^{\text{A',A}}, T - S^{\text{A,A'}}\right]_T & \text{A, A'} \in \mathcal{M},\ \text{A} \neq \text{A'}, \\
& & & & j \in \left\{1,\dots,N^{\text{A}}\right\}, j' \in \left\{1,\dots,N^{\text{A'}}\right\} \\
& d^{\text{A}}_{j} - d^{\text{A}}_{j-1} & \geq & S^{\text{A,A}} & \text{A} \in \mathcal{M} \text{ and } j \in \left\{2,\dots,N^{\text{A}}\right\} \\
& d^{\text{A}}_{j} - d^{\text{A}}_{j-2} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\}, j \in \left\{3,\dots,N^{\text{A}}\right\} \\
& d^{\text{A}}_{1} + T - d^{\text{A}}_{N^{\text{A}}-1} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \\
& d^{\text{A}}_{2} + T - d^{\text{A}}_{N^{\text{A}}} & \geq & L^{\text{A}} & \text{A} \in \left\{\text{PAX, HGV}\right\} \\
& d^{\text{PAX}}_{j} - d^{\text{PAX}}_{j-1} & \leq & T/N^{\text{PAX}} \times 1.5 & j \in \left\{2,\dots,N^{\text{PAX}}\right\} \\
& d^{\text{PAX}}_{1} + T - d^{\text{PAX}}_{N^{\text{PAX}}} & \leq & T/N^{\text{PAX}} \times 1.5 & \\
& d^{\text{A}}_{j} & \in & \left\{0,\eta,\dots,T-\eta\right\} & \text{A} \in \mathcal{M}, j \in \left\{1,\dots,N^{\text{A}}\right\},
\end{array}$$

where $Q_2(\boldsymbol{d},\boldsymbol{\delta})$ is the recourse function defined as

$$
\begin{array}{rrcll}
\multicolumn{5}{l}{Q_2(\boldsymbol{d},\boldsymbol{\delta}) = \mathrm{Min}_{\boldsymbol{e}} \ \dfrac{1}{HN^{\mathrm{all}}} \displaystyle\sum_{\mathrm{A}\in\mathcal{M}} \sum_{j=1}^{N^{\mathrm{A}}} \sum_{h=0}^{H-1} \left(e^{\mathrm{A}}_{jh} - d^{\mathrm{A}}_{j} - hT\right)} \\
\text{s.t.} & e^{\mathrm{A}}_{jhr} - e^{\mathrm{A}'}_{j'hr} & \in & \left[S^{\mathrm{A'A}}, M_r - S^{\mathrm{AA'}}\right]_{M_r} & \mathrm{A}, \mathrm{A'} \in \mathcal{M},\ \mathrm{A} \neq \mathrm{A'},\ j \in \{1,\ldots,N^{\mathrm{A}}\}, \\
 & & & & j' \in \{1,\ldots,N^{\mathrm{A'}}\}, h \in \{0,\ldots,H-1\}, \\
 & & & & r \in \{1,\ldots,R\} \\
 & e^{\mathrm{A}}_{1,h+1,r} - e^{\mathrm{A}'}_{N^{\mathrm{A}'}hr} & \geq & S^{\mathrm{A'A}} & \mathrm{A}, \mathrm{A'} \in \mathcal{M}, h \in \{0,\ldots,H-2\}, \\
 & & & & r \in \{1,\ldots,R\} \\
 & e^{\mathrm{A}}_{jhr} - e^{\mathrm{A}}_{j-1,hr} & \geq & S^{\mathrm{AA}} & \mathrm{A} \in \mathcal{M}, j \in \{2,\ldots,N^{\mathrm{A}}\}, \\
 & & & & h \in \{0,\ldots,H-1\}, r \in \{1,\ldots,R\} \\
 & e^{\mathrm{A}}_{jh} - e^{\mathrm{A}}_{j-2,h} & \geq & L^{\mathrm{A}} & \mathrm{A} \in \{\mathrm{PAX, HGV}\}, j \in \{3,\ldots,N^{\mathrm{A}}\}, \\
 & & & & h \in \{0,\ldots,H-1\} \\
 & e^{\mathrm{A}}_{1,h+1} - e^{\mathrm{A}}_{N^{\mathrm{A}}-1,h} & \geq & L^{\mathrm{A}} & \mathrm{A} \in \{\mathrm{PAX, HGV}\}, h \in \{0,\ldots,H-2\} \\
 & e^{\mathrm{A}}_{2,h+1} - e^{\mathrm{A}}_{N^{\mathrm{A}}h} & \geq & L^{\mathrm{A}} & \mathrm{A} \in \{\mathrm{PAX, HGV}\}, h \in \{0,\ldots,H-2\} \\
 & e^{\mathrm{A}}_{jh} - d^{\mathrm{A}}_{j} - hT & \geq & \delta^{\mathrm{A}}_{jh} & \mathrm{A} \in \mathcal{M}, j \in \{1,\ldots,N^{\mathrm{A}}\}, \\
 & & & & h \in \{0,\ldots,H-1\}, \\
 & e^{\mathrm{A}}_{jh} & \in & \mathbb{R} & \mathrm{A} \in \mathcal{M}, j \in \{1,\ldots,N^{\mathrm{A}}\}, \\
 & & & & h \in \{0,\ldots,H-1\},
\end{array}
$$

where $M_r = T + \sum_{\mathrm{A}\in\mathcal{M}} \sum_{j=1}^{N^{\mathrm{A}}} \sum_{h=0}^{H-1} \delta^{\mathrm{A}}_{jhr}$ for all $r \in \{0,\ldots,R\}$.

The constraints modelling the security constraints for the effective departures are different than for the first version since the order is unknown. We need cyclic time windows (see p.36) for the scheduled departure times $\boldsymbol{d}$ and for the effective departure times $\boldsymbol{e}$ of the recourse program. The order of the trains in the schedule and the effective departures are the same and thus, the dummy variables $p^{\mathrm{AA'}}_{jj'}$ used to model the cyclic time windows are also the same for these two set of variables. The constant M_r is an upper bound on the value of e^{A}_{jhr} for all j, h, r, and A. Among all the effective departures from the same period h, this constraint acts as if for every scenario r, the schedule was cyclic with a huge period M_r.

We also use the Sample Average Approximation method for our problem. We have the convergence in probability 1 with the same arguments as for Proposition 1 and the same methods to compute confidence intervals for a lower bound and an upper bound.

5.4 Numerical results

We tested the model on several instances provided by Eurotunnel. This section reports the numerical results of these experiments. The results give the 95%-confidence intervals for some lower and upper bounds on the optimal value for various instances, for Versions

1 and 2. These results are compared to the 95%-confidence intervals on the average delay with the scheduled departure times with no stochastic optimization computed in Chapter 4. These scheduled departure times are computed with minimum headways containing *a priori* buffer times. We simulate Λ samples of delays, the same way as for the upper bound estimation in Section 5.2.2, to compute the objective value.

For solving the Mixed Integer Linear Programs, we used the MIP solver CPLEX 12.6 with its standard settings. The computation time is limited to one hour. All the experiments were performed on a MacBook Pro of 2014 with four 2.2 Ghz processors and 16 Gb of ram.

5.4.1 Instances

The parameters T, L^{HGV}, L^{PAX}, and C^{Eur} are the same as the ones given in Section 4.3.1. The number of consecutive periods for which the cyclic schedule is duplicated is set to $H = 6$ h. The number R of scenarios differs for the two versions. It is fixed to 500 for the first version and to 10 for the second since the computation time is much larger. The number of samples Ω to estimate a lower bound is not fixed *a priori*, the algorithm computes as many iterations as possible in the limited time of one hour. The value of Ω is given in the results. The size of the samples and the number of these samples generated to estimate an upper bound is respectively fixed to $\Lambda = 10000$ and $Z = 100$. As described in Section 5.2.2, the method to compute the upper bounds consists in using the feasible solutions used for the computation of the lower bounds. Once these solutions have been computed, the computation time to simulate they values is negligible.

Security headways

The security headways $S^{\mathrm{A,A'}}$ for A,A' $\in \mathscr{M}$ are given in Tables 3.2 and 3.1.

The discretization of the schedule also plays an important role in the results. Indeed, using a full minute discretization imposes to schedule artificial buffer time between consecutive departures. Relaxing the discretization to continuous scheduled departure times (as it is done in Section 4.3.4) leaves more freedom to the schedule. Both cases are used: $\eta = 60$ for the full minute discretization and $\eta = 1$ for the continuous departures.

Primary disturbances

We set the primary disturbances $\boldsymbol{\delta}^{\mathrm{A}}$ using the data of the real delays for one month at Eurotunnel. These delays combine the primary disturbances and the secondary ones. We did not manage to separate them and we decided to use these values for the primary disturbances in our experiments. These delays are given in full minutes. We gathered all the delays of each type of trains (Eurostar, PAX shuttle, and HGV shuttle) and we draw uniformly one of these delays for each primary disturbance. We do not have real data for the delays of MA100 and ME120 freight trains, but since these trains are mainly scheduled at night time, when the traffic is low, we can consider that they are not delayed when arriving at the terminal and thus not subject to any primary disturbance. The probability distributions for Eurostars, PAX shuttles, and HGV shuttles are given in the following Figure 5.1.

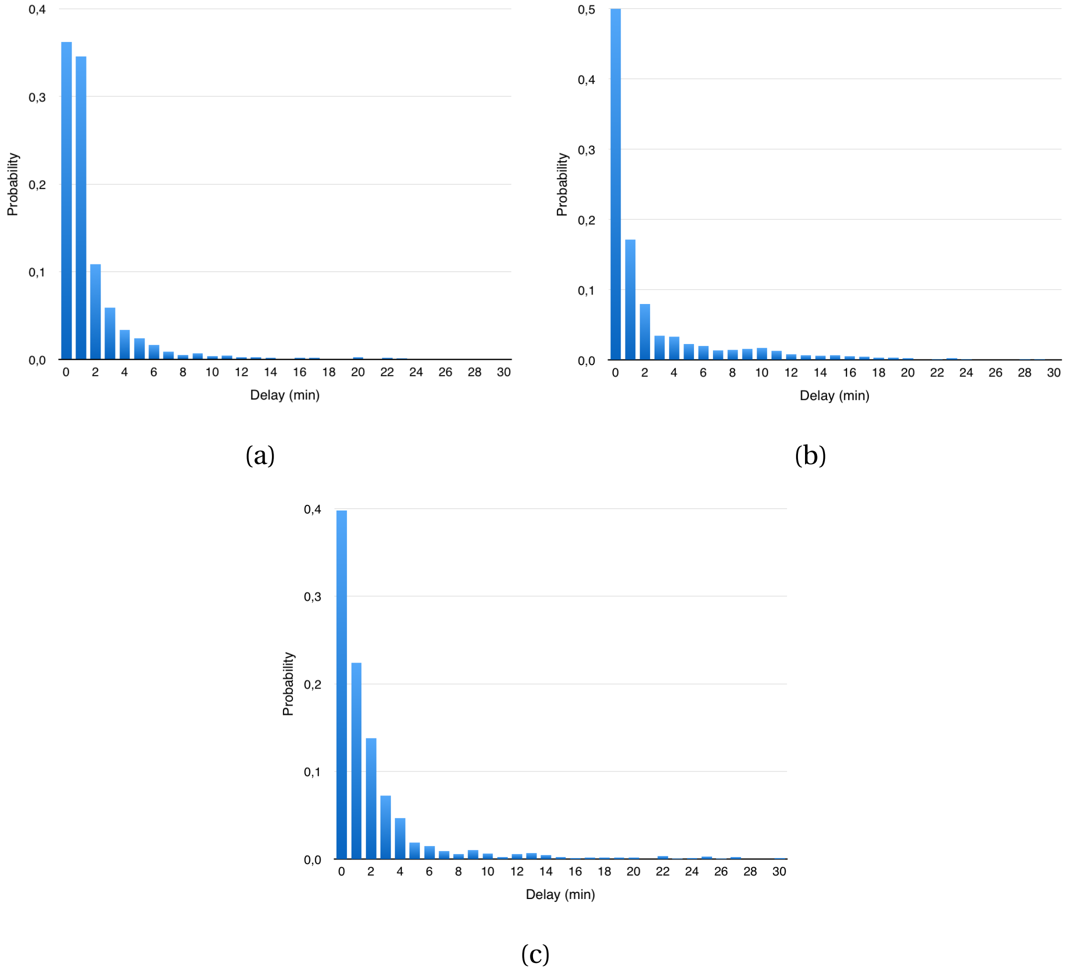

Figure 5.1 – Probabilities of a delay for an HGV shuttle (a), a PAX shuttle (b), and a Eurostar (c)

5.4.2 Results

Table 5.1 gives the numerical results for the buffer time allocation problem. The first column gives the order of the different trains. The value 'E' represents a Eurostar, 'Ma' a MA100 freight train, 'Me' a ME120 freight train, 'P' a PAX shuttle, and 'H' an HGV shuttle. The second column is the value of C^{Eur}, the third column provides the direction (FOCA or CAFO). The fourth column gives the discretization for the scheduled departure times. The next column provides the confidence interval on the average delay using the scheduled departure times computed in Chapter 4 (see Table 4.2). The next two columns give respectively the 95%-confidence intervals of a lower bound and an upper bound on the optimal value for the first version. The last two columns give the same values for the second version.

Instance					Version 1		Version 2	
Order	C^{Eur}	Dir	η	SOL	LB	UB	LB	UB
EHHPPHHPHHPHHP	3600	FOCA	60	477.89 ± 0.61	450.20 ± 1.90	449.33 ± 0.66	439.45 ± 11.83	448.29 ± 0.76
			1		447.31 ± 1.19	446.98 ± 0.69	424.98 ± 12.12	447.96 ± 0.66
		CAFO	60	505.53 ± 0.75	479.92 ± 2.16	477.34 ± 0.64	459.91 ± 12.45	475.74 ± 0.86
			1		474.99 ± 1.26	475.70 ± 0.80	447.53 ± 10.83	475.34 ± 0.59
EEPHHPHHMeHPHPH	3600	FOCA	60	439.14 ± 0.55	400.95 ± 1.87	402.90 ± 0.54	357.57 ± 25.99	371.77 ± 0.57
			1		397.91 ± 1.22	397.95 ± 0.47	330.70 ± 15.01	370.88 ± 0.57
		CAFO	60	460.87 ± 0.79	424.31 ± 1.83	423.79 ± 0.55	376.64 ± 19.28	385.75 ± 0.53
			1		419.16 ± 1.18	419.65 ± 0.68	347.31 ± 26.79	384.04 ± 0.48
EEHHPHHEPHMaMeP	1800	FOCA	60	377.11 ± 0.59	368.84 ± 1.34	368.17 ± 0.60	305.37 ± 54.01	328.69 ± 0.50
			1		362.11 ± 1.06	361.98 ± 0.57	303.73 ± 28.04	318.10 ± 0.53
		CAFO	60	385.29 ± 0.63	373.67 ± 1.57	374.10 ± 0.92	326.20 ± 20.51	331.32 ± 0.49
			1		369.21 ± 1.04	369.55 ± 0.57	298.07 ± 28.69	324.50 ± 0.47
EPMePHPEEPHHPHE	1800	FOCA	60	433.36 ± 0.80	420.38 ± 1.48	422.20 ± 0.60	364.85 ± 15.43	378.94 ± 0.44
			1		419.78 ± 1.10	420.13 ± 0.67	355.33 ± 28.56	377.88 ± 0.44
		CAFO	60	453.37 ± 0.73	443.25 ± 1.43	442.58 ± 0.77	373.79 ± 19.89	399.29 ± 0.63
			1		440.83 ± 1.13	440.67 ± 0.72	376.78 ± 18.56	398.78 ± 0.53

Table 5.1 – Numerical results for the buffer time allocation problem

5.4.3 Comments

The results are very convincing since the best upper bound (best feasible solution) for both versions is lower than the average waiting time given by the current solution. For the first version, the lower and upper bounds are very close for all the instances with and the confidence intervals are very tight. The lower and upper bounds and thus very accurate and give a good approximation of the optimal value.

For the second version, since the upper bounds are compute with the same number of samples Λ and Z than for the first version and since their computation is almost immediate, the confidence intervals are as tight as the first version. They also provide a significant improvement compared to the first version for some instances. However, the confidence interval on the lower bounds are not as tight since their computation is much slower than for the first version and the accurateness of the values is thus lower.

For both versions, there is an improvement of having a discretization fixed to one second compared to one minute, but this improvement is very small.

5.5 A related problem

5.5.1 Problem

We describe in this section the problem of computing a cyclic schedule with the two lexicographic objectives of maximizing the number of HGV shuttles and minimizing the expected value of the average delay of each train. On the one hand, Eurotunnel wants to maximize the number of HGV shuttles in a cyclic schedule, as explained in Chapter 4. On the other hand, operators like Eurotunnel want to compute schedules minimizing the expected value of the average delay of each train for H consecutive periods. The constraints are the same as the ones described in Sections 3.2 and 5.1. Additionally, we want to guarantee a certain amount of buffer time ρ (usually 5% or 10% of the size of the period) in the schedule: the sum of the security headways between the scheduled departures must not exceed $(1-\rho)T$. Indeed, a schedule without buffer times is not feasible in practice. According to the period of the day for example, operators sometimes require a better punctuality of the trains and thus want to guarantee more buffer times in the schedules. For some periods where the disturbances are low, they can require more trains and thus less buffer times. This constraint cannot be modelled as a linear constraint since the order of the departures in an optimal solution is not known *a priori*.

5.5.2 Method

The method to solve the problem is a heuristic computing good feasible solutions. The idea of the heuristic is first to compute a deterministic cyclic schedule on a period of length $(1-\rho)T$ maximizing the number of HGV shuttles (with the method described in Chapter 4 and with a parameter $(1-\rho)C^{\text{Eur}}$ for the commercial agreements with Eurostar). We use then the output of the previous algorithm to compute a cyclic schedule of a period of length T minimizing the expected value of the average delay of each train for H consecutive periods. We either keep the same order of the trains (for which we use the model given in Section 5.2.1) or only the number of different trains (for which we use the model given in Section 5.3.1).

5.5.3 Numerical results

The parameters for this problem are the same ones as in Table 4.1 and in Section 5.1, with a discretization η always set to 1.

Results

Table 5.2 gives the numerical results for this heuristic. The first four columns give respectively the number of Eurostars, ME120 freight trains, MA100 freight trains, and PAX shuttles in the instance. The next two columns are the parameter C^{Eur} the direction (FOCA or CAFO), and the next one is the value of ρ. The next column gives the maximum number of HGV shuttles in the schedule. As for Table 4.1, the next two columns give respectively the 95%-confidence intervals of an upper bound on the optimal value respectively for the first version and the second version.

Instance								Version 1	Version 2
N^{Eur}	N^{ME120}	N^{MA100}	N^{PAX}	C^{Eur}	Dir	ρ	N^{HGV}	Average delay	Average delay
1	0	0	5	3600	FOCA	0	10	1074.50 ± 3.03	990.71 ± 1.74
						5	8	501.12 ± 0.80	447.92 ± 0.68
						10	8	491.07 ± 0.91	447.58 ± 0.59
						15	8	451.70 ± 0.71	444.62 ± 0.69
						20	7	400.41 ± 0.57	367.99 ± 0.42
2	1	0	4	3600	CAFO	0	10	1061.37 ± 2.44	1006.97 ± 2.67
						10	8	571.34 ± 1.32	451.00 ± 0.50
						20	7	423.50 ± 0.70	383.47 ± 0.64
3	1	1	3	1800	FOCA	0	8	704.68 ± 2.11	504.44 ± 1.03
						5	8	706.70 ± 1.59	503.72 ± 1.00
						10	8	581.09 ± 1.25	506.27 ± 0.90
						20	6	421.82 ± 0.66	354.23 ± 0.53
4	1	0	5	1800	CAFO	0	8	770.00 ± 1.96	679.82 ± 1.20
						10	8	681.71 ± 1.03	668.03 ± 1.01
						15	6	553.10 ± 1.07	497.18 ± 1.02
						20	6	498.08 ± 1.05	495.74 ± 0.87

Table 5.2 – Numerical results for the heuristic

Comments

First note that $\rho = 0$ corresponds to the model where no buffer time is guaranteed *a priori*. The first part of the heuristics is then exactly the problem described in Chapter 4 with a cyclic period of length T. The number of scheduled HGV shuttles in thus larger than for other values of ρ but the average delay is also much larger.

First note that the average delays with the second version of the algorithm are much lower than the ones with the first version. The gain is about 30% for some instances.

For the first instance with 1 Eurostar and 5 PAX shuttles, the maximum number of HGV shuttles with 5, 10 and 15% of guaranteed buffer times is equal to 8, and it is equal to 7 for a schedule with 20% of guaranteed buffer time. For the first version of the problem minimizing average delay with the order of the trains in input, the results seem surprising: for the first instance with 5, 10 and 15% of buffer times, while the input is the same (1 Eurostar, 5 PAX shuttles) and the output of the first part of the algorithm is 8 HGV shuttles for all these values of ρ, the average delay decreases when ρ increases. The order of the trains plays a major role in the second part of the algorithm for the average delay. It seems that for large values of ρ, the first part of the algorithm computes a schedule with an order of the trains which leads to smaller average delays than for small values of ρ, whereas this objective is not taken into account at this step. We did not manage to give a satisfactory explanation to this. For the second version of the algorithm, the order of the trains is recomputed in the second part of the algorithm and thus, these instances give comparable average delays.

We performed complementary experiments on the heuristic. The results show that the first version of the algorithm cannot lead to smaller average delays than the ones given in Table 5.2 by computing lower bounds. For the second version, the results show that the upper bounds given in Table 5.2 may be improved, but they still give much smaller average delays than the first version.

Literature review. The robustness of train schedules is a major issue and has been studied a lot. Odijk [54, 55] used the PESP (see Chapter 4) with the objective of computing a feasible stable timetable. Stable meant reliable and low sensitive in minor operational online changes. The stochastic nature of train transportation had to be taken into account. Kroon and al. [39, 41] imagined a method to improve existing cyclic railway timetables to take train delays into account. Our work is inspired by these two articles. Their work is based on MIP models, Sample Average Approximation and simulation to provide a schedule minimizing the expected value of delays for all the trains. Cacchiani and Thot [14] and Fischetti et al. [26] also gave advanced methods based on heuristics and bi-criteria Lagrangian to compute robust timetables. Liebchen [47] and Vromans et al.[70] modelled with a Mixed Integer Programs the problem on computing delay resistant railway timetables. The allocation of supplements buffer time was described a lot (see Vromans [71]).

The improvement of the reliability of railway schedules can also be done with the Max-Plus algebra, described by Goverde [27, 28] and de Kort [21]. Other general analytic methods have been studied (see Peterson and Taylor [57], Hallowell and Harker [30], Higgins and Kozan [32], Carrey [17], Huisman and Boucherie [34], and Yuan's PhD thesis [72]).

Scheduling and pricing Part II

6 A joint scheduling and pricing problem

Consider a one-way trip from an origin station to a final one. The aim of the work in this chapter is to schedule a fixed number of shuttles on a period of time and to fix their prices.

We make the following assumptions. Each customer wants to purchase a ticket for this trip and has a preferred departure time in the period. He is also sensitive to the price of the trip. There are several economic classes of customers. For example, a business customer is much less sensitive to the price than a tourist, but much more sensitive to his departure time. A customer always purchases a ticket for the departure that satisfies him the most, if there is one, or does not buy any if none of the departures suits him. There is also a capacity constraint and thus, he cannot buy a ticket for a departure that is already full. We consider that the customers who are the most sensitive to the prices purchase their ticket first. The operator knows the total number of customers, the repartition of the classes, and the probability distribution of their preferred departure times.

What is the schedule and what are the prices of the tickets that maximize the expected revenue for the operator? This is the question addressed in this chapter.

This chapter gives a model for this problem and a Sample Average Approximation (SAA) method to solve it. A relaxation algorithm which gives upper bounds is also proposed, as well as two heuristics to compute good feasible solutions. Numerical experiments are performed to test the model and the heuristics.

6.1 Problem

We are given a fleet of N shuttles, all having a capacity $C \geq 0$, and situated at the origin station ready to leave. We want to determine their departure times d_j for all $j \in \{1, \ldots, N\}$ on a period of time modelled as the interval $[0, T]$. The departures have to satisfy the security constraint: two consecutive departures must be separated by a minimum security headway $K \geq 0$. We also want to fix the price $p_j \in \mathbb{R}_+$ of each departure j. Note that the price for each departure is common to all customers. We define a *product* as a pair (d_j, p_j). A feasible solution of this problem is characterized by a set of N products $(\boldsymbol{d}, \boldsymbol{p}) = (d_j, p_j)_{j \in \{1,\ldots,N\}}$.

We are given I customers. Each of them has to choose at most one product among the ones that are offered by the operator. A fake "opt-out" product is added to capture the customers who do not want to purchase any product. Each customer i has a preferred departure time, modelled as a random variable $\chi_i \in [0, T]$ at which he prefers to leave. The random variables χ_i are supposed to be independent. He also belongs to a class $b_i \in \{1, \ldots, B\}$ of customers, each class b having its own "value of time" $v_b \in [0,1]$. We assume that $v_1 \le v_2 \le \cdots \le v_B$. Customers of class b make their choice before those of class $b' > b$. In a given class b, the customers make their choice in a random order. The "opt-out" product is identified to a product indexed by 0 and with a price $p_0 = 0$.

We follow the traditional methodology of discrete choice models, explained in Section 6.1. For each customer $i \in \{1, \ldots, I\}$ and each product $j \in \{1, \ldots, N\}$, we define a *utility* $U_{ij} = V_{ij} + \xi_{ij}$ where

- V_{ij} is the deterministic part defined as

$$V_{ij} = A - v_{b_i} g_i(d_j) - (1 - v_{b_i}) p_j \tag{6.1}$$

  with

$$g_i(d) = h \max(0, d - \chi_i) + \max(0, \chi_i - d) \tag{6.2}$$

  and A some positive real constant,
- ξ_{ij} is a random variable following a Gumbel distribution (see Section 6.1).

We also set $U_{i0} = \xi_{i0}$. The probability that customer i chooses product j is

$$P_{ij}(\boldsymbol{d}, \boldsymbol{p}) = \mathbb{P}\left(U_{ij} = \max\{U_{ik} : \text{ for all } k \text{ not full}\}\right).$$

It means that every customer i is utility-maximizing: he chooses the product $j \in \{0, \ldots, N\}$ which maximizes his utility function U_{ij}, among all the departures which are available (i.e. which are not full). We assume the "opt-out" to have an infinite capacity.

We assume that for all j and k, the probability $\mathbb{P}\left(U_{ik} = U_{ij}\right)$ is equal to 0, which implies in particular that if $P_{ij}(\boldsymbol{d}, \boldsymbol{p}) > 0$, then the departure j is not full.

The problem can be written as the following program

$$\begin{array}{rlcll} \mathrm{Max}_{\boldsymbol{d},\boldsymbol{p}} & \displaystyle\sum_{j=1}^{N} p_j \sum_{i=1}^{I} P_{ij}(\boldsymbol{d}, \boldsymbol{p}) & & & \\ \text{s.t.} & d_j & \ge & d_{j-1} + K & j \in \{2, \ldots, N\} \\ & d_j & \in & [0, T] & j \in \{1, \ldots, N\} \\ & p_j & \in & \mathbb{R}_+ & j \in \{1, \ldots, N\}. \end{array} \tag{6.3}$$

We present in Section 6.2 a first special case for which the probabilities $P_{ij}(\boldsymbol{d}, \boldsymbol{p})$ take an explicit form. We give in Section 6.3 a Mixed Integer Piecewise Affine Program to model this problem and a method based on SAA to solve it. We then give a natural heuristic to compute feasible solutions in Section 6.5. This heuristic mimics what could be a natural way of scheduling the departure times and fixing their prices. Another heuristic based of a Gauss-Seidel algorithm in presented in Section 6.6.

Discrete choice model

This section gives complementary explanations on the discrete choice model used for the utility functions. The choices depend on many factors: some of them are observable and others are not. The utility U_{ij} is decomposed in an observable deterministic part V_{ij} and a nonobservable stochastic one ξ_{ij}. The error term ξ_{ij} is a random variable which captures everything that cannot be explicitly modelled by the operator. The random variables representing the error terms ξ_{ij} are supposed to be independent and to follow a standard Gumbel distribution. This is a classical assumption in the discrete choice model theory since it leads to closed formulas of the choice probabilities in some cases (see Section 6.2).

The utility V_{ij} depends on the customer and on the product, and captures everything that is observable. The observable utility V_{i0} of the "opt-out" is set to 0, whereas for every other product, V_{ij} is a function of the departure time d_j and the price p_j defined in Equation (6.1). The positive real constant A represents the value of the "initial satisfaction" of the customers. The function $g_i(\cdot)$ represents the weighted delay of a departure time compared to the preferred departure time χ_i. The value $h \in \mathbb{R}_+$ represents the importance of a delay compared to a departure in advance; it is the same for all customers. A value $h < 1$ means that leaving at a departure time $d_j > \chi_i$ is more acceptable than leaving at $d_j < \chi_i$. A value $h > 1$ means that leaving in advance is more acceptable than leaving with a delay (the most common case in practice). The values $v_b \in [0,1]$ are the "values of time" and give the part of the total sensitiveness of the customers of class b that is for the departure time: a value $v_b = 0$ means that the customers of class b are not sensitive to the departure time and a value $v_b = 1$ means that they are not sensitive to the prices.

6.2 A first special case

We have been able to identify a special case for which their is an explicit form of the objective function. In an even more specific case, we are able to give an explicit solution. We assume $C = +\infty$, $K = 0$, and that the random variables χ_i are deterministic. With these assumptions, it is known that $P_{ij}(\boldsymbol{d},\boldsymbol{p})$ has an explicit expression (see Ben-Akiva and Bierlaire [4], Ben-Akiva et al. [5], Bussieck et al. [11], and Train [66]):

$$P_{ij}(\boldsymbol{d},\boldsymbol{p}) = \frac{e^{V_{ij}}}{\sum_{k=0}^{N} e^{V_{ik}}} = \frac{e^{V_{ij}}}{1+\sum_{k=1}^{N} e^{V_{ik}}}.$$

Program (6.3) can be expressed as follows:

$$
\begin{array}{lrcl}
\text{Max}_{\boldsymbol{d},\boldsymbol{p}} & \displaystyle\sum_{j=1}^{N} p_j \sum_{i=1}^{I} \frac{e^{V_{ij}}}{1+\sum_{k=1}^{N} e^{V_{ik}}} & & \\
\text{s.t.} & V_{ij} & = & A - v_{b_i} g_i(d_j) - (1-v_{b_i}) p_j \quad i \in \{1,\dots,I\}, j \in \{1,\dots,N\}.
\end{array}
$$

where the function $g(\cdot)$ is defined in Equation (6.2).

Note that already in this case, the objective function has many local maxima. It explains why it is difficult to solve to the optimum. Consider the special case where there is one single customer (i.e. $I = 1$). The problem becomes

$$
\begin{array}{lrcl}
\text{Max}_{\boldsymbol{d},\boldsymbol{p}} & \displaystyle\sum_{j=1}^{N} p_j \frac{e^{V_j}}{1+\sum_{k=1}^{N} e^{V_k}} & & \\
\text{s.t.} & V_j & = & A - v_b g(d_j) - (1-v_b) p_j \quad j \in \{1,\dots,N\}.
\end{array}
\tag{6.4}
$$

Let $W_0(\cdot)$ be the unique function satisfying $W_0(x)e^{W_0(x)} = x$ for all $x \geq -1/e$ (this function is the "upper branch of the Lambert-W function" (see [20])).

Proposition 2. *If $v_b < 1$, the solution $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{p}})$ with $(\tilde{d}_j, \tilde{p}_j) = \left(\chi_1, \dfrac{1+W_0(Ne^{A-1})}{1-v_b}\right)$ for all $j \in \{1,\dots,N\}$ is optimal for program (6.4) and the optimal value is $\frac{W_0(Ne^{A-1})}{1-v_b}$. Otherwise, there is no optimal solution and the optimal value is $+\infty$.*

Proof. We denote by $f(\boldsymbol{d},\boldsymbol{p})$ the objective function. We assume first that $v_b < 1$. Since the function $x \mapsto \frac{e^x}{a+e^x}$ is increasing (where a is a constant), and since V_{ij} is maximal for $d_j = \chi_i$, it is clear that for all j, $d_j = \chi_i$ for any optimal solution. Moreover, the function $f(\cdot,\cdot)$ is differentiable for all p_j and we have

$$
\frac{\partial f}{\partial p_j}(\boldsymbol{d},\boldsymbol{p}) = \frac{e^{V_j}}{\left(\sum_{k=1}^{N} e^{V_k}\right)^2}\left((1-(1-v_b)p_j)(1+\sum_{k=1}^{N} e^{V_k}) + (1-v_b)\sum_{k=1}^{N} p_k e^{V_k}\right).
$$

If a solution $(\boldsymbol{d},\boldsymbol{p})$ is an optimal solution, then $\nabla f = 0$ and then for all j, we have $(1-(1-v_b)p_j)(1+\sum_{k=1}^{N} e^{V_k}) + (1-v_b)\sum_{k=1}^{N} p_k e^{V_k} = 0$. Subtracting this equality for all j and j' gives $p_j = p_{j'}$. Problem (6.4) becomes

$$
\text{Max}_{p \in \mathbb{R}_+} \phi(p),
$$

where

$$
\begin{array}{rccc}
\phi : & \mathbb{R}_+ & \longrightarrow & \mathbb{R}_+ \\
 & p & \longmapsto & p\frac{Ne^{A-(1-v_b)p}}{1+Ne^{A-(1-v_b)p}}.
\end{array}
$$

Let $\bar{p} \in \mathbb{R}_+^*$. There exists $\hat{p} \in \mathbb{R}_+$ such that for all $p > \hat{p}$, $0 < \phi(p) < \phi(\bar{p})$ since $\lim\limits_{p\to+\infty} \phi(p) = 0$. Thus, $\phi(\cdot)$ can be restricted to $\phi\colon [0,\hat{p}] \to \mathbb{R}_+$ and has a maximum value for $p = p^*$ where

$p^* \in (0, \hat{p})$. The function $\phi(\cdot)$ is differentiable and

$$\phi'(p) = \frac{Ne^{A-(1-v_b)p}(1 + Ne^{A-(1-v_b)p} - (1-v_b)p)}{(1 + Ne^{A-(1-v_b)p})^2}.$$

Since p^* is a local maximum, we have necessarily

$$1 + Ne^{A-(1-v_b)p^*} - (1-v_b)p^* = 0. \tag{6.5}$$

Denoting $X = Ne^{A-(1-v_b)p^*}$, we have $Xe^X = Ne^{A-1}$ and thus $X = W_0(Ne^{A-1})$ because $Ne^{A-1} \geq 0$. Then there is a unique possible value for p^* and $p^* = \frac{A - \ln N - \ln W_0(e^{A-1})}{1-v_b}$ which gives $p^* = \frac{1+W_0(Ne^{A-1})}{1-v_b}$.

If $v_b = 1$, $f(\cdot,\cdot)$ converges to $+\infty$ when p_j tends to $+\infty$ and the optimal value in $+\infty$. $\square$

6.3 Model and Sample Average Approximation

We provide an alternate expression for the objective function which will turn to be useful. We denote by $\boldsymbol{\xi} = (\xi_{ij})$, and by $\boldsymbol{\chi} = (\chi_i)$. Let X be the set of feasible solutions

$$X = \left\{(\boldsymbol{d}, \boldsymbol{p}) \in [0, T]^N \times \mathbb{R}_+^N : \; d_j \geq d_{j-1} + K, j \in \{2, \dots, N\}\right\}.$$

Program (6.3) can be equivalently written as a two-stage recourse program

$$\text{Max}_{(\boldsymbol{d},\boldsymbol{p})\in X} f(\boldsymbol{d}, \boldsymbol{p}) := \mathbb{E}\left[Q(\boldsymbol{d}, \boldsymbol{p}, \boldsymbol{\chi}, \boldsymbol{\xi})\right]. \tag{6.6}$$

Here, $Q(\boldsymbol{d}, \boldsymbol{p}, \boldsymbol{\chi}, \boldsymbol{\xi})$ is the recourse function defined as

$$Q(\boldsymbol{d}, \boldsymbol{p}, \boldsymbol{\chi}, \boldsymbol{\xi}) = \sum_{i=1}^{I} \sum_{j=1}^{J} p_j w_{ij} \tag{6.7}$$

where the w_{ij}'s are solutions of the following system

$$\begin{cases}
y_{ij} = \begin{cases} 1 \text{ if } \sum_{i'=1}^{i-1} w_{ij} < C \\ 0 \text{ otherwise} \end{cases} & i \in \{1, \dots, I\}, j \in \{1, \dots, J\} \\
y_{i0} = 1 & i \in \{1, \dots, I\} \\
U_{ij} = A - v_{b_i} g_i(d_j) - (1 - v_{b_i}) p_j + \xi_{ij} & i \in \{1, \dots, I\}, j \in \{1, \dots, J\} \\
U_{i0} = \xi_{i0} & i \in \{1, \dots, I\} \\
w_{ij} = \begin{cases} 1 \text{ if } U_{ij} = \max\left\{U_{ik} : \text{ for all } k \text{ such that } y_{ik} = 1\right\} \\ 0 \text{ otherwise.} \end{cases} & i \in \{1, \dots, I\}, j \in \{1, \dots, J\}
\end{cases} \tag{$\mathscr{S}$}$$

The functions $g_i(\cdot)$ are defined in Equation (6.2). The variables y_{ij} represent the availability of product j for customer i, the variables U_{ij} are the utilities and the variables w_{ij} represent the choice of customer i for product j. The variables $\boldsymbol{d}$ and $\boldsymbol{p}$ are fixed before knowing the realization of the random variables $\boldsymbol{\chi}$ and $\boldsymbol{\xi}$.

As we did in Chapter 5, we can solve program (6.6) with a Sample Average Approximation method as follows. We generate samples $(\boldsymbol{\chi}_1,\dots,\boldsymbol{\chi}_R)$ and $(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_R)$ of R realizations of the random variables. We can approximate the objective function $f(\cdot,\cdot)$ and program (6.6) becomes

$$\mathrm{Max}_{(\boldsymbol{d},\boldsymbol{p})\in X}\, f_R(\boldsymbol{d},\boldsymbol{p}) := \frac{1}{R}\sum_{r=1}^{R} Q(\boldsymbol{d},\boldsymbol{p},\boldsymbol{\chi}_r,\boldsymbol{\xi}_r), \tag{MIPAP}$$

where $Q(\cdot,\cdot,\cdot,\cdot)$ is the recourse function defined in (6.7). The recourse function can easily be determined for fixed $\boldsymbol{\chi}$ and $\boldsymbol{\xi}$. This is because $(\mathscr{S})$ can be described by piecewise affine inequalities, making it accessible to standard solvers.

As for Proposition 1, we have the convergence with probability 1 of $f_R(\boldsymbol{d},\boldsymbol{p})$ to $f(\boldsymbol{d},\boldsymbol{p})$ as R goes to $+\infty$, for all $(\boldsymbol{d},\boldsymbol{p})\in X$. The proof is obtained with the same arguments.

As it is described in Section 5.2.2, an upper bound on the optimal solution of program (6.6) is be computed by generating Ω samples of $(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_R)$ and solving program (MIPAP) for some predetermine Ω. We can estimate this upper bound with a $(1-\alpha)$-confidence interval. For each of these Ω feasible solutions, Z samples of size Λ, $(\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_\Lambda)$, give an estimation of its value (and thus a lower bound on the optimal value) with a $(1-\alpha)$-confidence interval. We return the feasible solution that has the largest estimated value.

6.4 Lagrangian relaxation

As an alternative way to get upper bounds on the optimal value, we propose a Lagrangian method. We use a standard trick: we add the variables $\boldsymbol{d}_{jr}$ and $\boldsymbol{p}_{jr}$ to model the departure times and prices for each scenario r. We add the constraints $d_{jr}=d_j$ and $p_{jr}=p_j$ to ensure that the departure times and prices are the same for all scenarios, and the constraints $d_{jr}\geq d_{j-1,r}+K$ to ensure that the departures satisfy the security constraint. Let us rewrite program (MIPAP) with these new variables:

$$\begin{array}{lrcll}
\mathrm{Max}_{(\boldsymbol{d},\boldsymbol{p})\in X} & \multicolumn{4}{l}{\dfrac{1}{R}\sum_{r=1}^{R} Q(\boldsymbol{d}_r,\boldsymbol{p}_r,\boldsymbol{\chi}_r,\boldsymbol{\xi}_r)} \\
\text{s.t.} & d_j & = & d_{jr} & j\in\{1,\dots,N\}, r\in\{1,\dots,R\} \quad \text{(i)} \\
 & p_j & = & p_{jr} & j\in\{1,\dots,N\}, r\in\{1,\dots,R\} \quad \text{(ii)} \\
 & d_{jr} & \geq & d_{j-1,r}+K & j\in\{2,\dots,N\}, r\in\{1,\dots,R\} \quad \text{(iii)} \\
 & d_{jr} & \in & [0,T] & j\in\{1,\dots,N\}, r\in\{1,\dots,R\} \quad \text{(iv)} \\
 & p_{jr} & \in & \mathbb{R}_+ & j\in\{1,\dots,N\}, r\in\{1,\dots,R\} \quad \text{(v)}.
\end{array} \tag{6.8}$$

We define $\boldsymbol{\lambda}\in\mathbb{R}^N\times\mathbb{R}^R$ and $\boldsymbol{\mu}\in\mathbb{R}^N\times\mathbb{R}^R$ to be the Lagrangian multipliers respectively associated with constraints (i) and (ii). The Lagrangian relaxation is

$$\begin{array}{rclllll} L_R(\boldsymbol{\lambda},\boldsymbol{\mu}) & = & \sup_{(\boldsymbol{d},\boldsymbol{p})\in X} & \multicolumn{4}{l}{\frac{1}{R}\sum_{r=1}^{R} Q(\boldsymbol{d}_r,\boldsymbol{p}_r,\boldsymbol{\chi}_r,\boldsymbol{\xi}_r) + \sum_{j=1}^{N}\sum_{r=1}^{R}\big(\lambda_{jr}(d_{jr}-d_j)+\mu_{jr}(p_{jr}-p_j)\big)} \\ & & \text{s.t.} & d_{jr} & \geq & d_{j-1,r}+K & j\in\{2,\dots,N\}, r\in\{1,\dots,R\} \\ & & & d_{jr} & \in & [0,T] & j\in\{1,\dots,N\}, r\in\{1,\dots,R\} \\ & & & p_{jr} & \in & [0,P^+] & j\in\{1,\dots,N\}, r\in\{1,\dots,R\}. \end{array}$$

This program is decorrelated for the R scenarios and can be expressed as follows.

$$\begin{aligned} L_R(\boldsymbol{\lambda},\boldsymbol{\mu}) \quad = \quad & \frac{1}{R}\sum_{r=1}^{R}\sup_{(\boldsymbol{d}_r,\boldsymbol{p}_r)\in X}\left(Q(\boldsymbol{d}_r,\boldsymbol{p}_r,\boldsymbol{\chi}_r,\boldsymbol{\xi}_r)+\sum_{j=1}^{N}(\lambda_{jr}d_{jr}+\mu_{jr}p_{jr})\right) \\ & -\inf_{(\boldsymbol{d},\boldsymbol{p})\in X}\sum_{j=1}^{N}\left(\sum_{r=1}^{R}\lambda_{jr}d_j+\sum_{r=1}^{R}\mu_{jr}p_j\right). \end{aligned} \tag{6.9}$$

We denote v_R^* the optimal value of program (MIPAP). For any $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, the value $L_R(\boldsymbol{\lambda},\boldsymbol{\mu})$ provides an upper bound on v_R^*. To obtain the sharpest possible upper bound, we solve the following Lagrangian Dual program:

$$L_R^* = \inf_{\boldsymbol{\lambda},\boldsymbol{\mu}} L_R(\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{6.10}$$

We have that the optimal value L_R^* of the Lagrangian Dual program (6.10) provides an upper bound on the value v_R of any feasible solution of program (MIPAP):

$$v_R \leq v_R^* \leq L_R^* \leq L_R(\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{6.11}$$

Note that the primal program (MIPAP) has integer variables and there is thus no reason that $L_R^* = v_R^*$.

To solve the Lagrangian Dual program (6.10), we use a subgradient method (see [6]). At any step k, from the current state vector $(\boldsymbol{\lambda}^k,\boldsymbol{\mu}^k)$, we compute a subgradient $\nabla L_R(\boldsymbol{\lambda}^k,\boldsymbol{\mu}^k) = \Big((d_{jr}^{*k}-d_j^{*k}),(p_{jr}^{*k}-p_j^{*k})\Big)$, where $d_{jr}^{*k}, d_j^{*k}, p_{jr}^{*k}$, and p_j^{*k} are the optimal solutions of (6.9). We then update the Lagrangian multipliers $\boldsymbol{\lambda}^k$ and $\boldsymbol{\mu}^k$ as follows:

$$\begin{aligned} \lambda_{jr}^{k+1} &\longleftarrow \lambda_{jr}^{k}-\theta_k(p_{jr}^{*k}-p_j^{*k}) \\ \mu_{jr}^{k+1} &\longleftarrow \mu_{jr}^{k}-\theta_k(d_{jr}^{*k}-d_j^{*k}). \end{aligned}$$

The step θ_k at each iteration specifies how far we move in the opposite direction of a subgradient. To ensure that this method converges, it is known that the following convergences are required.

$$\theta_k \underset{k\to+\infty}{\longrightarrow} 0 \quad \text{and} \quad \sum_{i=1}^{k} \theta_k \underset{k\to+\infty}{\longrightarrow} +\infty.$$

Choosing $\theta_k = \frac{1}{k}$ would work but leads to a bad convergence speed in certain cases. We choose the following heuristic for selecting the step:

$$\theta_k = \alpha_k \frac{L_R(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) - \mathrm{LB}}{\|\nabla L_R(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)\|^2},$$

where α_k is a scalar chosen between 0 and 2, empirically chosen equal to 0.5×0.75^k, and LB is the value of the best-known feasible solution of program (6.9). We stop the algorithm when there is no improvement of $\hat{L}_R$ and when the value $|L_R(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k) - \hat{L}_R|$ is nonlarger than $\hat{L}_R/100$ for 10 consecutive iterations, where

$$\hat{L}_R = \min_{\ell \in \{1,\dots,k\}} L_R(\boldsymbol{\lambda}^\ell, \boldsymbol{\mu}^\ell).$$

The estimation of a confidence interval for the upper bound is done the same way as for the previous model given in Section 6.3.

Note that we can easily build feasible solutions $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{p}})$, which gives raise to a Lagrangian heuristic. At each iteration of the Lagrangian relaxation, we draw uniformally at random the departure times and prices such that

$$(\hat{\boldsymbol{d}}, \hat{\boldsymbol{p}}) \in \mathrm{conv}_{r \in \{1,\dots,R\}} (\boldsymbol{d}_r, \boldsymbol{p}_r).$$

6.5 Sequential heuristic

As mentioned in the introduction, we try here to mimic what would be a natural way of scheduling the departure times and fixing their prices. First, we compute the departure times d_j's. To do that, we use a surrogate optimization program minimizing the sum of the weighted delays $g_i(d_j)$ for each customer i.

$$\begin{array}{llll}
\text{Min} & \mathbb{E}\left[\displaystyle\sum_{i=1}^{I}\sum_{j=1}^{N} g_i(d_j) x_{ij}\right] & & \\
\text{s.t.} & \displaystyle\sum_{j=1}^{N} x_{ij} \;\geq\; 1 & & i \in \{1,\dots,I\} \\
& d_j \;\geq\; d_{j-1} + K & & j \in \{0,\dots,N\} \\
& d_j \;\in\; [0,T] & & j \in \{0,\dots,N\} \\
& x_{ij} \;\in\; \{0,1\} & & i \in \{1,\dots,I\}, j \in \{0,\dots,N\}.
\end{array} \tag{6.12}$$

The variables x_{ij} model the choice of customer i. This program can be linearized with standard techniques. The rational is to define the departure times which satisfy the customers the best.

Second, we compute the prices p_j's with program (MIPAP) as in Section 6.3, where the d_j's are fixed.

6.6 Gauss-Seidel heuristic

This algorithm is inspired from the block nonlinear Gauss-Seidel algorithm. Let m be any positive integer and $J_1 \cup J_2 \cup \cdots \cup J_m$ a partition of $\{1,\ldots,N\}$ such that their cardinalities differ by at most one.

We initialize $(\boldsymbol{d},\boldsymbol{p})$ to some $(\boldsymbol{d}^0,\boldsymbol{p}^0)$. The algorithm generates a sequence of feasible solutions $(\boldsymbol{d}^k,\boldsymbol{p}^k)$. The solution $(\boldsymbol{d}^{k+1},\boldsymbol{p}^{k+1})$ is computed using the solution $(\boldsymbol{d}^k,\boldsymbol{p}^k)$ of previous step by solving

$$\begin{aligned}\big((d_j^{k+1})_{j\in J_\ell},(p_j^{k+1})_{j\in J_\ell}\big) \in \operatorname{argmax}_{\big((d_j)_{j\in J_\ell},(p_j)_{j\in J_\ell}\big)}\\ f\Big(\big((d_j^{k+1})_{j\in J_1},\ldots,(d_j)_{j\in J_\ell},\ldots(d_j^k)_{j\in J_m}\big),\big((p_j^{k+1})_{j\in J_1},\ldots,(p_j)_{j\in J_\ell},\ldots(p_j^k)_{j\in J_m}\big)\Big)\end{aligned}$$

for all $\ell \in \{1,\ldots,m\}$. The function $f(\cdot)$ is defined in (6.6) and the maximum is taken among the $\big((d_j)_{j\in J_\ell},(p_j)_{j\in J_\ell}\big) \in [0,T]^{J_\ell} \times \mathbb{R}_+^{J_\ell}$ such that $d_{j+1} \geq d_j + K$.

The value of m is empirically fixed equal to $\max(1,\lceil IN/5\rceil)$ when $I \leq 5$ (i.e. $m \leq N$). If $I > 5$, we use a variant where (d_j^{k+1},p_j^{k+1}) is computed by solving a similar program for each variable sequentially (i.e. d_1^{k+1}, then p_1^{k+1}, then $d_2^{k+1},\ldots$).

We initialize $(\boldsymbol{d}^0,\boldsymbol{p}^0)$ either with the solution of sequential heuristic described in Section 6.5 or the best feasible solution of with the Lagrangian heuristic described in Section 6.4.

6.7 Numerical results

We tested the models on several instances. To compute the upper bounds with the MIPAP algorithm and the Lagrangian relaxation, we used the MIP solver CPLEX 12.6 with its standard settings. To compute the lower bounds with the MIPAP algorithm, the sequential heuristic, and the Gauss-Seidel heuristic, we used the same version of CPLEX with a special setting boosting the computation of feasible solutions (see Fischetti et al. [25]). All the experiments were performed on a MacBook Pro of 2014 with four 2.2 Ghz processors and 16 Gb of ram. The computation time is limited to one hour for the upper bounds, and to twenty minutes for the lower bounds.

6.7.1 Instances

We build our own instances since the problem is completely prospective for Eurotunnel. The size of the period is one hour ($T = 3600$ s) and the security headway is fixed to $K = 155$ s (corresponds to the security headway between two HGV shuttles for direction FOCA). The capacity of each train is fixed to $C = \lfloor I/N \rfloor$.

Utilities

The value h is fixed to 3/2 and the preferred departure times χ_i follow a customized probability distribution given in the following Figure 6.1. The economic classes of the customer are Business, Economic, and Low-Cost. We take one third of Low-Cost customers ($b = 1$), one third of Economic customers ($b = 2$), and one third of Business customers ($b = 3$). The values of v_b for each class are given in Table 6.1. The parameter A is equal to 15. The error terms ξ_{ij} follow a standard Gumbel distribution, represented in Figure 6.2.

Class	v_b
Low-cost	0.25
Economic	0.50
Business	0.75

Table 6.1 – Values of time for the different economic classes

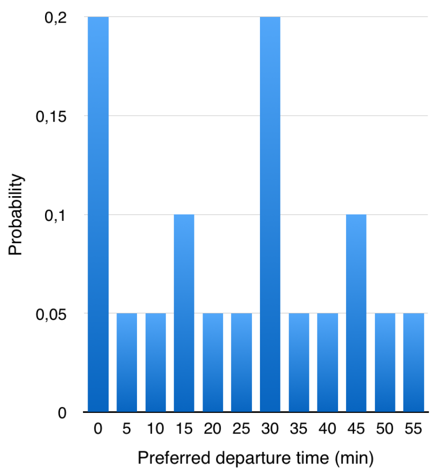

Figure 6.1 – Probability distribution of the preferred departure times

Confidence intervals

The number R of scenarios taken in the Sample Average Approximations varies according to the instance. Since the number of variables and constraints of the programs can be very large for some instances, the value R has to be fixed according to the number of

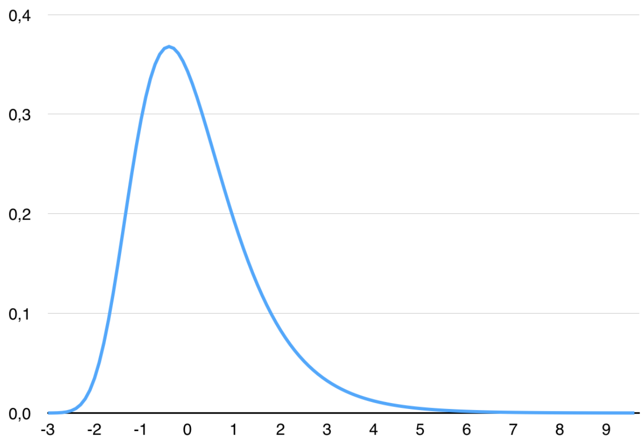

Figure 6.2 – Probability density function of standard Gumbel distribution

departures N and the number of customers I. We empirically fix R to $\lfloor 5000/(I^2N^2)\rfloor$ for the computation of the upper bounds and $\lfloor 200/(I^2N^2)\rfloor$ for the computation of the lower bounds. The number of samples Ω to estimate an upper bound is not fixed *a priori*, the algorithm computes as many as possible is the limited computation time. The size of the samples and the number of these samples generated to estimate a lower bound is respectively fixed to $\Lambda = 1000$ and $Z = 100$.

6.7.2 Results

Table 6.2 gives the numerical results for the joint scheduling and pricing problem. The first two columns are the instance and give respectively the number I of customers and the number N of trains. The next two columns are respectively the 95%-confidence intervals of a lower bound and an upper bound on the optimal value the Mixed Integer Piecewise Affine Program (MIPAP). The next two columns give the confidence interval on a lower bound respectively for the sequential heuristic and the Gauss-Seidel heuristic. The last column gives the confidence interval on the upper bound of the Lagrangian relaxation.

6.7.3 Comments

Several comments can be made in the light of these results. For some large instances, the confidence intervals for the upper bounds were so large that they had no sense compared to the lower bounds. The values in such cases were replaced by the upper bound of the confidence intervals with means that these values are upper bounds on the optimal value of the problem with a probability of 95%. The upper bounds computed by the Lagrangian relaxation are much worse than the ones computed with the MIPAP algorithm.

The feasible solutions computed with the sequential heuristic give values which are surprisingly close to values of the solutions computed with the MIPAP algorithm. These

Instance		Lower bounds			Upper bounds	
I	N	MIPAP	Seq. heur.	G-S heur.	MIPAP	Lag. relax.
1	1	8.3 ± 0.0	8.3 ± 0.0	8.3 ± 0.0	8.3 ± 0.0	14.7 ± 0.0
2	1	14.3 ± 0.0	14.1 ± 0.0	14.3 ± 0.0	14.7 ± 0.1	25.8 ± 0.1
10	1	66.0 ± 0.1	65.8 ± 0.1	66.0 ± 0.1	77.1 ± 0.9	96.6 ± 3.6
100	1	652.8 ± 0.8	651.4 ± 1.1	652.2 ± 1.3	769.6 ± 89.1	≤ 1050.4
200	1	1304.9 ± 1.8	1301.9 ± 2.0	1303.4 ± 1.8	≤ 1862.4	≤ 2668.3
2	2	17.5 ± 0.0	17.4 ± 0.0	17.4 ± 0.0	23.3 ± 2.1	33.4 ± 0.1
4	2	37.2 ± 0.0	36.8 ± 0.0	37.2 ± 0.0	48.9 ± 2.2	63.4 ± 1.5
20	2	199.4 ± 0.1	198.6 ± 0.2	198.7 ± 0.2	305.0 ± 11.6	307.4 ± 31.0
100	2	986.9 ± 1.3	998.4 ± 1.1	1001.8 ± 1.2	≤ 2229.3	≤ 4008.6
50	3	557.2 ± 0.7	591.4 ± 0.5	593.0 ± 0.6	834.7 ± 103.8	847.5 ± 204.9
100	3	1159.6 ± 0.8	1162.7 ± 1.1	1165.5 ± 1.1	1847.4 ± 769.8	≤ 4302.1
100	4	1241.3 ± 0.8	1263.4 ± 1.5	1274.8 ± 1.0	1958.9 ± 385.5	≤ 2989.9
200	4	2567.5 ± 1.8	2531.5 ± 3.0	2568.9 ± 2.1	3944.5 ± 1127.6	≤ 6974.2
200	5	4290.7 ± 2.2	4353.1 ± 2.8	4461.1 ± 2.2	5957.1 ± 1339.7	≤ 9687.2
300	6	4445.3 ± 2.6	4471.4 ± 2.9	4589.6 ± 2.9	6579.6 ± 1209.3	≤ 9608.4

Table 6.2 – Numerical results for the combined scheduling and pricing problem

values are even better for some large instances. However, the Gauss-Seidel algorithm manages to improve these solutions for all the instances. The feasible solutions computed with this algorithm give values which are very close to the ones computed with the MIPAP algorithm for some small instances, and which are much better for some large instances. The values are also much better than the ones computed by the sequential heuristic with proves the gain of optimizing in a joint way the departure times and the prices.

Literature review. We give a review on discrete choice models and interactions between network design and pricing. We did not find any literature dealing with scheduling and pricing for rail transportation. Since network design and pricing are a lot studied in transportation theory, we naturally looked at this literature. The problem involves two different kinds of agents: the operator who designs the network and fixes the prices for all the products in order to maximize the profit of the company, and the customers who act to maximize their own profits individually. These two parts act non cooperatively and the company aims to fix high prices to increase their profit and the customers want to buy at low prices. They usually act sequentially and the operator models the behaviour of the customers in its optimization model. This problem was described in economy by Van Ackere [68] where a principal wants to hire an agent at the best salary. These two agents have antagonist objectives, the behaviour of the agent is modelled by the principal in his optimization process.

To model the customers' behaviours, we use a discrete choice model to describe and predict the choices of customers for a finite set of alternatives. Such methods contrast with continuous models for which the choices are continuous among a set of continuous alternatives. The discrete choice models have been studied a lot by Ben-Akiva and Lerman [5] and Ben-Akiva et al. [4]. Some sets of models use a deterministic utility function and probabilistic decisions such as Luce [49] or Tversky [67]. Another set of models uses random utility functions and deterministic choices to describe uncertainty. We use in our work the latter models with random utility functions and deterministic choices. The different methods and discrete choice models are described in [4].

Assortment optimization is a common problem faced by companies. It consists in selecting a finite set of products among a large panel to offer to a group of customers so as to maximize the revenue. This problem is very close to the want we present here, and was studied for example by Talluri and Van Ryzin [65], Bront et al. [8], and by Rusmevichientong et al. [59, 60].

Network design and pricing have mostly been addressed sequentially: the prices are optimized on a network computed in a first step. However, these two issues are linked and have to been treated in a joint way. This joint problem has already been studied by Lederer [44], and Brotcorne et al. [9, 10] for instance. Bertsimas [7] combined network pricing with resource allocation and Erdelyi et al. [24] used decomposition methods for pricing problems on a network.

Discrete choice models are very useful in train scheduling to represent the behaviour of customers during their trip. Robenek et al. [58] studied cyclic and noncyclic scheduling on a network where discrete choice models are used to represent the route taken by a customer in the rail network. Hetrakul et al. [31] studied the optimal seat allocation and pricing for railway transportation. We mainly use the work of Pacheco et al. [56] who modelled with Mixed Integer Programs customer choices. We add the scheduling issue to their work.

Part III

Minimizing the waiting time for a one-way shuttle service

7 A theoretical problem

7.1 Introduction

The original motivation of this chapter comes from our partnership with Eurotunnel. Eurotunnel is currently facing an increasing congestion due to the trucks waiting in the terminal before being loaded in the shuttles. A way to address this issue consists in scheduling the shuttles so that the trucks do not wait too long in the terminal.

Since the trip of the trucks in the tunnel is a small part of their whole journey, it is a reasonable approximation to assume that they cannot choose their arrival time in the terminal. Moreover, changing the size of the fleet in the day to feat the changes in the demand is not possible (the shuttles are expensive and the tunnel is used by other vehicles, which limits the maximal number of shuttle trips over a day). Similarly, the capacity of the shuttles cannot be changed (the length of the platform constrains their size). We face thus a different problem than the one addressed in the literature: the demand is assumed to be fixed and nonelastic to the departures times, and the capacity of shuttles, as well as their number, cannot be adjusted to the demand. Given a fleet of shuttles and a demand of transportation known in advance, the problem consists in designing a schedule for the shuttles that minimizes the waiting time of the users. Moreover, in the present work, the schedules have to be designed in an offline manner. In a transportation context, and especially for Eurotunnel, computing the schedule in advance is mandatory.

We study several versions of the problem, mainly according to two features. The first feature is whether the shuttles are allowed to come back at the terminal after having realized the trip. The second feature is the objective function. We consider in turn the following two quantities to be minimized: the maximum waiting time and the average waiting time. The first objective is perhaps a fairer one regarding the users, while the second one is relevant for the global efficiency.

It seems that the question we address in the present chapter is new. Moreover, it may be relevant for any situation where a demand, known in advance, has to be processed by batches and for which we want to minimize the processing time. This kind of situation is often met in chemical industry. An example whose motivation is very close to ours considers a test to be performed on samples, which arrive continuously (Lehoux-Lebacque et al. [45]). The test takes a certain amount of time and can be performed on several samples simultaneously. The question is then to determine the test schedule that minimizes the processing time. Another example is provided by the TCP protocol: a server faces a flow of packets that it has to treat by batches, while minimizing the waiting time of the packets. Actually, only the online version of the problem makes sense in practice, but the offline version, close to our problem, is used for comparison purpose, see Dooly et al. [23]. Barbosa and Friedman [2] have considered a dynamic lot-sizing problem with a motivation similar to ours: a continuous demand is known in advance and replenishments have to be scheduled in order to satify the demand at a minimum cost.

We propose efficient algorithms for the different versions. “Efficient” here means “theoretically efficient”, according to their time complexity and the performance guarantee. It also means “practically efficient” for almost all cases, as shown by numerical experiments conducted on real-world instances. It might also be worth noting that, depending on the version considered, the proof techniques rely on various fields of optimization (convexity, Karush-Kuhn-Tucker conditions, binary search, domination results, shortest paths in finite graphs, ...).

7.2 Model

7.2.1 The problems

We are given a fleet of S shuttles, for which departure times have to be determined. All shuttles have a capacity $C \geq 0$ and are situated in the same loading terminal at the beginning of the day. The users are infinitesimal and arrive continuously in the terminal over a finite period of time, modeled as the interval $[0, T]$, following the cumulative nondecreasing function $D : [0, T] \to \mathbb{R}_+$, where $D(t)$ is the total number of users arrived in the terminal during the interval $[0, t]$. We assume throughout this chapter that $D(T) > 0$. The shuttles have to carry the users over a fixed trip.

The loading process is the one described in Section 2.2.4 for Heavy Gross Vehicles: when the users arrive in the terminal, they enter a queue. This queue closes when all the users who will leave with the next shuttle have arrived in the queue and users can enter a new queue only if the previous one is closed. When a queue is closed, the users in that queue can start boarding the shuttle. The process in illustrated on Figure 2.2. Loading a shuttle with a total of x users takes a time νx. This ν is a parameter, which we call the *loading speed*. Note that setting ν to zero allows to model the case where the users do not have to wait for the last user before boarding. Even if no users arrive strictly after time T, loading and departures are allowed after that instant.

Two possibilities are considered regarding the shuttles. Either return is not allowed: once the shuttles leave, they never come back to the terminal; or it is allowed: once the shuttles leave, they come back to the terminal after a *return time* $\pi \geq 0$. Two objective functions to be minimized are considered: the maximum waiting time and the average waiting time.

We have thus four problems:

- $\mathrm{P}^{\max}_{\text{no return}}$, which consists of not allowing return and minimizing the maximum waiting time.
- $\mathrm{P}^{\text{ave}}_{\text{no return}}$, which consists of not allowing return and minimizing the average waiting time.
- $\mathrm{P}^{\max}_{\text{return}}$, which consists of allowing return and minimizing the maximum waiting time.
- $\mathrm{P}^{\text{ave}}_{\text{return}}$, which consists of allowing return and minimizing the average waiting time.

Practical constraints impose that overtake is not possible and thus, when return is allowed, the departure orders of the shuttles remain the same over the whole period. It is nevertheless possible to have simultaneous trips. This is an approximation in the case of Eurotunnel (we neglect the security distance and the length of the shuttles). For other situations, as the chemical application mentioned in the introduction, it may match what is met in practice.

7.2.2 The demand

Throughout the chapter, we assume that $D(\cdot)$ is upper semicontinuous. It allows to model discontinuity in the arrival process (batch of users arriving simultaneously). Yet, a weaker requirement could lead to mathematical difficulties, e.g., nonexistence of optimal solutions even for very simple cases.

The *pseudo-inverses* of $D(\cdot)$, defined by

$$\begin{array}{rccl} \tau & : & [0, D(T)] & \longrightarrow & [0, T] \\ & & y & \longmapsto & \begin{cases} \inf\{t \in [0,T] : D(t) > y\} & \text{if } y < D(T) \\ T & \text{otherwise,} \end{cases} \end{array}$$

and

$$\begin{array}{rccl} \bar{\tau} & : & [0, D(T)] & \longrightarrow & [0, T] \\ & & y & \longmapsto & \inf\{t \in [0,T] : D(t) \geq y\} \end{array}$$

play an important role in this chapter. Note that they are nondecreasing functions and that $\tau(y) \geq \bar{\tau}(y)$ for all $y \in [0, D(T)]$. Times $\tau(y)$ and $\bar{\tau}(y)$ are respectively interpreted as the arrival times of the first user after the y firsts and of the last user of these y first. Since $D(\cdot)$ is upper semicontinuous, we have the following properties, with proofs for sake of completeness.

Lemma 1. *We have $D(\tau(y)) \geq D(\bar{\tau}(y)) \geq y$ for every $y \in [0, D(T)]$.*

Proof. Since $D(\cdot)$ is nondecreasing, the first inequality is a direct consequence of the inequality $\tau(y) \geq \bar{\tau}(y)$, which is obvious from the definition. To prove the second inequality, consider (t_n) a nonincreasing sequence converging toward $\bar{\tau}(y)$ such that $D(t_n) \geq y$ for all n. By the upper semicontinuity of $D(\cdot)$, we get then $D(\bar{\tau}(y)) \geq y$. □

Lemma 2. *$\tau(\cdot)$ is upper semicontinuous and $\bar{\tau}(\cdot)$ is lower semicontinuous.*

Proof. We first prove that $\tau(\cdot)$ is upper semicontinuous. Let α be some real number such that $\{y\colon \tau(y) < \alpha\}$ is nonempty and take from it an arbitrary element y_0. We want to prove that $\{y\colon \tau(y) < \alpha\}$ is open for the induced topology on $[0, D(T)]$. If $y_0 = D(T)$, then this set is $[0, D(T)]$ and thus open. Otherwise, by the definition of $\tau(\cdot)$, we know that there exists $t_0 < \alpha$ such that $D(t_0) > y_0$. For any element y in $[0, D(t_0))$, we have $\tau(y) \le t_0$, and thus $[0, D(t_0))$ is an open set containing y_0 and fully contained in $\{y\colon \tau(y) < \alpha\}$. The set $\{y\colon \tau(y) < \alpha\}$ is thus an open set of $[0, D(T)]$ for every real number α, which precisely means that $\tau(\cdot)$ is upper semicontinuous.

We prove now that $\bar{\tau}(\cdot)$ is lower semicontinuous. Let α be some nonnegative number. Consider a converging sequence (y_n) in $[0, D(T)]$ such that $\bar{\tau}(y_n) \le \alpha$ for all n. We want to prove that $\bar{\tau}(\lim_{n\to\infty} y_n)$ is at most α. This is obvious when $\alpha = T$. We can thus assume that $\alpha < T$. By definition of $\bar{\tau}(y_n)$, we know that $D(\alpha + \frac{1}{n}) \ge y_n$ (we consider n large enough so that $\alpha + \frac{1}{n} \le T$). Since $D(\cdot)$ is upper semicontinuous and nondecreasing, we get that $D(\alpha) \ge \lim_{n\to\infty} y_n$, and the conclusion follows.

□

Lemma 3. *If $D(\cdot)$ is increasing, then $\tau(y) = \bar{\tau}(y)$ for every $y \in [0, D(T)]$.*

Proof. If $\bar{\tau}(y) = T$, then the equality is obvious. We can thus assume that $\bar{\tau}(y) < T$. For every $t > \bar{\tau}(y)$, we have $D(t) > D(\bar{\tau}(y))$ since $D(\cdot)$ is increasing, and Lemma 1 implies that $D(t) > y$. By definition of $\tau(\cdot)$, we have $\tau(y) \le \bar{\tau}(y)$. The reverse inequality being clear from the definitions, we get the result. □

7.2.3 Mathematical model

For the four problems $\mathrm{P}^{\max}_{\text{no return}}$, $\mathrm{P}^{\text{ave}}_{\text{no return}}$, $\mathrm{P}^{\max}_{\text{return}}$, and $\mathrm{P}^{\text{ave}}_{\text{return}}$, a feasible solution is characterized by two nondecreasing sequences of nonnegative real numbers $\boldsymbol{d} = d_1, d_2, \ldots$ and $\boldsymbol{y} = y_1, y_2, \ldots$. The d_j's are the successive departure times of the shuttles, and the y_j's are their successive cumulative loads: the jth departure occurs at time d_j with a load of $y_j - y_{j-1}$ users, where we set $y_0 = 0$.

Denote by $g^{\max}(\boldsymbol{d}, \boldsymbol{y})$ the value of the maximum waiting time and by $g^{\text{ave}}(\boldsymbol{d}, \boldsymbol{y})$ the value of the average waiting time. There are explicit expressions of these objective functions. Note that $\tau(y_j)$ can be interpreted as the first arrival time of a user leaving with the "$(j+1)$th shuttle".

$$
\begin{aligned}
g^{\max}(\boldsymbol{d}, \boldsymbol{y}) &= \max_{j\colon y_j > y_{j-1}} \big(d_j - \tau(y_{j-1})\big), \\
g^{\text{ave}}(\boldsymbol{d}, \boldsymbol{y}) &= \frac{1}{D(T)} \sum_j \int_{y_{j-1}}^{y_j} \big(d_j - \bar{\tau}(y)\big)\, dy,
\end{aligned}
$$

where the indices j range over all departures.

Problems $P^{\max}_{\text{no return}}$ and $P^{\text{ave}}_{\text{no return}}$ can be written under the following form,

$$
\begin{array}{llrcllr}
\text{Min} & g(\boldsymbol{d},\boldsymbol{y}) & & & & & \\
\text{s.t.} & & y_j - y_{j-1} & \leq & C & j\in\{1,\dots,S\} & \text{(i)}\\
& & y_{j-1} & \leq & y_j & j\in\{1,\dots,S\} & \text{(ii)}\\
& & d_{j-1} & \leq & d_j & j\in\{2,\dots,S\} & \text{(iii)}\\
& & y_S & = & D(T) & & \text{(iv)}\\
& & \bar{\tau}(y_j)+\nu(y_j-y_{j-1}) & \leq & d_j & j\in\{1,\dots,S\} & \text{(v)}\\
& & y_0 & = & 0, & &
\end{array}
\qquad (\mathrm{P}_{\text{no return}})
$$

where $g(\cdot)$ is either $g^{\max}(\cdot)$ or $g^{\text{ave}}(\cdot)$. Constraint (i) ensures that the total number of users in any shuttle does not exceed the shuttle capacity. Constraint (ii) ensures that the indices of the y_j variables are consistent. Constraint (iii) ensures that the shuttles do not overtake. Constraint (iv) ensures that every user eventually leaves the terminal in a shuttle. Constraint (v) ensures that the departure time of a shuttle occurs once the last user of this shuttle has arrived and the loading is over.

Problems $P^{\max}_{\text{no return}}$ and $P^{\text{ave}}_{\text{no return}}$ always admit optimal solutions when they are feasible, i.e., when $CS \geq D(T)$. Indeed, $\bar{\tau}(y_j)+\nu(y_j-y_{j-1})$ is upper-bounded by $T+\nu C$ and adding a constraint $d_j \leq T+\nu C$ for all j does not change the optimal value; since $\bar{\tau}(\cdot)$ is lower semicontinuous (Lemma 2), the set of feasible solutions of the optimization problem obtained with this new constraint is compact; its objective function is lower semicontinuous (and even continuous in the case of $P^{\text{ave}}_{\text{no return}}$).

The following properties for $P^{\max}_{\text{no return}}$ and $P^{\text{ave}}_{\text{no return}}$ will be useful in some proofs.

Lemma 4. *Replacing $g^{\max}(\cdot)$ by $\max_j\left(d_j-\tau(y_{j-1})\right)$ does not change the optimal value of $P^{\max}_{\text{no return}}$.*

Proof. Let $(\boldsymbol{d},\boldsymbol{y})$ be a feasible solution of $P^{\max}_{\text{no return}}$. We are going to build a feasible solution $(\boldsymbol{d}',\boldsymbol{y})$ (with the same $\boldsymbol{y}$) such that

$$g^{\max}(\boldsymbol{d},\boldsymbol{y}) \geq g^{\max}(\boldsymbol{d}',\boldsymbol{y}) = \max_j\left(d'_j-\tau(y_{j-1})\right). \tag{7.1}$$

We set $d'_1=\bar{\tau}(y_1)+\nu y_1$ and define inductively $d'_j=\left(d'_{j-1},\bar{\tau}(y_j)+\nu(y_j-y_{j-1})\right)$. We have $d'_j\leq d_j$ for all j and it implies the inequality in (7.1). Let us prove the equality of (7.1): if $\max_j\left(d'_j-\tau(y_{j-1})\right)$ is obtained for a $\bar{j}$ such that $y_{\bar{j}-1}<D(T)$, then there exists $k\geq\bar{j}$ such that $y_k>y_{k-1}=y_{\bar{j}-1}$ and $d'_k\geq d'_{\bar{j}}$, which means that the maximum is also obtained for a k such that $y_k>y_{k-1}$; and if $\max_j\left(d'_j-\tau(y_{j-1})\right)$ is attained for a $\bar{j}$ such that $y_{\bar{j}-1}=D(T)$, then there exists $\ell\leq\bar{j}-1$ such that $y_{\ell-1}<y_\ell=y_{\bar{j}-1}$ and by construction $d'_\ell=d'_{\ell+1}=\cdots=d'_S$ (since $y_\ell=y_{\ell+1}=\cdots=y_S=D(T)$), which means that the maximum is also obtained for an ℓ such that $y_\ell>y_{\ell-1}$. □

Lemma 5. *If $D(\cdot)$ is increasing, for any of $\mathrm{P}^{\max}_{\text{no return}}$ and $\mathrm{P}^{\text{ave}}_{\text{no return}}$, there is an optimal solution such that $d_j = \bar{\tau}(y_j) + \nu(y_j - y_{j-1})$ for all $j \in \{1,\ldots,S\}$.*

Proof. Let $(\boldsymbol{d},\boldsymbol{y})$ be an optimal solution (we do not care which objective function is used yet). We choose this optimal solution with minimal $\sum_{j=1}^{S} y_j$ among all possible optimal solutions. Such a solution exists by continuity and compactness: as explained right before Lemma 4, we can add the constraint $d_j \le T + \nu C$ for all j, without changing the optimal value and without changing the possible values for $\boldsymbol{y}$ at optimality. Without loss of generality, we can moreover assume that $d_1 = \bar{\tau}(y_1) + \nu y_1$ and that for all $j \in \{2,\ldots,S\}$ we have

$$d_j = \max\big(d_{j-1}, \bar{\tau}(y_j) + \nu(y_j - y_{j-1})\big) \tag{7.2}$$

(just redefine d_j according to these equalities if necessary). When $\nu = 0$, a straightforward induction on j shows that we have then always $d_j = \bar{\tau}(y_j)$. We can thus assume that $\nu > 0$.

Suppose for a contradiction that there is a j such that $d_j > \bar{\tau}(y_j) + \nu(y_j - y_{j-1})$. Denote by j_1 the smallest index for which this inequality holds. We necessarily have $d_{j_1} = d_{j_1-1}$ (because of the equality (7.2)). Denote by j_0 the smallest index $j < j_1$ such that $d_j = d_{j_1}$. Note that since $D(\cdot)$ is increasing, we have that $\bar{\tau}(\cdot)$ is continuous (it is upper and lower semicontinuous with Lemma 3).

For some small $\varepsilon > 0$, we define $(\bar{\boldsymbol{d}},\bar{\boldsymbol{y}})$ as follows:

$$\bar{y}_j = \begin{cases} y_j - \varepsilon & \text{for } j \in \{j_0,\ldots,j_1-1\} \\ y_j & \text{otherwise} \end{cases}$$

and

$$\bar{d}_j = \begin{cases} \max\big(\bar{d}_{j-1}, \bar{\tau}(\bar{y}_j) + \nu(\bar{y}_j - \bar{y}_{j-1})\big) & \text{for } j \in \{j_0,\ldots,j_1\} \\ d_j & \text{otherwise,} \end{cases}$$

where $\bar{d}_0 = 0$. We first check that $(\bar{\boldsymbol{d}},\bar{\boldsymbol{y}})$ is a feasible solution of $(\mathrm{P}_{\text{no return}})$.

The definition of j_1 implies that $d_{j_0} > 0$. Thus if $j_0 = 1$, we have $y_1 > 0$ and for a small enough ε, the vector $\bar{\boldsymbol{y}}$ satisfies constraint (ii). Otherwise, we have $\bar{\tau}(y_{j_0-1}) + \nu(y_{j_0-1} - y_{j_0-2}) = d_{j_0-1} < d_{j_0} = \bar{\tau}(y_{j_0}) + \nu(y_{j_0} - y_{j_0-1})$. It implies that $y_{j_0-1} < y_{j_0}$ (as otherwise the equality would imply that $y_{j_0-1} < y_{j_0-2}$). Thus, for a small enough ε, we have $\bar{\boldsymbol{y}}$ satisfies constraint (ii). It also satisfies obviously constraint (iv).

For $j \in \{2,\ldots,j_1\} \cup \{j_1+2,\ldots,S\}$, checking $\bar{d}_{j-1} \le \bar{d}_j$ is straightforward. The remaining case is $j = j_1 + 1$. A direct induction shows that $\bar{d}_j \le d_j$ for $j \le j_1 - 1$. Since $\bar{\tau}(y_{j_1}) + \nu(y_{j_1} - y_{j_1-1}) < \bar{\tau}(y_{j_1-1}) + \nu(y_{j_1-1} - y_{j_1-2})$ (because $d_{j_1-1} = d_{j_1}$), for ε small enough, we have $\bar{d}_{j_1-1} \ge \bar{\tau}(\bar{y}_{j_1}) + \nu(\bar{y}_{j_1} - \bar{y}_{j_1-1})$. Here, we use the fact that $\bar{\tau}(\cdot)$ is continuous. Thus $\bar{d}_{j_1} = \bar{d}_{j_1-1}$. Since we have $\bar{d}_{j_1-1} \le d_{j_1-1}$ by the above induction, we finally obtain $\bar{d}_{j_1} \le d_{j_1} \le d_{j_1+1} = \bar{d}_{j_1+1}$. Therefore, $\bar{\boldsymbol{d}}$ satisfies constraint (iii).

Constraint (i) is satisfied for all j, except maybe for $j = j_1$. We have proved that $\bar{d}_{j_1} = \bar{d}_{j_1-1}$. Since $\bar{d}_{j_1-1} = \bar{\tau}(\bar{y}_{j'}) + \nu(\bar{y}_{j'} - \bar{y}_{j'-1})$ for some $j' \le j_1 - 1$, we have $\bar{\tau}(\bar{y}_{j_1}) + \nu(\bar{y}_{j_1} - \bar{y}_{j_1-1}) \le \bar{d}_{j_1} =$

$\bar{\tau}(\bar{y}_{j'}) + \nu(\bar{y}_{j'} - \bar{y}_{j'-1})$, and thus $\nu(\bar{y}_{j_1} - \bar{y}_{j_1-1}) \leq \nu(\bar{y}_{j'} - \bar{y}_{j'-1}) \leq \nu C$. Therefore constraint (i) is also satisfied for $j = j_1$.

Since the constraint (v) is clearly satisfied, $(\bar{\boldsymbol{d}}, \bar{\boldsymbol{y}})$ is a feasible solution of $(\mathrm{P}_{\text{no return}})$.

We have proved that $\bar{d}_{j_1} \leq d_{j_1}$. Therefore,

$$\begin{aligned}\sum_{j=j_0}^{j_1} \int_{\bar{y}_{j-1}}^{\bar{y}_j} \big(\bar{d}_j - \bar{\tau}(u)\big)du &\leq \int_{\bar{y}_{j_0-1}}^{\bar{y}_{j_1}} \big(\bar{d}_{j_1} - \bar{\tau}(u)\big)du \\ &< \int_{y_{j_0-1}}^{y_{j_1}} \big(d_{j_1} - \bar{\tau}(u)\big)du \\ &= \sum_{j=j_0}^{j_1} \int_{y_{j-1}}^{y_j} \big(d_j - \bar{\tau}(u)\big)du,\end{aligned}$$

which shows that $(\bar{\boldsymbol{d}}, \bar{\boldsymbol{y}})$ is also an optimal solution of $\mathrm{P}^{\text{ave}}_{\text{no return}}$, which is in contradiction with the minimality assumption on $\sum_{j=1}^{S} y_j$. The case of $\mathrm{P}^{\max}_{\text{no return}}$ is dealt with similarly. $\square$

Problems $\mathrm{P}^{\max}_{\text{return}}$ and $\mathrm{P}^{\text{ave}}_{\text{return}}$ can be written almost identically under the following form. We use infinitely many variables since there is no *a priori* reason to have a bounded number of departures, and there are indeed special cases for which there is no optimal solution with a finite number of departures. However, if $\pi > 0$, we prove that any optimal solution of $\mathrm{P}^{\max}_{\text{return}}$ requires a finite number of departures, see Proposition 5. The case of $\mathrm{P}^{\text{ave}}_{\text{return}}$ remains open.

$$\begin{array}{lrcllr} \text{Min} & g(\boldsymbol{d}, \boldsymbol{y}) & & & & \\ \text{s.t.} & y_j - y_{j-1} & \leq & C & j \in \{1, \ldots, +\infty\} & \text{(i)} \\ & y_{j-1} & \leq & y_j & j \in \{1, \ldots, +\infty\} & \text{(ii)} \\ & d_{j-1} & \leq & d_j & j \in \{2, \ldots, S\} & \text{(iii)} \\ & \lim\limits_{j \to +\infty} y_j & = & D(T) & & \text{(iv)} \\ & \bar{\tau}(y_j) + \nu(y_j - y_{j-1}) & \leq & d_j & j \in \{1, \ldots, +\infty\} & \text{(v)} \\ & d_j + \pi + \nu(y_{j+S} - y_{j+S-1}) & \leq & d_{j+S} & j \in \{1, \ldots, +\infty\} & \text{(vi)} \\ & y_0 & = & 0, & & \end{array} \qquad (\mathrm{P}_{\text{return}})$$

where $g(\cdot)$ is either $g^{\max}(\cdot)$ or $g^{\text{ave}}(\cdot)$. Constraints (i), (ii), (iii), (iv), and (v) have the same meaning as for the previous problems. Constraint (vi) ensures that the time between two consecutive departures of a same shuttle is not smaller than the time required for a full trip plus the time needed to load the users.

In the model $(\mathrm{P}_{\text{return}})$, the shuttles are not identified. Note, however, that their schedules can be easily be recovered: the departure times of a shuttle s is of the form

$$d_s, d_{s+S}, d_{s+2S}, \ldots$$

and the time at which the loading starts for a shuttle with departure time d_j can be chosen to be $d_j - v(y_j - y_{j-1})$ (the loading starts as late as possible).

While it can be shown that problem $\mathrm{P}^{\max}_{\text{return}}$ always admits an optimal solution when it is feasible (see Proposition 5), we were not able to settle the case of problem $\mathrm{P}^{\text{ave}}_{\text{return}}$.

7.2.4 Computational model

We assume that the following operations take constant time:

- Evaluation of $D(t)$ for any $t \in [0, T]$.
- Integration of $D(\cdot)$ between two values.
- Evaluation of $\tau(y)$ and $\bar{\tau}(y)$ for any $y \in \mathbb{R}_+$.
- Evaluation of $\sup\{y\colon \bar{\tau}(y) + vy \leq \alpha\}$ for any $\alpha \in \mathbb{R}_+$.

Note that if $D(\cdot)$ is piecewise affine with a natural description, as it is usually the case in practice, these assumptions are easily matched. Moreover, we set as constants of the computational model the capacity C, the length of the period T, the cumulative demand $D(\cdot)$, the loading speed v, and the return time π. The complexity functions will be expressed in terms of S and the accuracy of the computed solution.

7.3 Main results

In the present section, we present our main findings. Many results state the existence of algorithms with a guarantee that the returned solution has a value close to the optimal value OPT of the considered problem. Except for two easy results – Corollary 1 and Proposition 5 – all proofs are postponed to other sections.

We organize the results presented in that section in three subsections. The first subsection – Section 7.3.1 – deals with the special case where $D(\cdot)$ is a constant function, i.e., when all users are in the loading terminal from the beginning of the period. It seems to us that these results are also interesting for themselves: there are very efficient algorithms and the situation where the demand is already present makes sense, whether it is for our original motivation, for other logistic applications, or for the applications in chemical industry mentioned in the introduction.

The second subsection – Section 7.3.2 – deals with the general case where the shuttles are not allowed to come back, i.e., with the case covered by the problems $\mathrm{P}^{\max}_{\text{no return}}$ and $\mathrm{P}^{\text{ave}}_{\text{no return}}$. The case where the shuttles are allowed to come back, i.e., when we deal with the problems $\mathrm{P}^{\max}_{\text{return}}$ and $\mathrm{P}^{\text{ave}}_{\text{return}}$, is discussed in Section 7.3.3.

7.3.1 All users in the terminal from the beginning

In this subsection, we present results regarding the four problems when $D(t) = D(T)$ for all $t \in [0, T]$ (all users are from the beginning in the terminal). For the problems for which return is not allowed ($\mathrm{P}^{\max}_{\text{no return}}$ and $\mathrm{P}^{\text{ave}}_{\text{no return}}$), an obvious optimal solution is given by

$y_j^* = jD(T)/S$ and $d_j^* = \nu D(T)/S$ for $j \in \{1,\ldots,S\}$ and the optimal value is $\nu D(T)/S$ for both problems, provided that $D(T) \leq CS$ (otherwise, there is no feasible solution at all): the shuttles take all the same number of users, start immediately the loading process, and have the same departure time.

The rest of the section is devoted to the results regarding the problems $\mathrm{P}^{\max}_{\text{return}}$ and $\mathrm{P}^{\text{ave}}_{\text{return}}$. For the first one, there are closed-from expressions for the optimal value.

Proposition 3. *When $D(t) = D(T)$ for all $t \in [0,T]$, the optimal value of $\mathrm{P}^{\max}_{\text{return}}$ is*

$$\frac{\nu D(T)}{S} + \left(\left\lceil \frac{D(T)}{CS} \right\rceil - 1\right)\pi.$$

In the proof, we actually provide a closed-form expression for an optimal solution. For $\mathrm{P}^{\text{ave}}_{\text{return}}$ however, there does not seem to be a closed-form expression for an optimal solution, and not even for the optimal value. There is nevertheless an efficient algorithm.

Proposition 4. *Suppose $\pi > 0$. When $D(t) = D(T)$ for all $t \in [0,T]$, the optimal value of $\mathrm{P}^{\text{ave}}_{\text{return}}$ can be computed in constant time and an optimal solution can be computed in $O(S)$.*

If $\pi = 0$, it can be shown that the optimal value is $\frac{\nu D(T)}{2S}$, and it is not too difficult to see that there is no optimal solution. In a transportation context, $\pi = 0$ looks unrealistic. However, it corresponds to the problem where people must leave a building by an airlock (then, νx is the time needed by x people to enter and leave an airlock; S is the total number of airlocks, people being able to leave the building by any of them.)

7.3.2 When return is not allowed

We have the existence of an efficient approximation algorithm for $\mathrm{P}^{\max}_{\text{no return}}$. The algorithm is actually an easy binary search (Section 7.5.1).

Theorem 3. *Let $\rho > 0$. A feasible solution $(\boldsymbol{d},\boldsymbol{y})$ of $\mathrm{P}^{\max}_{\text{no return}}$ – if the problem is feasible – satisfying $g^{\max}(\boldsymbol{d},\boldsymbol{y}) \leq OPT + \rho$ can be computed in $O\left(S \log \frac{1}{\rho}\right)$.*

With an additional assumption on $D(\cdot)$, this theorem provides actually an approximation scheme.

Corollary 1. *If $D(\cdot)$ is increasing, the algorithm of Theorem 3 computes in $O\left(S \log \frac{S}{\varepsilon}\right)$ a $(1+\varepsilon)$-approximation for $\mathrm{P}^{\max}_{\text{no return}}$.*

A schedule for the shuttles requires to specify S real numbers. Taking an output sensitive point of view, this corollary states thus the existence of a polynomial approximation scheme in this particular case.

Proof of Corollary 1. Suppose $D(\cdot)$ increasing. Let $(\boldsymbol{d},\boldsymbol{y})$ be a feasible solution. According to Lemma 3, we have then $\tau(y_{j-1}) = \bar{\tau}(y_{j-1})$ for every j and the maximum waiting time for shuttle j is at least $\tau(y_j) + \nu(y_j - y_{j-1}) - \tau(y_{j-1})$. Note that if $y_j = y_{j-1}$, this quantity is zero. Hence, the sum of the maximum waiting times over all nonempty shuttles is at least $T + \nu D(T)$ and the optimal value OPT of $\mathrm{P}^{\max}_{\text{no return}}$ is at least $(T + \nu D(T))/S$. Setting ρ to $\varepsilon(T + \nu D(T))/S$ in Theorem 3 leads to the result. □

For $\mathrm{P}^{\text{ave}}_{\text{no return}}$, there exists an efficient approximation algorithm too. The algorithm is also described later (Section 7.5.2). We already outline that this algorithm is not a binary search as in the former case, but consists in building a quite simple weighted "approximative" graph, in which a shortest path is computed.

Theorem 4. *Suppose that $D(\cdot)$ admits right derivatives everywhere (denoted $D'_+(t)$) and $\inf_{t\in[0,T)} D'_+(t)$ is positive. Then, for any positive integer M, a feasible solution $(\boldsymbol{d}, \boldsymbol{y})$ of $\mathrm{P}^{\text{ave}}_{\text{no return}}$ – if the problem is feasible – satisfying*

$$g^{\text{ave}}(\boldsymbol{d}, \boldsymbol{y}) \leq OPT + O\left(\frac{S^2}{M}\right)$$

can be computed in $O(SM^3)$.

As for Corollary 1 above, this theorem could be interpreted as a polynomial approximation scheme by using the fact that $D(\cdot)$ is increasing.

7.3.3 When return is allowed

The following proposition implies that when π is larger than 0, any optimal solution of $\mathrm{P}^{\max}_{\text{return}}$ requires a finite number of nonempty departures.

Proposition 5. *If $\pi > 0$, there exists an optimal solution of $\mathrm{P}^{\max}_{\text{return}}$ and the number of nonempty departures in any optimal solution is at most*

$$\left(\left\lceil \frac{T}{\pi} \right\rceil + 1\right) S + \left(\frac{\nu}{\pi} + \frac{1}{C}\right) D(T).$$

Proof. The schedule consisting in making the shuttles wait until time T, loading them at full capacity (except maybe for the last departure), and making them leave as soon as the loading is completed provides a feasible solution of $\mathrm{P}^{\max}_{\text{return}}$ with a value $T + \nu D(T)/S + (\lceil D(T)/(CS) \rceil - 1)\pi$.

Consider a feasible solution of $\mathrm{P}^{\max}_{\text{return}}$ with a number k (possibly infinite) of departures after time T. The users in the last shuttle to leave have waited at least $T + (k/S - 1)\pi$. To get a solution as good as the one described above, we must have

$$T + \left(\frac{k}{S} - 1\right)\pi \leq T + \frac{\nu D(T)}{S} + \frac{\pi D(T)}{CS},$$

which is equivalent to

$$k \leq \frac{\nu D(T)}{\pi} + \frac{D(T)}{C} + S.$$

Before time T, the number of departures is at most $\lceil T/\pi \rceil S$. Hence, we do not worsen the optimal value by adding the constraint that the total number of departures is bounded by the right-hand term of the inequality plus $\lceil T/\pi \rceil S$. The set of feasible solutions can thus be reduced to the solutions where the number of nonempty departures is bounded by such a quantity. Since $\bar{\tau}(\cdot)$ is lower semicontinuous (Lemma 2), it makes the set of feasible solutions compact. The objective function being lower semicontinuous, there exists an optimal solution. The bound on the number of departures has been proved along the lines. □

The next theorem states that there exists an algorithm computing arbitrarily good feasible solutions for $\mathrm{P}^{\max}_{\text{return}}$ within reasonable computational times when S is small. As for Section 7.3.2, this algorithm is described later in the chapter (Section 7.6). It is based on the computation of a shortest path in an "approximative" graph, as for Theorem 4. It also uses Proposition 5 in a crucial way (actually a slight variation of it: Lemma 15).

Theorem 5. *Suppose that $D(\cdot)$ admits right derivatives everywhere, π is positive, and $\inf_{t\in[0,T)} D'_+(t)$ is positive. Then, for any positive integer M, a feasible solution $(\boldsymbol{d},\boldsymbol{y})$ of $\mathrm{P}^{\max}_{\text{return}}$ satisfying*

$$g^{\max}(\boldsymbol{d},\boldsymbol{y}) \leq OPT + O\left(\frac{S^2}{M}\right)$$

can be computed in $O\left(\beta^{3S}M^{3S+2}\right)$, where β depends only on the constants of the computational model.

As above, the theorem actually ensures that the algorithm is an approximation scheme since we can bound from below OPT using only the input values. If S is considered as constant, this becomes even a polynomial approximation scheme.

We do not know whether there is a counterpart to Proposition 5 for problem $\mathrm{P}^{\text{ave}}_{\text{return}}$. If such a counterpart existed, then almost the same technique as the one used in Section 7.6 would lead to a theorem similar to Theorem 5 for $\mathrm{P}^{\text{ave}}_{\text{return}}$. The existence of such a theorem remains thus open.

7.4 All users in the terminal from the beginning

In this section, we prove Propositions 3 and 4, which are about the case where all users are in the loading terminal from the beginning and where the shuttles are allowed to come back. The case where the shuttles are not allowed to come back has been dealt with in Section 7.3.1. To ease the reading, and for the present section only, we use D to denote the quantity $D(T)$.

We treat first the case of problem $\mathrm{P}^{\max}_{\text{return}}$.

Proof of Proposition 3. For $\mathrm{P}^{\max}_{\text{return}}$, when $S=1$, an optimal solution is obtained by loading at full capacity the shuttle for each departure (except maybe for the last departure for which the shuttle load is $D - C\lfloor D/C \rfloor$) and by making the shuttle leave immediately after each loading process. The optimal value is then $\nu D + (\lceil D/C \rceil - 1)\pi$. When $S > 1$, consider the problem Q defined as the problem $\mathrm{P}^{\max}_{\text{return}}$ without the constraint that the shuttles do not overtake (constraint (iii) in ($\mathrm{P}_{\text{return}}$)). The optimal value of Q provides a lower bound of the optimal value of $\mathrm{P}^{\max}_{\text{return}}$. Since there is no constraint linking the different shuttles, problem Q can be solved separately for each shuttle s with a demand D_s to carry, such that $\sum_s D_s = D$. The optimal solutions of Q are thus obtained from the optimal solutions of

$$\begin{array}{lrcll} \text{Min} & \displaystyle\max_{s\in\{1,\ldots,S\}} & \multicolumn{3}{l}{\left(\nu D_s + \left(\left\lceil \dfrac{D_s}{C} \right\rceil - 1\right)\pi\right)} \\ \text{s.t.} & \displaystyle\sum_{s=1}^{S} D_s & = & D & \\ & D_s & \geq & 0 & s \in \{1,\ldots,S\}. \end{array}$$

The solution given by $D_s = D/S$ for all s is clearly optimal (and it is actually the unique optimal solution when $\nu > 0$). Hence, there is an optimal solution for Q in which all shuttles have the same departure times and, for each trip, carry the same number of users. Its value is $\nu D/S + (\lceil D/(CS) \rceil - 1)\pi$. Since the shuttles do not overtake in this optimal solution of Q, it is actually a feasible solution for the problem $\mathrm{P}^{\max}_{\text{return}}$, and thus an optimal solution for this latter problem (its value being equal to a lower bound). $\square$

The rest of this section is devoted to the proof of Proposition 4, which ensures the existence of an efficient algorithm solving problem $\mathrm{P}^{\text{ave}}_{\text{return}}$ when $D(t) = D$ for all $t \in [0, T]$. We start by considering the special case of problem $\mathrm{P}^{\text{ave}}_{\text{return}}$ when $S = 1$. In such a case, it is always beneficial to define $d_j = (j-1)\pi + \nu y_j$. Assuming that the y_j's are given, it provides a feasible solution since $\bar{\tau}(y) = 0$ for all $y \in [0, D]$. The objective function of $\mathrm{P}^{\text{ave}}_{\text{return}}$ becomes thus

$$\frac{1}{D}\sum_{j=1}^{+\infty}\left((j-1)\pi + \nu\sum_{i=1}^{j} x_i\right)x_j$$

where $x_j = y_j - y_{j-1}$. It can be alternatively written as

$$\frac{1}{D}\left(\sum_{j=1}^{+\infty}(j-1)\pi x_j + \frac{1}{2}\nu\sum_{j=1}^{+\infty} x_j^2\right) + \frac{\nu D}{2}.$$

Solving $\mathrm{P}^{\text{ave}}_{\text{return}}$ when $S = 1$ reduces thus to solving

$$\begin{array}{ll} \text{Min} & \displaystyle\sum_{j=1}^{+\infty}(j-1)\pi x_j + \frac{1}{2}\nu\sum_{j=1}^{+\infty} x_j^2 \\ \text{s.t.} & \displaystyle\sum_{j=1}^{+\infty} x_j = D \\ & 0 \leq x_j \leq C \qquad j \in \{1,\ldots,+\infty\}. \end{array} \tag{$P(D)$}$$

which is a convex program (with infinitely many variables). We will show that there is always an optimal solution of $(P(D))$ with a finite support. Then, we will solve $(P_0(D))$, defined as the program $(P(D))$ with the additional constraint $|\{j\colon x_j \neq 0\}| < +\infty$, with the help of the Karush-Kuhn-Tucker conditions (that do not apply otherwise).

Lemma 6. *Suppose that $\pi > 0$. Then $(P_0(D))$ has an optimal solution and it is necessarily of the form*

$$x_0^* = 0$$
$$x_j^* = \begin{cases} C & \text{if } j \leq a, \\ \dfrac{D-aC}{\theta(a)-a} + \dfrac{\pi}{\nu}\left(\dfrac{a+\theta(a)+1}{2} - j\right) & \text{if } a+1 \leq j \leq \theta(a), \\ 0 & \text{otherwise,} \end{cases}$$

with $a \in \mathbb{Z}_+$ such that $a \leq \frac{D}{C}$ and where

$$\theta(a) = a + \left\lceil \frac{-1+\sqrt{1+\frac{8\nu}{\pi}(D-aC)}}{2} \right\rceil.$$

Proof. Consider the following program

$$\begin{aligned} \text{Min} \quad & \sum_{j=1}^{n}(j-1)\pi x_j + \frac{1}{2}\nu\sum_{j=1}^{n} x_j^2 \\ \text{s.t.} \quad & \sum_{j=1}^{n} x_j = D \\ & 0 \leq x_j \leq C \qquad j \in \{1,\ldots,n\}. \end{aligned} \tag{$P_0^n(D)$}$$

Note that $(P_0^n(D))$ is actually $(P_0(D))$ with the additional constraint that $\sup\{j\colon x_j \neq 0\} \leq n$.

For $n < D/C$, $(P_0^n(D))$ has no feasible solutions, and for $n \geq D/C$, the set of feasible solutions is nonempty. In this case, by compactness and continuity of the objective function, $(P_0^n(D))$ has an optimal solution $\boldsymbol{x}^*$. We necessarily have $x_j^* \geq x_{j+1}^*$ for every $j \in \{1,\ldots,n-1\}$, otherwise, exchanging the two values would strictly decrease the objective function. Let a be the largest index j such that $x_j^* = C$, with the convention that $a = 0$ if there are no such index j, and let $b+1$ be the smallest index j such that $x_j^* = 0$, with the convention that $b = n$ if there is no such index j.

The constraints being all affine, the Karush-Kuhn-Tucker conditions apply. There is thus a real number $\lambda \in \mathbb{R}$ and two collections $\boldsymbol{\mu}, \boldsymbol{\omega} \in \mathbb{R}_+^n$ such that for every $j \in \{1,\ldots,n\}$ we have simultaneously

$$\begin{cases} \nu x_j^* + (j-1)\pi + \lambda + \mu_j - \omega_j = 0 \\ \omega_j x_j^* = \mu_j(x_j^* - C) = 0. \end{cases} \tag{7.3}$$

Summing the first equality from $j = a+1$ to $j = b$ and noting that $\mu_j = \omega_j = 0$ and $\sum_{j=a+1}^{b} x_j^* = D - aC$ by definition of a and b provides an expression of λ in terms of a and

b. Replacing λ by this expression in the same equality leads to

$$x_j^* = \begin{cases} C & \text{if } j \le a, \\ \dfrac{D-aC}{b-a} + \dfrac{\pi}{\nu}\left(\dfrac{a+b+1}{2} - j\right) & \text{if } a+1 \le j \le b, \\ 0 & \text{otherwise.} \end{cases}$$

Using this equality for $j = b$ gives the following equation.

$$(b-a)(b-a-1) < \frac{2\nu}{\pi}(D-aC).$$

Equation (7.3) specialized for $j = b+1$ gives

$$(b-a)(b-a+1) \ge \frac{2\nu}{\pi}(D-aC).$$

These two inequalities together – treated as conditions on a second order polynomial in $b-a$ – imply the necessary condition

$$-\frac{1}{2} + \frac{\sqrt{1+\frac{8\nu}{\pi}(D-aC)}}{2} \le b-a < \frac{1}{2} + \frac{\sqrt{1+\frac{8\nu}{\pi}(D-aC)}}{2}$$

which imposes a unique integer value for $b-a$ and b takes a unique value $\theta(a)$ for each value of a. We have proved that any optimal solution of $(P_0^n(D))$ is of this form. Now, note that by definition of a, we necessarily have $a \le \lfloor D/C \rfloor$. It means that there are only finitely many optimal solutions of the $(P_0^n(D))$'s when n goes to infinity. Since the set of feasible solutions of the $(P_0^n(D))$'s is nondecreasing when n goes to infinity, it means actually that there exists an n_0 such that any optimal solution of $(P_0^n(D))$ for $n \ge n_0$ is an optimal solution of $(P_0^{n_0}(D))$. Moreover, any feasible solution of $(P_0(D))$ is a feasible solution of $(P_0^n(D))$ for some $n \ge n_0$, and thus is dominated by the optimal solutions of $(P_0^{n_0}(D))$. These latter are thus the optimal solutions of $(P_0(D))$. □

Let $v(D)$ and $v_0(D)$ be the optimal values of respectively $(P(D))$ and $(P_0(D))$. Note that $v(D) \le v_0(D)$.

Lemma 7. *If $\pi > 0$, we have $v_0(D-\varepsilon) \le v(D)$ for every $\varepsilon \in (0, D]$.*

Proof. Let $\varepsilon \in (0, D]$. Consider a feasible solution $\boldsymbol{x}$ of $(P(D))$. Let $N_\varepsilon \in \mathbb{Z}_+$ be such that $\sum_{j=N_\varepsilon+1}^{+\infty} x_j < \varepsilon$. Define inductively

$$x'_j = \begin{cases} \min\left(x_j, D-\varepsilon - \sum_{i=1}^{j-1} x'_i\right) & \text{for } j \le N_\varepsilon \\ 0 & \text{for } j \ge N_\varepsilon + 1. \end{cases}$$

This $\boldsymbol{x}'$ is a feasible solution of $(P_0(D-\varepsilon))$. Since $x'_j \le x_j$ for all j, the value given by $\boldsymbol{x}'$ to the objective value of $(P_0(D-\varepsilon))$ is nonlarger that the value obtained by $\boldsymbol{x}$ for $(P(D))$. The inequality follows. $\square$

Proof of Proposition 4. Let us deal with the case $S=1$. Using the fact that $(P_0(D))$ is a convex program, we easily get that $v_0(\cdot)$ is a convex function. It is thus continuous on $(0,+\infty)$, and since $v_0(0)=0$, we have that $v_0(\cdot)$ is continuous everywhere on $[0,+\infty)$. Making ε tend toward 0 in Lemma 7 and the inequality $v(D)\leq v_0(D)$ imply that $v_0(D)=v(D)$. Since any feasible solution of $(P_0(D))$ is a feasible solution of $(P(D))$ with the same value for the objective function, every optimal solution of $(P_0(D))$ is an optimal solution of $(P(D))$. An algorithm computing an optimal solution of $(P(D))$ can then be derived from Lemma 6: we just have to try all the finitely many possible values for a. The proof for any value of S will be obtained by showing that an optimal solution in this case consists just in replicating optimal solutions for the one-shuttle case.

When $S>1$, consider the problem Q defined as the problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{return}}$ without the constraint that the shuttles do not overtake (constraint (iii) in $(\mathrm{P}_{\mathrm{return}})$). The optimal value of Q provides a lower bound of the optimal value of $\mathrm{P}^{\mathrm{ave}}_{\mathrm{return}}$. Since there is no constraint linking the different shuttles, problem Q can be solved separately for each shuttle s with a demand D_s to carry, such that $\sum_s D_s = D$. The optimal solutions of Q are thus obtained from the optimal solutions of

$$\begin{array}{lrcll} \text{Min} & \displaystyle\sum_{s=1}^{S}\left(v(D_s)+\frac{\nu}{2}D_s^2\right) & & & \\ \text{s.t.} & \displaystyle\sum_{s=1}^{S} D_s & = & D & \\ & D_s & \geq & 0 & s\in\{1,\ldots,S\}. \end{array}$$

The fact that $(P(D))$ is a convex program implies that the map $v(\cdot)$ is convex. As a consequence, the solution $D_s = D/S$ for all s is an optimal solution of the previous program. Hence, there is an optimal solution for Q in which all shuttles have the same departure times and, for each travel, carry the same amount of users. Since the shuttles do not overtake in this optimal solution of Q, it is actually a feasible solution for the problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{return}}$, and thus an optimal solution for this latter problem (its value being equal to a lower bound). $\square$

7.5 When return is not allowed

7.5.1 Minimizing the maximum waiting time

The algorithm

If $CS < D(T)$, there is no feasible solution. We can thus assume that $CS \geq D(T)$. The algorithm is a binary search starting with the values $h^+ = T+\nu D(T)$ and $h^- = 0$ which are respectively upper and lower bounds of the optimal value. While the gap $h^+ - h^-$ is larger than ρ, we consider the tentative value $h = \frac{h^+ + h^-}{2}$ and the system

$$\begin{cases} y_j = \sup \mathscr{S}_j^h & j \in \{1, \dots, S\} \\ y_S = D(T) & \\ y_0 = 0 & \\ d_j = h + \tau(y_{j-1}) & j \in \{1, \dots, S\}, \end{cases} \tag{S_h}$$

where $\mathscr{S}_j^h = \left\{ y \in \mathbb{R}_+ : y \le C + y_{j-1}, \bar{\tau}(y) + \nu(y - y_{j-1}) - \tau(y_{j-1}) \le h, y \le D(T) \right\}$. Each iteration of the binary search consists in deciding whether (S_h) has a feasible solution or not, and it can be done in $O(S)$ by computing the values of the y_j's and the d_j's iteratively (here we use in particular the computational assumptions on $D(\cdot)$). As we are going to prove, (S_h) has a feasible solution if and only if the problem has a feasible solution with a value of the objective function at most h. If (S_h) has a feasible solution, we update thus the value of h^+ with the current value of h, otherwise, we update h^- with h. When $h^+ - h^- \le \rho$, the solution of program (S_{h^+}) is feasible for $\mathrm{P}^{\max}_{\text{no return}}$ and its value h^+ is at most at ρ from the optimal value.

Proof of Theorem 3

For any fixed h, $\mathrm{P}^{\max}_{\text{no return}}$ has a feasible solution with a value of the objective function at most h if and only if the following system has a feasible solution.

$$\begin{cases} d_j - \tau(y_{j-1}) \le h & j \in \{1, \dots, S\} & \text{(Qi)} \\ y_j - y_{j-1} \le C & j \in \{1, \dots, S\} & \text{(Qii)} \\ y_{j-1} \le y_j & j \in \{1, \dots, S\} & \text{(Qiii)} \\ d_{j-1} \le d_j & j \in \{2, \dots, S\} & \text{(Qiv)} \\ y_S = D(T) & & \text{(Qv)} \\ \bar{\tau}(y_j) + \nu(y_j - y_{j-1}) \le d_j & j \in \{1, \dots, S\} & \text{(Qvi)} \\ y_0 = 0. & & \end{cases} \tag{Q_h}$$

We claim that (Q_h) has a feasible solution if and only if (S_h) has one. Once this equivalence is established, the correctness of the binary search described above is almost immediate using Lemma 4.

Let $(\boldsymbol{d}, \boldsymbol{y})$ be a feasible solution of (S_h). We use without further mention that $\mathscr{S}_j^h$ is closed. It satisfies the constraints (Qi), (Qii), and (Qv). We have $y_{j-1} \le C + y_{j-1}$ and $y_{j-1} \le D(T)$. Since $\bar{\tau}(y) \le \tau(y)$ for all y, we also have $\bar{\tau}(y_{j-1}) + \nu(y_{j-1} - y_{j-1}) - \tau(y_{j-1}) \le h$. It means that y_{j-1} belongs to $\mathscr{S}_j^h$, and thus $y_{j-1} \le y_j$. Hence, $(\boldsymbol{d}, \boldsymbol{y})$ satisfies also constraint (Qiii). Since $\bar{\tau}(y_j) + \nu(y_j - y_{j-1}) - \tau(y_{j-1}) \le h$, the solution also satisfies constraint (Qvi) and since $\tau(\cdot)$ is nondecreasing, it satisfies constraint (Qiv). Therefore, it is a feasible solution of (Q_h) and the existence of a feasible solution of (S_h) implies the existence of a feasible solution of (Q_h).

For the converse implication, suppose that (Q_h) admits a feasible solution, and consider the optimization problem consisting in maximizing $\sum_{j=1}^{S} y_j$ over its feasible solutions. These feasible solutions form a compact set of $\mathbb{R}_+^S$ since it is obviously bounded and since the semicontinuities of $\tau(\cdot)$ and $\bar{\tau}(\cdot)$ imply that it is closed. There is thus an optimal solution $(\boldsymbol{d}^*, \boldsymbol{y}^*)$ to that optimization problem. Suppose for a contradiction that there is a j such that $y_j^* < \sup \mathscr{S}_j^h$. Denote j_0 the largest such index. Let us slightly increase $y_{j_0}^*$, while letting the other y_j^* untouched. Redefine d_j^* to be $h + \tau(y_{j-1}^*)$ for all $j \geq j_0$. The pair $(\boldsymbol{d}^*, \boldsymbol{y}^*)$ remains feasible for (Q_h) (we use here the fact that $\sup \mathscr{S}_j^h$ is nondecreasing with j), while increasing the quantity $\sum_{j=1}^{S} y_j^*$, which is a contradiction with the optimality assumption. Thus, we have $y_j^* = \sup \mathscr{S}_j^h$ for all j and $d_j^* := h + \tau(y_{j-1}^*)$ for all j provides a feasible solution for (S_h).

7.5.2 Minimizing the average waiting time

The algorithm

The following map will be useful in the description of the algorithm.

$$f^{\mathrm{ave}} : (d, y, y') \longmapsto \int_y^{y'} (d - \bar{\tau}(u))\, du.$$

Define the directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ by

$$\begin{aligned}
\mathscr{V} &= \{(0,0)\} \cup \{\eta, 2\eta, \ldots, M\eta\} \times \{\eta, 2\eta, \ldots, R\eta\} \\
\mathscr{A} &= \left\{ \big((z,r),(z',r')\big) \in \mathscr{V}^2 \colon r + z' = r',\ z' > 0, \bar{\tau}(r') - \bar{\tau}(r) + \nu(z' - z) + \frac{1}{2}\gamma\eta \geq 0 \right\},
\end{aligned}$$

where we use the following notations:

$$\alpha = \inf_{t \in [0,T)} D'_+(t), \quad R = \left\lfloor \frac{D(T)M}{C} \right\rfloor,$$
$$\eta = \frac{C}{M}, \quad \text{and} \quad \gamma = 2\left(\frac{1}{\alpha} + 2\nu\right).$$

Set for each arc $a = \big((z,r),(z',r')\big)$ a weight $w(a) = f^{\mathrm{ave}}\big(\bar{\tau}(r') + \nu(z' - \eta), r + \eta, r'\big)$.

If $CS < D(T)$, there is no feasible solution. We can thus assume that $CS \geq D(T)$. The algorithm consists first in computing a path $\tilde{p}$ minimizing $\sum_{a \in \mathscr{A}(p)} w(a)$, among all paths p with at most S arcs starting at $(0,0) \in \mathscr{V}$ and ending at a vertex (z,r) with $r = R\eta$. Such paths exist, see Lemma 10 below. The computation of $\tilde{p}$ can be done in $O(S|\mathscr{A}|)$ via dynamic programming. Let the vertex sequence of $\tilde{p}$ be $\big((z_0,r_0),(z_1,r_1),\ldots,(z_n,r_n)\big)$. The

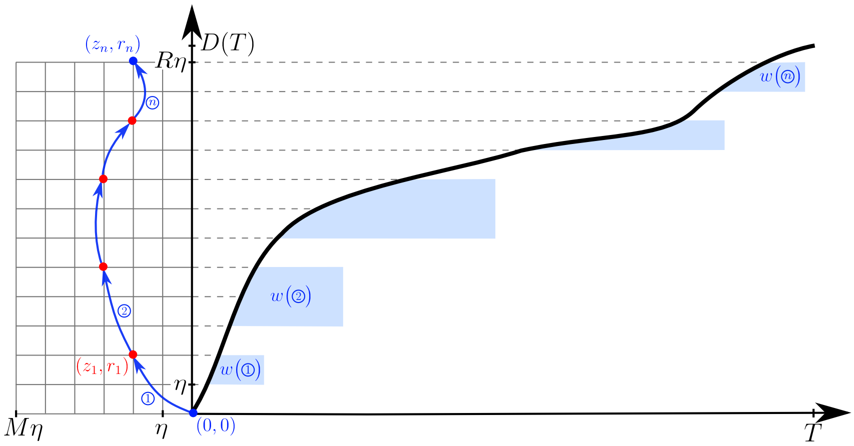

Figure 7.1 – A feasible path in the algorithm proposed for solving $\mathrm{P}^{\text{ave}}_{\text{no return}}$.

algorithm consists then in defining recursively

$$\tilde{y}_j = \begin{cases} 0 & \text{for } j = 0 \\ \min\left(r_j + \eta, \tilde{y}_{j-1} + C, D(T)\right) & \text{for } j \in \{1, \ldots, n\} \\ D(T) & \text{for } j \in \{n+1, \ldots, S\} \end{cases}$$

and

$$\tilde{d}_j = \begin{cases} \bar{\tau}(\tilde{y}_j) + \nu(\tilde{y}_j - \tilde{y}_{j-1}) + j\gamma\eta & \text{for } j \in \{1, \ldots, n\} \\ \max\left(\tilde{d}_n, T + \nu(\tilde{y}_{n+1} - \tilde{y}_n)\right) & \text{for } j \in \{n+1, \ldots, S\} \end{cases}$$

and outputting the pair $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$. The construction of the graph is sketched on Figure 7.1.

As it will be shown below, this $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is a feasible solution of $\mathrm{P}^{\text{ave}}_{\text{no return}}$ providing a value to the objective function within a $O\left(\frac{S^2}{M}\right)$ gap to the optimal value.

Proof of Theorem 4

We first give 2 technical Lemmas.

Lemma 8. *We have* $r_j \leq \tilde{y}_j \leq r_j + \eta$ *for* $j \in \{0, \ldots, n\}$.

Proof. We have $\tilde{y}_j \leq r_j + \eta$ by definition. Using $r_j - r_{j-1} \leq M\eta$ in a feasible path, a direct induction shows that $\tilde{y}_j \geq r_j$ for $j \in \{0, \ldots, n\}$. □

Lemma 9. *Suppose that* $\alpha > 0$. *Then for all* $y \in [0, D(T)]$ *and* $\delta \in [0, D(T) - y]$, *we have* $\bar{\tau}(y + \delta) \leq \bar{\tau}(y) + \delta/\alpha$ *and* $\tau(y + \delta) \leq \tau(y) + \delta/\alpha$.

Proof. Diewert [22] extended the Mean Value Theorem to semicontinuous functions.

According to his result, for any $0 \le a \le b \le T$, there exists $c \in [a,b)$ such that

$$\limsup_{t\to 0^+} \frac{D(c+t)-D(c)}{t} \le \frac{D(b)-D(a)}{b-a}.$$

Since

$$\alpha = \inf_{t\in[0,T)} D'_+(t) \le D'_+(c)$$

and thus

$$\alpha \le \limsup_{t\to 0^+} \frac{D(c+t)-D(c)}{t},$$

we have $D(a)+\alpha(b-a) \le D(b)$ for any $0 \le a \le b \le T$. With $a = \bar\tau(y)$ and $b = \bar\tau(y)+\delta/\alpha$, we get $y+\delta \le D(\bar\tau(y))+\delta \le D(\bar\tau(y)+\delta/\alpha)$ (the first inequality is given by Lemma 1). By definition of $\bar\tau$, we have $\bar\tau(y+\delta) \le \bar\tau(y)+\delta/\alpha$. The second inequality is proved along the same lines. □

We provide now three lemmas and Theorem 4 results immediately from their combination.

In the proofs of the lemmas, we assume that M is large enough so that $\eta < D(T)$. Since in Theorem 4, M appears in 'big O' formulas, it is a valid assumption. Anyway it is what is sought in practice: the larger M, the larger the accuracy of the solution. An η of same order of magnitude of $D(T)$ would be useless.

Lemma 10. *There exists a path p with at most S arcs starting at $(0,0) \in \mathcal{V}$ and ending at a vertex (z,r) with $r = R\eta$ and such that*

$$\frac{1}{D(T)} \sum_{a\in A(p)} w(a) \le OPT.$$

Proof. Let $(\boldsymbol{d}^*, \boldsymbol{y}^*)$ be an optimal solution of $\mathrm{P}^{\mathrm{ave}}_{\text{no return}}$ such that $d_j^* = \bar\tau(y_j^*) + \nu(y_j^* - y_{j-1}^*)$ for all $j \in \{1,\ldots,S\}$ (Lemma 5). Consider the sequence $\lfloor y_1^*/\eta \rfloor\eta, \ldots, \lfloor y_S^*/\eta\rfloor\eta$ and remove the repetitions. Since the sequence is nondecreasing, we obtain an increasing sequence $\boldsymbol{r} = r_1,\ldots,r_n$. We introduce $\sigma\colon \{1,\ldots,n\} \to \{1,\ldots,S\}$ with $\sigma(j)$ being the smallest index such that $r_j = \lfloor y^*_{\sigma(j)}/\eta\rfloor\eta$. We then define $z_j = r_j - r_{j-1}$ for $j \in \{1,\ldots,n\}$, with $r_0 = 0$. We prove that the sequence $(z_j, r_j)_{j\in\{1,\ldots,n\}}$ provides a feasible path from the vertex $(0,0)$ to (z_n, r_n) in $\mathcal{G}$. First note that $r_n = R\eta$ since $y_S^* = D(T)$ and that $z_j > 0$. For all $j \in \{1,\ldots,n\}$, we have

$$\begin{aligned} z_j &= r_j - r_{j-1} \\ &= \left(\left\lfloor \frac{y^*_{\sigma(j)}}{\eta}\right\rfloor - \left\lfloor \frac{y^*_{\sigma(j)-1}}{\eta}\right\rfloor + \left\lfloor \frac{y^*_{\sigma(j)-1}}{\eta}\right\rfloor - \left\lfloor \frac{y^*_{\sigma(j-1)}}{\eta}\right\rfloor\right)\eta \\ &< M\eta + \eta, \end{aligned}$$

since $\lfloor y^*_{\sigma(j)-1}/\eta\rfloor = \lfloor y^*_{\sigma(j-1)}/\eta\rfloor$ and $y^*_{\sigma(j)} - y^*_{\sigma(j)-1} \le C$. Thus $z_j \le M\eta$. Moreover by definition, $r_j \le R\eta$. Therefore $(z_j, r_j) \in \mathcal{V}$ for all $j \in \{1,\ldots,n\}$. Let us now prove that $\big((z_{j-1}, r_{j-1}), (z_j, r_j)\big) \in \mathcal{A}$ for all $j \in \{2,\ldots,n\}$. By definition, $z_j + r_{j-1} = r_j$. Note that because of the definition of r_j, we have $r_j \le y^*_{\sigma(j)} \le y^*_{\sigma(j+1)-1} < r_j + \eta$. Combining these inequalities for all j with Lemma 9 leads to

$$\begin{aligned}
\bar{\tau}(r_j)-\bar{\tau}(r_{j-1})+\nu(z_j-z_{j-1}) &\geq \bar{\tau}\left(y^*_{\sigma(j)}\right)-\frac{\eta}{\alpha}-\bar{\tau}\left(y^*_{\sigma(j-1)}\right)\\
&\quad +\nu\left(y^*_{\sigma(j)}-y^*_{\sigma(j)-1}-y^*_{\sigma(j-1)}+y^*_{\sigma(j-1)-1}-2\eta\right)\\
&= d^*_{\sigma(j)}-d^*_{\sigma(j-1)}-\left(\frac{1}{\alpha}+2\nu\right)\eta\\
&\geq -\left(\frac{1}{\alpha}+2\nu\right)\eta.
\end{aligned}$$

The sequence $(z_j,r_j)_{j\in\{1,\ldots,n\}}$ is then a feasible path p from the vertex $(0,0)$ to (z_n,r_n) in $\mathscr{G}$, with at most S arcs. The only thing that remains to be checked in that the claimed inequality holds.

We have $f^{\mathrm{ave}}\left(d^*_{\sigma(j)},y^*_{\sigma(j)-1},y^*_{\sigma(j)}\right)\geq f^{\mathrm{ave}}\big(\bar{\tau}(r_j)+\nu(z_j-\eta),r_{j-1}+\eta,r_j\big)$ for all $j\in\{1,\ldots,n\}$ since $f^{\mathrm{ave}}(\cdot)$ is nonincreasing in the second term and nondecreasing in the first and third terms. Thus,

$$\begin{aligned}
\sum_{a\in A(p)} w(a) &\leq \sum_{j=1}^{n} f^{\mathrm{ave}}\left(d^*_{\sigma(j)},y^*_{\sigma(j)-1},y^*_{\sigma(j)}\right)\\
&\leq D(T)g^{\mathrm{ave}}(\boldsymbol{d}^*,\boldsymbol{y}^*).
\end{aligned}$$

□

Lemma 11. *The pair* $(\tilde{\boldsymbol{d}},\tilde{\boldsymbol{y}})$ *is a feasible solution of* $\mathrm{P}^{\mathrm{ave}}_{\text{no return}}$.

Proof. We are going to check that $(\tilde{\boldsymbol{d}},\tilde{\boldsymbol{y}})$ is feasible for $\mathrm{P}^{\mathrm{ave}}_{\text{no return}}$.

For $j\in\{1,\ldots,n\}$, we have $\tilde{y}_j-\tilde{y}_{j-1}\leq C$ by definition of $\tilde{\boldsymbol{y}}$. For $j=n+2,\ldots,S$, we have $\tilde{y}_j-\tilde{y}_{j-1}=0$. Finally, we have $\tilde{y}_{n+1}-\tilde{y}_n\leq D(T)-r_n<\eta\leq C$ (where we use Lemma 8 to bound $\tilde{y}_n$). Thus, $\tilde{\boldsymbol{y}}$ satisfies constraint (i).

For $j\in\{1,\ldots,n\}$, we have $r_j>r_{j-1}$ and thus $\tilde{y}_{j-1}\leq r_{j-1}+\eta\leq r_j\leq\tilde{y}_j$ (the last inequality being Lemma 8). Thus, $\tilde{\boldsymbol{y}}$ satisfies constraint (ii).

Consider $j\in\{2,\ldots,n\}$. We have

$$\begin{aligned}
\tilde{d}_j-\tilde{d}_{j-1} &= \bar{\tau}(\tilde{y}_j)+\nu(\tilde{y}_j-\tilde{y}_{j-1})-\tau(\tilde{y}_{j-1})-\nu(\tilde{y}_{j-1}-\tilde{y}_{j-2})+\gamma\eta\\
&\geq \bar{\tau}(r_j)-\bar{\tau}(r_{j-1}+\eta)+\nu(r_j-2r_{j-1}+r_{j-2}-2\eta)+\gamma\eta\\
&\geq \bar{\tau}(r_j)-\bar{\tau}(r_{j-1})-\frac{\eta}{\alpha}+\nu(z_j-z_{j-1}-2\eta)+\gamma\eta\\
&\geq 0.
\end{aligned}$$

The first inequality is obtained with the help of Lemma 8. For the second one, we use Lemma 9 and also that $z_j=r_j-r_{j-1}$ and $z_{j-1}=r_{j-1}-r_{j-2}$ which hold because

$$\tilde{p}=\big((z_0,r_0),(z_1,r_1),\ldots,(z_n,r_n)\big)$$

is a path in $\mathscr{G}$. For the last inequality, we use $\bar{\tau}(r_j)-\bar{\tau}(r_{j-1})+\nu(z_j-z_{j-1})+\frac{1}{2}\gamma\eta\geq 0$, which holds again because $\tilde{p}$ is a path, and the definition of γ. For $j \geq n+1$, we have $\tilde{d}_j \geq \tilde{d}_{j-1}$ by definition. Constraint (iii) is thus satisfied for all j.

If $n < S$, then $\tilde{y}_S = D(T)$ by definition. From now on, we suppose thus that $n = S$. We also suppose that $S \geq 2$. The case $S = 1$ being easy to check (and anyway, for a complexity point of view, this case does not matter). If $\tilde{y}_{S-1} = r_{S-1}+\eta$, then $\tilde{y}_{S-1}+C = r_{S-1}+\eta+C \geq r_S+\eta > D(T)$ (here we use that $z_S \leq C$ and that $r_S = R\eta$) and thus $\tilde{y}_S = D(T)$. If $\tilde{y}_{S-1} = D(T)$, then $\tilde{y}_S = D(T)$ since $\tilde{y}_{S-1} \leq \tilde{y}_S \leq D(T)$. Hence, in all these cases, $\tilde{\boldsymbol{y}}$ satisfies constraint (iv). The only remaining case is when $\tilde{y}_{S-1} = \tilde{y}_{S-2}+C$. If j is an index in $\{1,\ldots,S-2\}$ such that $\tilde{y}_j = r_j+\eta$, then we have $r_{j+1}+\eta \leq r_j+C+\eta = \tilde{y}_j+C$ and $r_{j+1}+\eta \leq D(T)$, and thus $\tilde{y}_{j+1} = r_{j+1}+\eta$. It implies that as soon as some $j_0 \in \{1,\ldots,S-1\}$ is such that $\tilde{y}_{j_0} = r_{j_0}+\eta$, we have $\tilde{y}_{S-1} = r_{S-1}+\eta$, which is a case we have already dealt with. Since $r_j+\eta \leq r_S \leq D(T)$ for $j \in \{1,\ldots,S-1\}$, we are left with the case where $\tilde{y}_j = \tilde{y}_{j-1}+C$ for every $j \in \{1,\ldots,S-1\}$. In this situation, we have $\tilde{y}_{S-1} = (S-1)C$ and hence $\tilde{y}_{S-1}+C = CS \geq D(T)$. Since $r_S+\eta > D(T)$, we get that $\tilde{y}_S = D(T)$, and $\tilde{\boldsymbol{y}}$ satisfies constraint (iv) in every case.

For $j \in \{1,\ldots,n\}$, we have $\tilde{d}_j \geq \bar{\tau}(\tilde{y}_j)+\nu(\tilde{y}_j-\tilde{y}_{j-1})$ by definition, and for $j \geq n+1$, we have $\tilde{d}_j \geq T+\nu(\tilde{y}_{n+1}-\tilde{y}_n) \geq \bar{\tau}(\tilde{y}_j)+\nu(\tilde{y}_j-\tilde{y}_{j-1})$. Thus $\tilde{\boldsymbol{d}}$ satisfies constraint (v) and $(\tilde{\boldsymbol{d}},\tilde{\boldsymbol{y}})$ is feasible for $\mathrm{P}^{\mathrm{ave}}_{\text{no return}}$. □

Lemma 12. *The following inequality holds:*

$$g^{\mathrm{ave}}(\tilde{\boldsymbol{d}},\tilde{\boldsymbol{y}}) \leq \frac{1}{D(T)}\sum_{a\in A(\tilde{p})} w(a) + O\left(\frac{S^2}{M}\right).$$

Proof. Our goal is to bound from above the following quantity

$$g^{\mathrm{ave}}(\tilde{\boldsymbol{d}},\tilde{\boldsymbol{y}}) = \frac{1}{D(T)}\sum_{j=1}^{S} f^{\mathrm{ave}}(\tilde{d}_j,\tilde{y}_{j-1},\tilde{y}_j) \tag{7.4}$$

We proceed by splitting the expression into two parts: the sum from $j = 1$ to $j = n$, and the sum from $j = n+1$ to $j = S$.

Using Lemmas 8 and 9, we have $\bar{\tau}(\tilde{y}_j)+\nu(\tilde{y}_j-\tilde{y}_{j-1}) \leq q_j+\eta/\alpha+\nu\eta$, where $q_j = \bar{\tau}(r_j)+\nu(r_j-r_{j-1})$. Thus we have for all $j \leq n$,

$$\sum_{j=1}^{n} f^{\mathrm{ave}}(\tilde{d}_j,\tilde{y}_{j-1},\tilde{y}_j) \leq \sum_{j=1}^{n} f^{\mathrm{ave}}\left(q_j+\frac{\eta}{\alpha}+\nu\eta+j\gamma\eta, r_{j-1}, r_j+\eta\right), \tag{7.5}$$

since $f^{\mathrm{ave}}(\cdot)$ is nonincreasing in the second term and nondecreasing in the first and third terms and where we extend the definition of $\bar{\tau}(\cdot)$ by letting $\bar{\tau}(y) = T$ for all $y > D(T)$.

For the second part, we proceed as follows. Since $r_n+\eta = (R+1)\eta > D(T)$, Lemma 8 immediately implies $D(T)-\tilde{y}_n \leq \eta$. With Lemma 9, we get thus $T \leq \bar{\tau}(\tilde{y}_n)+\eta/\alpha$, where we used $T = \bar{\tau}\big(D(T) - \tilde{y}_n + \tilde{y}_n\big)$. This provides

$$\begin{aligned} \tilde{d}_{n+1} &\le \bar{\tau}(\tilde{y}_n) + \frac{\eta}{\alpha} + \nu(r_n - r_{n-1}) + \nu\eta + n\gamma\eta \\ &= q_n + \left(\frac{1}{\alpha} + \nu + n\gamma\right)\eta. \end{aligned}$$

Using again the fact that $f^{\text{ave}}(\cdot)$ is nonincreasing in the second term and nondecreasing in the first and third terms and with the help of Lemma 8, we get

$$\sum_{j=n+1}^{S} f^{\text{ave}}(\tilde{d}_j, \tilde{y}_{j-1}, \tilde{y}_j) \le f^{\text{ave}}\left(q_n + \frac{\eta}{\alpha} + \nu\eta + n\gamma\eta, r_n, r_n + \eta\right), \tag{7.6}$$

since the terms indexed by $j = n+2, \ldots, S$ are all zero and since $D(T) < r_n + \eta$.

We aim at comparing the upper bounds in Equations (7.5) and (7.6) with

$$\sum_{a \in A(\tilde{p})} w(a) = \sum_{j=1}^{n} f^{\text{ave}}(q_j - \nu\eta, r_{j-1} + \eta, r_j). \tag{7.7}$$

We first compare the jth term of the bound in (7.5) with the jth term of the sum in (7.7).

$$f^{\text{ave}}\left(q_j + \frac{\eta}{\alpha} + \nu\eta + j\gamma\eta, r_{j-1}, r_j + \eta\right) - f^{\text{ave}}(q_j - \nu\eta, r_{j-1} + \eta, r_j) = I_j^1 + I_j^2 + I_j^3$$

with

$$\begin{aligned} I_j^1 &= \int_{r_{j-1}}^{r_{j-1}+\eta} \left(q_j + \frac{\eta}{\alpha} + \nu\eta + j\gamma\eta - \bar{\tau}(u)\right) du \\ I_j^2 &= \int_{r_{j-1}+\eta}^{r_j} \left(j\gamma\eta + \frac{\eta}{\alpha} + 2\nu\eta\right) du \\ I_j^3 &= \int_{r_j}^{r_j+\eta} \left(q_j + \frac{\eta}{\alpha} + \nu\eta + j\gamma\eta - \bar{\tau}(u)\right) du. \end{aligned}$$

Since $\bar{\tau}(\cdot)$ in nondecreasing, we get

$$\begin{aligned} I_j^1 &\le \big(\bar{\tau}(r_j) - \bar{\tau}(r_{j-1}) + \nu(r_j - r_{j-1})\big)\eta + \left(\frac{1}{\alpha} + \nu + j\gamma\right)\eta^2 \\ I_j^2 &\le (r_j - r_{j-1})\left(j\gamma + \frac{1}{\alpha} + 2\nu\right)\eta - \left(j\gamma + \frac{1}{\alpha} + 2\nu\right)\eta^2 \\ I_j^3 &\le \nu(r_j - r_{j-1})\eta + \left(\frac{1}{\alpha} + \nu\eta + j\gamma\right)\eta^2. \end{aligned}$$

Using $j\gamma \le n\gamma$ and $\gamma = 2(1/\alpha + 2\nu)$, we obtain

$$I_j^1 + I_j^2 + I_j^3 \le \big(\bar{\tau}(r_j) - \bar{\tau}(r_{j-1}) + 2\nu(r_j - r_{j-1})\big)\eta + \left(n + \frac{1}{2}\right)\gamma\eta^2 + (r_j - r_{j-1})\left(n + \frac{1}{2}\right)\gamma\eta.$$

We now bound the term in Equation (7.6). Let $I = f^{\text{ave}}(q_n + \eta/\alpha + \nu\eta + n\gamma\eta, r_n, r_n + \eta)$. We have

$$\begin{aligned} I &= \int_{r_n}^{r_n+\eta} \left(q_n + \frac{\eta}{\alpha} + \nu\eta + n\gamma\eta - \bar{\tau}(u) \right) du. \\ &\leq \nu(r_n - r_{n-1})\eta + \left(\frac{1}{\alpha} + \nu + n\gamma \right) \eta^2. \end{aligned}$$

We have thus

$$\begin{aligned} & g^{\text{ave}}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) - \frac{1}{D(T)} \sum_{a \in A(\tilde{p})} w(a) \\ \leq \;& \frac{1}{D(T)} \left(\sum_{j=1}^{n} \left(I_j^1 + I_j^2 + I_j^3 \right) + I \right) \\ \leq \;& \frac{1}{D(T)} \left(\bar{\tau}(r_n) + 2\nu r_n + r_n(n+1)\gamma + \nu C + (n+1)^2 \gamma\eta \right) \eta. \end{aligned}$$

Using $r_n \leq D(T)$ and $\bar{\tau}(r_n) \leq T$ leads to

$$\begin{aligned} g^{\text{ave}}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) \leq & \frac{1}{D(T)} \sum_{a \in A(\tilde{p})} w(a) \\ & + \left(\frac{T + \nu C}{D(T)} + \gamma(S+1) \right) \eta + \frac{\gamma(S+1)^2}{D(T)} \eta^2. \end{aligned}$$

□

7.5.3 When the demand function is a step function

Better complexity results can be obtained when the demand is a step function. A *step function* is a function that can be written as a finite linear combination of indicator functions of intervals. The assumption of $D(\cdot)$ being a step function means that the users arrive only on a finite number of instants. As it has already been noted, the assumption $\nu = 0$ is equivalent to the assumption that every user boards a shuttle as soon as he arrives in the terminal.

Proposition 6. *Assume that $D(\cdot)$ is a step function defined with K discontinuities, supposed to be part of the input. Suppose moreover that $\nu = 0$. Then for each of $\mathrm{P}^{\max}_{\text{no return}}$ and of $\mathrm{P}^{\text{ave}}_{\text{no return}}$, there is an algorithm computing an optimal solution in $O(K^2 S)$.*

It turns out that when C and the values taken by $D(\cdot)$ are integer, the loads of the shuttles in the optimal solution returned by the algorithm are also integer. We cover thus the case where the users are atoms.

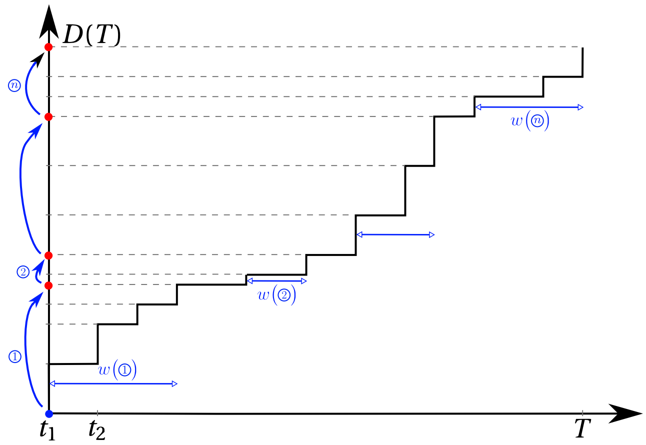

Figure 7.2 – A feasible path in the algorithm proposed for solving $\mathrm{P}^{\max}_{\text{no return}}$ when the the demand function is a step function

The algorithm

We provide only the algorithm for $\mathrm{P}^{\max}_{\text{no return}}$, the other case can be dealt with similarly. Let $t_1 < \cdots < t_K$ be the K discontinuities. Define the directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{A})$ by

$$\begin{aligned} \mathscr{V} &= \{Cq\colon q \in \{0,1,\ldots,Q\}\} \cup \{D(t_k) + Cq\colon k \in \{1,\ldots,K\}, q \in \{0,1,\ldots,Q\}\} \\ \mathscr{A} &= \{(y,y') \in \mathscr{V}^2\colon 0 \le y' - y \le C\}, \end{aligned}$$

where $Q = \lfloor D(T)/C \rfloor$. Note that the vertex set is a finite subset of $\mathbb{R}_+$. Set for each arc $a = (y, y')$ a weight $w(a) = \bar{\tau}(y') - \tau(y)$. Note that this weight may be negative when $y = y'$ (a loop in $\mathscr{G}$) but this will not cause any issue. We consider the two vertices 0 and $D(T)$ (obtained with $k = K$ and $q = 0$).

If $CS < D(T)$, there is no feasible solution. We can thus assume that $CS \ge D(T)$. The algorithm consists first in computing a 0-$D(T)$ path $\tilde{p}$ with S arcs minimizing $\max_{a \in \mathscr{A}(\tilde{p})} w(a)$. Within the proof of Proposition 6 below, we show that from any feasible solution we can build a 0-$D(T)$ path with S arcs in $\mathscr{G}$. Thus, when the problem is feasible, such paths exist in $\mathscr{G}$. The computation of $\tilde{p}$ can be done in $O(S|\mathscr{A}|)$ via dynamic programming. Let the vertex sequence of $\tilde{p}$ be $(\tilde{y}_0, \tilde{y}_1, \ldots, \tilde{y}_S)$. The end of the algorithm consists in defining $\tilde{d}_j = \bar{\tau}(\tilde{y}_j)$ for all $j \in \{1,\ldots,S\}$ and outputting the pair $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$. The construction of the graph is sketched on Figure 7.2.

As it will be shown below, this $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is an optimal solution of $\mathrm{P}^{\max}_{\text{no return}}$.

Proof of Proposition 6

According to Lemma 4, we replace the objective function of $\mathrm{P}^{\max}_{\text{no return}}$ by $\max_{j \in \{1,\ldots,S\}} (d_j - \tau(y_{j-1}))$. It can easily be checked that $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is feasible for $\mathrm{P}^{\max}_{\text{no return}}$. It provides a value

$\max_{j\in\{1,\ldots,S\}}(\bar{\tau}(\tilde{y}_j)-\tau(\tilde{y}_{j-1}))$ for $\mathrm{P}^{\max}_{\text{no return}}$ (with the alternative objective function), which coincides with $\max_{a\in\mathscr{A}(\tilde{p})} w(a)$. The path $\tilde{p}$ describes therefore a solution of $\mathrm{P}^{\max}_{\text{no return}}$ with a value equal to $\max_{a\in\mathscr{A}(\tilde{p})} w(a)$.

Conversely, let $(\boldsymbol{d},\boldsymbol{y})$ be any feasible solution of $\mathrm{P}^{\max}_{\text{no return}}$. Let $\bar{\boldsymbol{y}}$ be the sequence defined by $\bar{y}_j=\min\{y\in\mathscr{V}\colon y\geq y_j\}$. On the one hand, we have $\bar{y}_{j-1}\leq\bar{y}_j$ because $y_{j-1}\leq y_j$. On the other hand, we have $\bar{y}_{j-1}+C\geq y_{j-1}+C\geq y_j$. If $\bar{y}_{j-1}+C\in\mathscr{V}$, we have $\bar{y}_{j-1}+C\geq\bar{y}_j$ by definition of $\bar{y}_j$. If $\bar{y}_{j-1}+C\notin\mathscr{V}$, then $\bar{y}_{j-1}+C>D(T)\geq\bar{y}_j$ since $D(T)\in\mathscr{V}$. Thus, $(\bar{y}_{j-1},\bar{y}_j)\in\mathscr{A}$ for all $j\in\{1,\ldots,S\}$. We have $\bar{y}_0=0$ and $\bar{y}_S=D(T)$ and the sequence $\bar{\boldsymbol{y}}$ is a 0-$D(T)$ path p with S arcs.

Second, we prove that $\bar{\tau}(\bar{y}_j)-\tau(\bar{y}_{j-1})\leq d_j-\tau(y_{j-1})$ as follows. There exists a unique k such that $D(t_k)<y_j\leq D(t_{k+1})$. By definition of $D(\cdot)$, we have $D(t)=D(t_k)$ for all $t\in[t_k,t_{k+1})$, and thus $\bar{\tau}(y_j)=t_{k+1}$. Since $D(t_{k+1})\in\mathscr{V}$, we have $\bar{y}_j\leq D(t_{k+1})$ by definition of $\bar{y}_j$, and hence $\bar{\tau}(\bar{y}_j)\leq t_{k+1}$ (directly by definition of $\bar{\tau}(\cdot)$). Therefore, $\bar{\tau}(\bar{y}_j)-\tau(\bar{y}_{j-1})\leq\bar{\tau}(y_j)-\tau(y_{j-1})\leq d_j-\tau(y_{j-1})$ (where we use the fact that $\tau(\cdot)$ is nondecreasing).

Finally, we have

$$\max_{a\in\mathscr{A}(p)} w(a)\leq\max_{j\in\{1,\ldots,S\}}\big(d_j-\tau(y_{j-1})\big).$$

As the path $\tilde{p}$ is optimal, $\max_{a\in\mathscr{A}(\tilde{p})} w(a)$ is a lower bound on the value taken by the (alternative) objective function on $(\boldsymbol{d},\boldsymbol{y})$.

7.6 When return is allowed

The algorithm

The following map will be useful:

$$f^{\max}\colon(\ell,y,y')\mapsto\begin{cases}\max(\ell,\bar{\tau}(y'))+\nu(y'-y)-\tau(y) & \text{if } y'\geq y\\ 0 & \text{if } y'<y.\end{cases}$$

We introduce the following two sets

$$\begin{aligned}\mathscr{Q} &= \{0,\eta,\ldots,(\lfloor T^+/\eta\rfloor+1)\eta\}^S\\ \mathscr{R} &= \big\{\boldsymbol{r}\in\{0,\eta,\ldots,R\eta\}^S\colon 0\leq r_k-r_{k-1}\leq M\eta\quad\text{for } k\in\{2,\ldots,S\}\big\},\end{aligned}$$

where

$$\eta=\frac{C}{M},\qquad R=\left\lfloor\frac{D(T)M}{C}\right\rfloor,\qquad\text{and}$$
$$T^+=T+\frac{\nu D(T)}{S}+\left(\left\lceil\frac{D(T)}{CS}\right\rceil-1\right)\pi.$$

Define the directed graph $\mathscr{G}=(\mathscr{V},\mathscr{A})$ by

$$\begin{aligned}\mathscr{V} &= \big\{(z,\boldsymbol{q},\boldsymbol{r})\in\{0,\eta,\ldots,M\eta\}\times\mathscr{Q}\times\mathscr{R}\text{ satisfying }(\star)\big\}\\ \mathscr{A} &= \big\{\big((z,\boldsymbol{q},\boldsymbol{r}),(z',\boldsymbol{q}',\boldsymbol{r}')\big)\in\mathscr{V}^2\text{ satisfying }(\star\star)\big\},\end{aligned}$$

where

$$(\star)\quad \begin{cases} r_k \le D(q_k) & \text{for } k \in \{1,\ldots,S\} \\ q_k - q_{k-1} + \nu(r_k - 2r_{k-1} + r_{k-2}) + (1+2\nu)\eta \ge 0 & \text{for } k \in \{2,\ldots,S\}. \end{cases}$$

and

$$(\star\star)\quad \begin{cases} r_S + z' = r'_1 \\ q'_1 - q_S + \nu(z' - r_S + r_{S-1}) + (1+2\nu)\eta \ge 0 \\ q'_k - q_k - \nu(r_k - r_{k-1}) - \pi + (1+\nu)\eta \ge 0 & \text{for } k \in \{1,\ldots,S\}. \end{cases}$$

We adopt the convention $D(t) = D(T)$ when $t \ge T$ and we define $r_0 = r_1 - z$. Set for each arc $a = \big((z, \boldsymbol{q}, \boldsymbol{r}), (z', \boldsymbol{q}', \boldsymbol{r}')\big)$ a weight

$$w(a) = \max_{k\in\{1,\ldots,S\}} f^{\max}(q'_k - \eta, r'_{k-1} + \eta, r'_k)$$

where $r'_0 = r'_1 - z'$.

The algorithm consists first in computing a path $\tilde{p}$ minimizing $\max_{a\in A(p)} w(a)$ among all paths p starting at $(0, \boldsymbol{0}, \boldsymbol{0}) \in \mathcal{V}$ (the 'all zero' vector) and ending at a vertex $(z, \boldsymbol{q}, \boldsymbol{r})$ with $r_S = R\eta$. Such paths exist, see Lemma 14 below. It can be done in $O(|\mathcal{V}||\mathscr{A}|)$ via dynamic programming. Let the vertex sequence of $\tilde{p}$ be

$$\big((0, \boldsymbol{0}, \boldsymbol{0}), (z_0, \boldsymbol{q}^0, \boldsymbol{r}^0), \ldots, (z_n, \boldsymbol{q}^n, \boldsymbol{r}^n)\big).$$

The vector $\boldsymbol{r}^i$ models the cumulative loads of the S shuttles when they perform their ith departure. The vector $\boldsymbol{q}^i$ models the times at which the loading of the S shuttles starts when they perform their ith departure. These quantities are computed only approximatively (with an accuracy η).

We set $\tilde{y}_0 = 0$, $r^i_0 = r^{i-1}_S$, $r^0_0 = 0$, and $\tilde{\gamma} = (1 + 4\nu + 1/\alpha)$. The algorithm consists in defining recursively for all $j = iS + k$ with $i \in \{0,\ldots,n\}$ and $k \in \{1,\ldots,S\}$

$$\begin{aligned} \tilde{y}_j &= \begin{cases} \min\big(r^i_k + \eta, \tilde{y}_{j-1} + C, D(T)\big) & \text{if } r^i_k > r^i_{k-1} \\ \tilde{y}_{j-1} & \text{otherwise} \end{cases} \\ \tilde{d}_j &= \max(q^i_k, \bar{\tau}(\tilde{y}_j)) + j\tilde{\gamma}\eta + \nu(\tilde{y}_j - \tilde{y}_{j-1}) \end{aligned}$$

and for $j = (n+1)S + 1$ if $\tilde{y}_{(n+1)S} < D(T)$

$$\begin{aligned} \tilde{y}_j &= D(T) \\ \tilde{d}_j &= \max\big(\tilde{d}_{j-S} + \pi, \tilde{d}_{(n+1)S}, T\big) + \nu(\tilde{y}_j - \tilde{y}_{j-1}). \end{aligned}$$

We then output the pair $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$.The construction of the graph is sketched on Figure 7.3.

As it will be stated below, this $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is a feasible solution of $\mathrm{P}^{\max}_{\mathrm{return}}$ providing a value to the

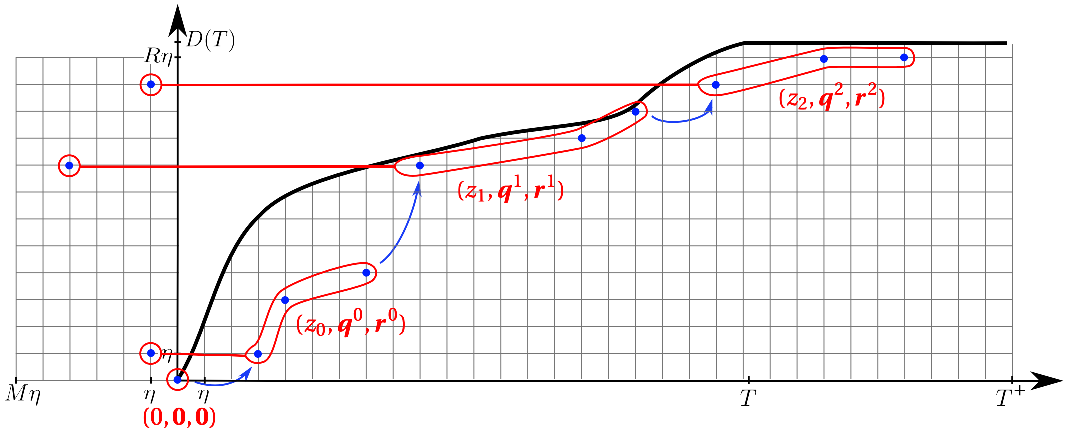

Figure 7.3 – A feasible path in the algorithm proposed for solving $\mathrm{P}^{\max}_{\mathrm{return}}$.

objective function $g^{\max}(\cdot)$ within a $O\left(\frac{S^2}{M}\right)$ gap to the optimal value.

Proof of Theorem 5

We provide four lemmas. The proof of Lemma 15 is almost the one of Proposition 5 and is thus omitted. The proofs of the three others follow the same scheme as the ones of Lemmas 10, 11, and 12. Theorem 5 results immediately from the combination of these three latter lemmas. Lemma 15 is only used for proving Lemma 14.

Lemma 13. *We have* $r^i_k \le \tilde{y}_{iS+k} \le r^i_k + \eta$ *for* $i \in \{0,\dots,n\}$ *and* $k \in \{1,\dots,S\}$.

Proof. We have $\tilde{y}_{iS+k} \le r^i_k + \eta$ by definition. Using $r^i_k - r^i_{k-1} \le M\eta$ in a feasible path, a direct induction shows that $r^i_k \le \tilde{y}_{iS+k}$. □

Lemma 14. *There exists a path p starting at $(0,\mathbf{0},\mathbf{0}) \in \mathcal{V}$ and ending at a vertex $(z,\boldsymbol{q},\boldsymbol{r})$ with $r_S = R\eta$ and such that*

$$\max_{a\in A(p)} w(a) \le OPT.$$

Proof. Let $(\boldsymbol{d}^*,\boldsymbol{y}^*)$ be an optimal solution of $\mathrm{P}^{\max}_{\mathrm{return}}$ such that $y^*_N > y^*_{N-1}$, where N is its number of departures. Such a solution always exists. Consider the nondecreasing sequences $\tilde{\boldsymbol{r}} = \lfloor y^*_1/\eta\rfloor\eta,\dots,\lfloor y^*_N/\eta\rfloor\eta$ and $\tilde{\boldsymbol{q}} = \lceil (d^*_1 - \nu(y^*_1 - y^*_0))/\eta\rceil\eta,\dots,\lceil (d^*_N - \nu(y^*_N - y^*_{N-1}))/\eta\rceil\eta$. Repeat the last terms $\lfloor y^*_N/\eta\rfloor\eta$ and $\lceil (d^*_N - \nu(y^*_N - y^*_{N-1}))/\eta\rceil\eta$ until the sequences have a multiple of S terms. Denote by $n = \lceil N/S\rceil - 1$, we define the sequence $\boldsymbol{r} = r^0,\dots,r^n$ and $\boldsymbol{q} = q^0,\dots,q^n$ as follows: for all $i \in \{0,\dots,n\}$,

$$\begin{aligned} r^i &= \left(\tilde{r}_{iS+1},\dots,\tilde{r}_{(i+1)S}\right) \\ q^i &= \left(\tilde{q}_{iS+1},\dots,\tilde{q}_{(i+1)S}\right). \end{aligned}$$

We denote by r_k^i (resp. q_k^i) the kth term in r^i (resp. q^i) and we define $z_i = r_1^i - r_S^{i-1}$ for all $i \in \{0,\dots,n\}$, where $r_S^{-1} = 0$. We prove that the sequence $\big((0,\mathbf{0},\mathbf{0}),(z_0,r^0,q^0),\dots,(z_n,r^n,q^n)\big)$ provides a feasible path to a vertex $(z_n,\boldsymbol{q}^n,\boldsymbol{r}^n)$ with $r_S^n = R\eta$. First note that $r_S^n = R\eta$ since $y_N^* = D(T)$ and that $z_i \le 0$. For all $i \in \{0,\dots,n\}$, we have

$$\begin{aligned} z_i &= r_1^i - r_S^{i-1} \\ &= \left(\left\lfloor \frac{y^*_{iS+1}}{\eta}\right\rfloor - \left\lfloor \frac{y^*_{iS}}{\eta}\right\rfloor\right)\eta \\ &< M\eta + \eta, \end{aligned}$$

where for all $j > N$, we set y_j to y_N. Thus $z_i \le M\eta$. We show the same way that $r^i \in \mathbb{R}$. The loading time of the jth departure is $d_j^* - \nu(y_j^* - y_{j-1}^*)$. Since the d_jth are nondecreasing, for all k and i, we have $q_k^i \le q_S^n = \lceil (d_N^* - \nu(y_N^* - y_{N-1}^*))/\eta\rceil\eta \le T^+ + \eta$ using Lemma 15 for the last inequality. For all k and i, we have $D(q_k^i) \ge D(\tau(y^*_{iS+k})) \ge y^*_{iS+k} \ge r_k^i$, where we use the feasibility of $(\boldsymbol{d}^*,\boldsymbol{y}^*)$ for the first inequality and Lemma 1 for the second. We have for $k \in \{2,\dots,S\}$

$$\begin{aligned} q_k^i - q_{k-1}^i = \nu(r_k^i - 2r_{k-1}^i + r_{k-2}^i) &\ge d^*_{iS+k} - d^*_{iS+k-1} - \eta - 2\nu\eta \\ &\ge -(1+2\nu)\eta. \end{aligned}$$

Thus, $(z_i,r^i,q^i) \in \mathscr{V}$ for all $i \in \{0,\dots,n\}$. Let us prove now that $((z_{i-1},r^{i-1},q^{i-1}),(z_i,r^i,q^i)) \in \mathscr{A}$ for all $i \in \{1,\dots,n\}$. By definition, $z_i = r_1^i - r_S^{i-1}$. The second inequality is satisfied for the same reasons as above. Note that because of the definitions of r_k^i and q_k^i, we have $r_k^i \le y^*_{iS+k} < r_k^i + \eta$ and $q_k^i - \eta < d^*_{iS+k} - \nu(y^*_{iS+k} - y^*_{iS+k-1}) \le q_k^i$, where for all $j > N$, $d_j^* = d_N^*$ and $y_j^* = y_N^*$. Combining these inequalities with the feasibility of $(\boldsymbol{d}^*,\boldsymbol{y}^*)$ for all $i \in \{1,\dots,n-1\}$ leads to

$$\begin{aligned} q_k^i - q_k^{i-1} - \nu(r_k^{i-1} - r_{k-1}^{i-1}) &\ge d^*_{iS+k} - \nu(y^*_{iS+k} - y^*_{iS+k-1}) - \eta - d^*_{(i-1)S+k} - \nu\eta \\ &\ge \pi - (1+\nu)\eta, \end{aligned}$$

where $r_0^i = r_S^{i-1}$. The sequence $\big((0,\mathbf{0},\mathbf{0}),(z_0,r^0,q^0),\dots,(z_n,r^n,q^n)\big)$ is then a feasible path p in $\mathscr{G}$ from the vertex $(0,\mathbf{0},\mathbf{0})$ to a vertex $(z_n,\boldsymbol{q}^n,\boldsymbol{r}^n)$ with $r_S^n = R\eta$. The only thing that remains to be checked in that the claimed inequality holds.

We have $f^{\max}(d^*_{iS+k}, y^*_{iS+k_1}, y^*_{iS+k}) \ge f^{\max}(q_k^i - \eta, r_{k-1}^i + \eta, r_k^i)$ for all $i \in \{0,\dots,n\}$ and $j \in \{1,\dots,S\}$ since $f^{\max}(\cdot)$ is nonincreasing in the second term and nondecreasing in the first and third terms. Thus, since $f^{\max}(\ell,y,y) = 0$ for all ℓ and y, we have

$$\begin{aligned} \max_{a\in A(p)} w(a) &\le \max_{i\in\{0,\dots,n\},k\in\{1,\dots,S\}} f^{\max}\left(d^*_{iS+k}, y^*_{iS+k-1}, y^*_{iS+k}\right) \\ &\le g^{\max}(\boldsymbol{d}^*,\boldsymbol{y}^*). \end{aligned}$$

□

Lemma 15. *For every optimal solution, T^+ is an upper bound on the loading time of the* *last nonempty departure.*

Lemma 16. *The pair $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is a feasible solution of* $\mathrm{P}^{\max}_{\mathrm{return}}$.

Proof. We are going to check that $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is feasible for $\mathrm{P}^{\max}_{\mathrm{return}}$.

For $j \in \{1, \ldots, (n+1)S\}$, we have $\tilde{y}_j - \tilde{y}_{j-1} \leq C$ by definition. If $\tilde{y}_{(n+1)S} < D(T)$, we have $\tilde{y}_{(n+1)S+1} - \tilde{y}_{(n+1)S} \leq D(T) - r^n_S < \eta \leq C$ (where we use Lemma 13 to bound $\tilde{y}_{(n+1)S}$). Thus, $\tilde{\boldsymbol{y}}$ satisfies constraint (i).

For $i \in \{0, \ldots, n\}$ and $k \in \{1, \ldots, S\}$, if $r^i_k > r^i_{k-1}$, we have $\tilde{y}_{iS+k-1} \leq r^i_{k-1} + \eta \leq r^i_k \leq \tilde{y}_{iS+k}$ (with Lemma 13). If $r^i_{k-1} = r^i_k$, we have by defintion $\tilde{y}_{iS+k} \geq \tilde{y}_{iS+k-1}$. If $\tilde{y}_{(n+1)S} < D(T)$, the constraint is also satisfied and constraint (ii) is satisfied.

For $i \in \{0, \ldots, n\}$ and $k \in \{1, \ldots, S\}$, we have $\tau(\tilde{y}_{iS+k} \leq \tau(r^i_k) + \eta/\alpha$ with Lemmas 13 and 9. Thus, $\max\left(q^i_k, \tau(y_{iS+k})\right) \leq q^i_k + \eta/\alpha$ and $\tilde{d}_{iS+k} - d_{iS+k-1} \geq q^i_k - q^i_{k-1} - \eta/\alpha + \nu(r^i_k - 2r^i_{k-1} + r^i_{k-2} - 2\eta) + \tilde{\gamma}\eta \geq 0$. If $\tilde{y}_{(n+1)S} < D(T)$, the constraint is satisfied by defintion and $\tilde{\boldsymbol{d}}$ satisfies constraint (iii).

If $\tilde{y}_{(n+1)S} < D(T)$, the $\tilde{y}_{(n+1)S+1} = D(T)$ by definition and constraint (iv) is satisfied.

For $j \in \{1, \ldots, (n+1)S\}$, $\tilde{d}_j \geq \tau(\tilde{y}_j) + \nu(\tilde{y}_j - \tilde{y}_{j-1})$ bu definition. If $\tilde{y}_{(n+1)S} < D(T)$, we have $\tau(\tilde{y}_{(n+1)S}) = T$ and constraint is also satisfied. Thus, $\tilde{\boldsymbol{d}}$ satisfied constraint (v).

For $j \in \{1, \ldots, (n+1)S\}$, we have with Lemmas 13 and 9

$$
\begin{aligned}
\tilde{d}_{iS+k} - \tilde{d}_{(i-1)S+k} - \nu(\tilde{y}_{iS+k} - \tilde{y}_{iS+k-1}) \quad &\geq \quad q^i_k - q^{i-1}_k - \frac{1}{\alpha} - \nu(r^{i-1}_k - r^{i-1}_{k-1} - \eta) \\
&\quad + \left(1 + 2\nu + \frac{1}{\alpha}\right)\eta \\
&\geq \quad \pi.
\end{aligned}
$$

If $\tilde{y}_{(n+1)S} < D(T)$, the constraint is satisfied by definition and constraint (vi) is satisfied. Thus $(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}})$ is feasible for $\mathrm{P}^{\max}_{\mathrm{return}}$. □

Lemma 17. *The following inequality holds:*

$$
g^{\max}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) \leq \max_{a \in A(\tilde{p})} w(a) + O\left(\frac{S^2}{M}\right).
$$

Proof. Out goal is to bound from above the following quantity

$$
\begin{aligned}
g^{\max}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) \quad &= \quad \max_{\tilde{y}_{j-1} < \tilde{y}_j} (\tilde{d}_j - \tau(\tilde{y}_{j-1})) \\
&= \quad \max_{\tilde{y}_{j-1} < \tilde{y}_j} f^{\max}(\tilde{d}_j - \nu(y_j - y_{j-1}), y_{j-1}, y_j).
\end{aligned}
\tag{7.8}
$$

For $j = iS + k$ such that $\tilde{y}_{j-1} < \tilde{y}_j$, we have $r^i_{k-1} < r^i_k$ (where r^i_0 is set to r^{i-1}_S). We have $\tilde{d}_j - \nu(\tilde{y}_{j-1} - \tilde{y}_j) = \max(q^i_k, \bar{\tau}(\tilde{y}_j)) + j\tilde{\gamma}\eta \leq q^i_k + \eta/\alpha + j\tilde{\gamma}\eta$ using Lemma 9. With Lemma 13,

$\tilde{y}_{j-1} \geq r^i_{k-1}$, $\tilde{y}_j \leq r^i_k + \eta$, and thus

$$f^{\max}(\tilde{d}_j - v(y_j - y_{j-1}), y_{j-1}, y_j) \leq f^{\max}\left(q^i_k + \frac{\eta}{\alpha} + j\tilde{\gamma}\eta, r^i_{k-1}, r^i_k + \eta\right) \tag{7.9}$$

since $f^{\max}(\cdot)$ is nonincreasing in the second term and nondecreasing in the first and third terms. We have with Equations (7.8) and (7.9):

$$g^{\max}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) \leq \max_{i,k:\ r^i_{k-1} < r^i_k} f^{\max}\left(q^i_k + \frac{\eta}{\alpha} + (iS+k)\tilde{\gamma}\eta, r^i_{k-1}, r^i_k + \eta\right). \tag{7.10}$$

The only remaining term is if $\tilde{y}_{(n+1)S} < D(T)$, then $\tilde{y}_{(n+1)S+1} - \tilde{y}_{(n+1)S} > 0$. This case only happens when $\tilde{y}_{(n+1)S} = R\eta$ is a multiplier of C. Increasing the value of M can avoid this case, since M appears in 'big O' formulas, it is a valid assumption.

We have $\max_{a \in A(\tilde{p})} w(a) = \max_{i \in \{0,\dots,n\}} \max_{k \in \{1,\dots,S\}} f^{\max}(q^i_k - \eta, r^i_{k-1} + \eta, r^i_k)$. We compare this equation with the bound in Equation (7.10), for i, k such that $r^i_{k-1} < r^i_k$:

$$\begin{aligned}
& f^{\max}\left(q^i_k + \frac{\eta}{\alpha} + (iS+k)\tilde{\gamma}\eta, r^i_{k-1}, r^i_k + \eta\right) - f^{\max}(q^i_k - \eta, r^i_{k-1} + \eta, r^i_k) \\
\leq \quad & q^i_k + \frac{\eta}{\alpha} + (iS+k)\tilde{\gamma}\eta + v(r^i_k + \eta - r^i_{k-1}) - \tau(r^i_{k-1}) - q^i_k + \eta \\
& -v(r^i_k - r^i_{k-1} - \eta) + \tau(r^i_{k-1} + \eta) \\
\leq \quad & (iS + k + 2)\tilde{\gamma}\eta.
\end{aligned}$$

We have thus with Equation (7.10):

$$\begin{aligned}
g^{\max}(\tilde{\boldsymbol{d}}, \tilde{\boldsymbol{y}}) \quad \leq \quad & \max_{i,k:\ r^i_{k-1} < r^i_k} \left(f^{\max}(q^i_k - \eta, r^i_{k-1} + \eta, r^i_k) + \frac{\eta}{\alpha} + (iS + k + 1)\tilde{\gamma}\eta\right) \\
\leq \quad & \max_{i,k} \left(f^{\max}(q^i_k - \eta, r^i_{k-1} + \eta, r^i_k) + (iS + k + 2)\tilde{\gamma}\eta\right) \\
\leq \quad & \max_{a \in A(\tilde{p})} w(a) + (n+1)S((n+1)S + 2)\tilde{\gamma}\eta.
\end{aligned}$$

With Proposition 5, we have the conclusion. □

7.7 Experimental results

In this section, we test the performance of the algorithms described in previous sections for problems $\mathrm{P}^{\max}_{\text{no return}}$, $\mathrm{P}^{\text{ave}}_{\text{no return}}$, and $\mathrm{P}^{\max}_{\text{return}}$. As explained in Section 7.3, we do not have such an algorithm for problem $\mathrm{P}^{\text{ave}}_{\text{return}}$.

7.7.1 Data

Our experiments are based on real data provided by our partner Eurotunnel. They are related to the transportation of freight trucks between France and Great Britain. Some parameters are fixed as constants of the problems and do not vary from an instance to another. For the constants C, T, v, π of our problem, we take the real values used in practice by the company:

$$C = 32, \qquad T = 1440\,\text{min (one day)},$$
$$\nu = 0.625\,\text{min}, \qquad \pi = 34\,\text{min}.$$

(The value taken for π is actually the duration of a trip going from France to Great-Britain, and not of the round trip, which lasts approximatively twice this quantity.)

Two functions $D(\cdot)$ are used. The first one ("1P") is a piecewise affine map obtained by averaging the real demand over several days. It turns out that this function has a peak period in the morning. The second function ("2P"), also piecewise affine, is obtained from the first by artificially adding a second peak period in the evening. In both cases, $D(\cdot)$ is increasing and $D(T) = 2016$. For problems $\mathrm{P}^{\text{max}}_{\text{no return}}$ and $\mathrm{P}^{\text{ave}}_{\text{no return}}$, we consider $S \in [100, 250]$ since the number of shuttle trips in every direction is within this range for a typical day.

The numerical experiments have been performed on a Macbook Pro of 2014 with four 2.2 Ghz processors and 16 Gb of ram.

7.7.2 Results

The problems $\mathrm{P}^{\text{max}}_{\text{no return}}$, $\mathrm{P}^{\text{ave}}_{\text{no return}}$, and $\mathrm{P}^{\text{max}}_{\text{return}}$ are solved with algorithms described in this chapter. The results are summarized in the following tables.

Table 7.1 gives the numerical results for problem $\mathrm{P}^{\text{max}}_{\text{no return}}$. The first column provides the name of the instance. The next column is the number of shuttles S in the fleet. The third column provides the parameter ε of the algorithm, which is an *a priori* upper bound on the optimality gap (Corollary 1). The next two columns give respectively the lower bound and the upper bound (value of the feasible solution returned by the algorithm), both expressed in minutes. The next column is the optimality gap. The last column provides the CPU time spent solving the problem.

D	S	ε	LB	UB	gap (%)	CPU (s)
1P	100	10^{-4}	27.2	27.2	0.0	0
1P	150	10^{-4}	18.1	18.1	0.0	0
1P	200	10^{-4}	13.6	13.6	0.0	0
1P	250	10^{-4}	11.0	11.0	0.0	0
2P	100	10^{-4}	27.0	27.0	0.0	0
2P	150	10^{-4}	18.0	18.0	0.0	0
2P	200	10^{-4}	13.5	13.5	0.0	0
2P	250	10^{-4}	10.8	10.8	0.0	0

Table 7.1 – Numerical results for problem $\mathrm{P}^{\text{max}}_{\text{no return}}$

Table 7.2 gives the numerical results for problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{no\ return}}$. The columns are the same as for Table 7.1 except the third one which provides here the parameter M of the algorithm. We know from Theorem 4 that the gap between the upper bound and the lower bound converges asymptotically to 0 when M goes to infinity. We tried $M = 32$ and $M = 128$.

D	S	M	LB	UB	gap (%)	CPU (s)
1P	100	32	17.3	19.2	10.0	34
1P	100	128	18.7	19.2	2.5	1930
1P	200	32	7.7	9.6	19.4	70
1P	200	128	9.1	9.6	5.0	4035
2P	100	32	17.5	19.4	9.9	38
2P	100	128	18.9	19.4	2.5	2387
2P	200	32	7.9	9.7	19.2	76
2P	200	128	9.2	9.7	5.0	4463

Table 7.2 – Numerical results for problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{no\ return}}$

Table 7.3 gives the numerical results for problem $\mathrm{P}^{\mathrm{max}}_{\mathrm{return}}$. The columns are the same as for Table 7.2. Since the computation time was prohibitively long as soon as $S \geq 2$, we made experiments for $S = 1$ only. To get realistic waiting times for the users, we divided the demand functions by 3.5 leading to ("1P*") and ("2P*"). Again, we know from Theorem 5 that for large M, we will be close to the optimal solution and we tried $M = 16$ and $M = 32$.

D	S	M	LB	UB	gap (%)	CPU (s)
1P*	1	16	168.6	214.2	21.3	104
1P*	1	32	184.5	210.8	12.5	1654
2P*	1	16	101.0	131.0	22.9	106
2P*	1	32	106.9	126.3	15.4	1848

Table 7.3 – Numerical results for problem $\mathrm{P}^{\mathrm{max}}_{\mathrm{return}}$

7.7.3 Comments

In Table 7.1, the results for problem $\mathrm{P}^{\mathrm{max}}_{\mathrm{no\ return}}$ are extremely convincing, the optimal solutions were found almost immediately. In Table 7.2, the algorithm for problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{no\ return}}$ was able to find provable good solutions within reasonable computation times. We may note that increasing M after some threshold does not seem to improve the quality of the return solution. This was confirmed by other experiments not shown here. It may indicate that the algorithm could be used efficiently in practice. In Table 7.3, the same holds for $\mathrm{P}^{\mathrm{max}}_{\mathrm{return}}$ once we have accepted to work with one shuttle. Finding an efficient algorithm with at least two shuttles seems to remain a challenging task.

Literature review. In railway transportation, a traditional point of view considers that the demand can be smoothed by offering sufficiently enough departures over a day. Timetabling is then guided by other considerations, such as robustness, maintainability, or rolling stock. For instance, Swiss, Dutch, and German companies usually design periodic timetables, which present many advantages (see Cordone and Redaelli [19] and Kroon et al. [40]. The way to optimize this kind of timetables has been the topic of many researches, initiated by Serafini and Ukovich [62] and by Voorhoeve [69] explicitly in the railway context; see Kroon and Peeters [42], Liebchen [46], Liebchen and Möhring [48], and Nachtigall and Voget [53] for further works. In the context of periodic timetables, a way to adapt the schedules to a demand with strong variations consists in inserting new departures at peak hours and deleting departures when the demand is low, or in changing the size of the trains.

There are timetabling problems with similar features as the ones discussed in this chapter, see Berrena et al. [3], Cacchiani et al. [12, 13], Caprara et al. [16], Cai et al. [15], and Ingolotti et al. [35] for instance. These articles are at a more macroscopic level than what we require in our case.

Conclusion Part

Conclusion

This thesis gives operational algorithms, prospective heuristics and theoretical methods to deal with some problems of rail transportation. Our work was applied to the case of Eurotunnel, which was our partner during the thesis.

The first part is the most operational one. It deals with the problem of computing optimized schedules with different natural objectives for Eurotunnel. The first objective is the maximization of the number of HGV shuttles in the schedule while satisfying the operational constraints and the commercial agreements. Scheduling HGV shuttles is very crucial for Eurotunnel because of the future important increase of freight demand. The second objective is the minimization of the average delay of each train, taking into account stochastic disturbances. Minimizing the delays in the schedules is also a very important issue, in order to limit the propagation of the delays from train to train during the day. We used here the historical data of Eurotunnel to create scenarios of delays as close as possible to what Eurotunnel daily faces. A heuristic for the problem dealing simultaneously with both objectives is also given. These algorithms would be useful for real operational needs of Eurotunnel: they would help engineers to design the everyday schedules. They were enthusiastic about this tool when we presented it. These algorithms would also be useful at a strategic level. By comparing the objective values with different sets of constraints, it could help to give the « price » of every operational and commercial constraint. These metrics are very important for Eurotunnel towards their future investments and commercial negotiations.
Regarding possible improvements upon our work, the number of variables (especially binary ones) and constraints rapidly become very large as the instances grow and the resolution with a solver gets very long. Moreover, the use of commercial solvers is very limiting for big companies since they can be very expensive. Finding efficient algorithms for such problems would be a natural evolution of our work.

The second part is the most prospective one. It deals with a simultaneous scheduling and pricing problem for a transportation company. Each customer wants to purchase a ticket for a fixed trip and has a preferred departure time. He purchases the ticket for the departure that satisfies him the most, if there is one, or does not buy any if no one suits him. The company wants to schedule jointly the departure times and the prices to maximize its revenue. We give algorithms to fix the departure times and the prices, and to give upper bounds on the optimal value of the problem.

This part is very prospective for Eurotunnel. Indeed, their booking system is slightly different than the one considered in this part: the HGVs buy their ticket just before their trip and not in advance. Moreover, the problem is here much simpler than the real one since we only consider one type of trains, without any other traffic. However, Eurotunnel was receptive to the problem we considered and to the methods we gave to compute solutions. In the long run, this method could be interesting for the company to increase jointly its revenue and the customers' satisfaction. As detailed in the chapter, the upper bounds computed by the Lagrangian method are not very convincing and an interesting future work would be to tighten the gap between the upper and lower bounds. A natural evolution to our work would also be to extend the results to more complex networks, several types of trains and to allow evolutive prices over time.

The third part is the most theoretical one. It deals with problems consisting in transporting people or goods from an initial station to another one. We use only a fixed number of shuttles, allowed to perform several rotations in some variants of the problem. The loading process is inspired from Eurotunnel: when the users arrive in the terminal, they enter a queue. This queue closes when all the users who will leave with the next shuttle have arrived and users can enter a new queue only if the previous one is closed. When a queue is closed, the users in that queue can start boarding the shuttle. The objective is to minimize the average waiting time or the maximal waiting time of the passengers. We give efficient algorithms to solve these problems or to compute good approximated solutions. As for the second part, the problem does not exactly correspond to the one faced by Eurotunnel and therefore, the algorithms cannot be directly applied by the company. However, they give results for interesting transportation problems in various domains (rail, chemistry, logistics, ...). Moreover, our algorithms use original methods for such problems, and these methods could be used to adapt the problems to more complex ones in the future.
As already mentioned, an important question remains open: we do not know if there always exists an optimal solution for the problem where the shuttles are allowed to perform several rotations and where the average waiting time is considered (a counterpart to Proposition 5 for problem $\mathrm{P}^{\mathrm{ave}}_{\mathrm{return}}$). If such a result were true, then similar algorithms than the ones for the maximum waiting time would exist. Moreover, the results for this problem dealing with the maximum waiting time are not convincing as long as we use more than one shuttle. Finding an efficient algorithm with at least two shuttles seems to remain a challenging task. Dealing with stochastic demands would also be a natural evolution of the algorithms, though we do not know if all our algorithms would be adaptable. We have started to investigate the question and we got already stuck: as it is often the case, even evaluating $\mathbb{E}[\max(X)]$ where X is a random value is already difficult. Our problem considering the maximum waiting time can be expressed this way and is thus difficult to optimize. Finally, considering other more complex objective functions on the waiting time such as a CVar-type function would also be very interesting to investigate.

Appendix Part

A Security headways computation

This section provides an algorithm to compute the security headway between two trains for Eurotunnel, in order to satisfy some security rules.

A.1 Rules to be satisfied

In order to guarantee the security in the tunnel, some rules have been imposed.

A.1.1 Minimum 500 m-gap

This rule is applicable to every pair of trains of any type. It specifies that in case of an emergency stop of the leading train, the following train has to stop and to leave a gap of at least 500 m with the tail of the leading train.

A.1.2 Gap in case of fire

This rule is only related to HGV shuttles followed or following a train with some passengers. These trains are Eurostars, PAX shuttles, and HGV shuttles (which convey the drivers of all the trucks loaded on the shuttle). Freight trains MA100 and ME120 only convey the driver of the train who can escape easily and these trains are thus not subject to the rule. The rule specifies that in case of a fire on the HGV shuttle, the previous and the following train (if it transports passengers) has to stop and to leave a gap of at least 4 km with that train. This gap corresponds to the distance travelled by the smoke during the time the passengers are evacuated.

For safety reasons, in case of a fire, a shuttle does not have to perform an emergency stop: the shuttle goes to the next fire station (see Section 2.2) if it is reachable, or out of the Tunnel if not.

A.1.3 Buffer blocks

The block safety system (see Section 2.2.5) is used to indicate to the drivers the speed in each block. An operational rule imposes two entire buffer blocks are permanently empty between two trains.

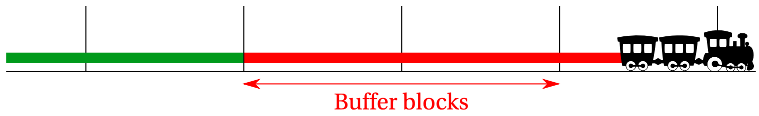

Figure A.1 – Buffer blocks

This rule is illustrated in Figure A.1. Two entire buffer blocks follow dynamically the tail of the train and no train can enter in the red zone.

A.2 Model

This section describes the mathematical model for computing the security headway for each pair of trains according to the previous rules.

For each type of train A, the figures of a typical trip in the Tunnel are known, for both directions. These figures include for 576 distinct positions in the Tunnel:

- The exact position x_i^{A} for all $i \in \{1, \dots, 576\}$
- The slope in the Tunnel p_i^{A} at this position x_i^{A}
- The speed v_i^{A} at this position x_i^{A}
- The cumulative time since the entrance in the Tunnel t_i^{A}.

We denote by ℓ^{A} the length of the train, t_b^{A} the time before the train starts braking once the order in given (a few seconds), t_d the time needed for an incident to be detected and an emergency stop to be activated (all together around 15 s), and γ^{A} the deceleration of the train in case of an emergency brake. We define T_{out}^{A} the travel time in the Tunnel. The length of the Tunnel is denoted by L and the acceleration due to gravity by g. The first fire station is at the first crossover (more or less at one third of the Tunnel) and we denote by s_1 its position. The second one is at the second crossover more or less at two thirds of the Tunnel and we denote by s_2 its position.

The position of the blocks in the Tunnel are known and we define $\mathscr{B} = \{b_1, \dots, b_{N_b}\}$ where b_k is the position of the kth block in the Tunnel and $N_b - 1$ is the number of blocks (b_1 corresponds to the entrance gate and b_{N_b} corresponds to the exit gate).

For each type of train A, we define the piecewise affine functions $p^{A} : [0, T_{\text{out}}^{A}] \to \mathbb{R}$, and $v^{A} : [0, T_{\text{out}}^{A}] \to \mathbb{R}_+$, respectively the slope and the speed. These functions of the time are computed to fit the data. The function $x^{A} : (-\infty, T_{\text{out}}^{A}] \to \{-\infty\} \cup [0, L]$ is defined the same way on $[0, T_{\text{out}}^{A}]$ and is equal to $-\infty$ if $t < 0$. This models the fact that the train is not in the Tunnel when $t < 0$.

We use for each type of train, the function provided by Eurotunnel $f^{A} : \mathbb{R} \to \mathbb{R}_+ \cup \{-\infty, +\infty\}$

modelling the stopping position in case of an emergency brake at t:

$$f^{\mathrm{A}}(t) = \begin{cases} -\infty & \text{if } t < 0 \\ x^{\mathrm{A}}(t + t_b^{\mathrm{A}} + t_d) + \frac{v^{\mathrm{A}}(t+t_b^{\mathrm{A}}+t_d)^2}{2(\gamma - g p^{\mathrm{A}}(t+t_b+t_d))} & \text{if } t + t_b^{\mathrm{A}} + t_d \le T_{\mathrm{out}}^{\mathrm{A}} \\ +\infty & \text{otherwise.} \end{cases}$$

If $t < 0$, the train is not in the Tunnel and can take another track in case of an emergency brake. If $t + t_b^{\mathrm{A}} + t_d > T_{\mathrm{out}}^{\mathrm{A}}$, the train is out of the Tunnel when it starts braking and thus the stopping position is set to $+\infty$.

The pseudo-inverse of $f^{\mathrm{A}}(\cdot)$ is defined by

$$f^{\mathrm{A},-1} : y \in \mathbb{R} \longmapsto \begin{cases} \sup\{t : \; f^{\mathrm{A}}(t) \le y\} & \text{if } y \le f^{\mathrm{A}}(T_{\mathrm{out}}^{\mathrm{A}} - t_b^{\mathrm{A}} - t_d) \\ +\infty & \text{otherwise.} \end{cases}$$

We can check that for all $t \in \mathbb{R}$, $f^{\mathrm{A},-1}(f^{\mathrm{A}}(t)) \ge t$, which will be useful in the remaining of this section. We also define the pseudo-inverse of x^{A} by

$$x^{\mathrm{A},-1} : y \in \mathbb{R} \longmapsto \sup\left\{t \in (-\infty, T_{\mathrm{out}}^{\mathrm{A}}] : \; x^{\mathrm{A}}(t) \le y\right\},$$

and as before, we can check that for all $t \in (-\infty, T_{\mathrm{out}}^{\mathrm{A}}]$, $x^{\mathrm{A},-1}(x^{\mathrm{A}}(t)) \ge t$.

We explain now how to determine the security headway $S^{\mathrm{A,A'}}$ between a train of type A followed by a train of type A'.

For any pair of trains, the security headway $S^{\mathrm{A,A'}}$ has to satisfy the 500 m-gap constraint: for all $t \in [0, T_{\mathrm{out}}^{\mathrm{A}}]$,

$$f^{\mathrm{A}}(t) - f^{\mathrm{A'}}(t - S^{\mathrm{A,A'}} + t_d) \ge 500 + \ell^{\mathrm{A}}$$

which gives

$$S^{\mathrm{A,A'}} \ge t - f^{\mathrm{A'}-1}(f^{\mathrm{A}}(t) - \ell^{\mathrm{A}} - 500) - t_d. \tag{A.1}$$

If the leading train is of type A = HGV, then for all A' $\in$ {PAX, HGV, Eur}, the security headway $S^{\mathrm{A,A'}}$ has to satisfy the gap in case of fire. We define the function $s^{\mathrm{A}} : [0, T_{\mathrm{out}}^{\mathrm{A}}] \to \mathbb{R} \cup \{+\infty\}$ as follows:

$$s^{\mathrm{A}}(t) = \begin{cases} s_1 & \text{if } f^{\mathrm{A}}(t) \le s_1 \\ s_2 & \text{if } s_1 < f^{\mathrm{A}}(t) \le s_2 \\ +\infty & \text{otherwise.} \end{cases}$$

This function models the stopping position of a shuttle for which a fire is detected at time t. If $f^{\mathrm{A}}(t) \le s_1$ (the first safe station is reachable), then s_1 will be the stopping position of the shuttle. For the same reason, if $s_1 < f^{\mathrm{A}}(t) \le s_2$, the shuttle will stop at s_2. If s_2 is not reachable ($f^{\mathrm{A}}(t) > s_2$), the shuttle will drive out of the Tunnel and its stopping position is set to $+\infty$.

The constraint can be modelled as the previous one: for all $t \in [0, T^{\text{A}}_{\text{out}}]$,

$$s^{\text{A}}(t) - f^{\text{A}'}(t - S^{\text{A},\text{A}'} + t_d) \geq 4000 + \ell^{\text{A}}$$

which gives

$$S^{\text{A},\text{A}'} \geq t - f^{\text{A}'-1}(s^{\text{A}}(t) - \ell^{\text{A}} - 4000) - t_d. \tag{A.2}$$

The buffer block constraint has to be satisfied for any pair of trains. We define for all $t \in [0, T^{\text{A}}_{\text{out}}]$, $j^{\text{A}}(t) = \max\{i \in \{1, \ldots, N_b\} : \; b_i \leq x^{\text{A}}(t) - l^{\text{A}}\}$. The constraint can be modelled as follows: for all $t \in [0, T^{\text{A}}_{\text{out}}]$,

$$x^{\text{A}'}(t - S^{\text{A},\text{A}'}) \leq b_{j^{\text{A}}(t)-2},$$

where b_0 and b_{-1} are set to $-\infty$. It gives

$$S^{\text{A},\text{A}'} \geq t - x^{\text{A}'-1}(b_{j^{\text{A}}(t)-2}). \tag{A.3}$$

Combining equations (A.1), (A.2), and (A.3) gives the minimal security value of $S^{\text{A},\text{A}'}$: we compute all the possibles values of right-hand terms of Equations (A.1), (A.2), and (A.3) and we set $S^{\text{A},\text{A}'}$ to be the minimal value found this way.

A.3 Results

We computed the security headways for all the couples of trains. We give here the values of the parameters for all the trains before providing the numerical results.

A.3.1 Instances

As described in the previous section, some parameters depend on the type of train considered. The following Table A.1 provides the values given by Eurotunnel.

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
ℓ	400 m	800 m	800 m	750 m	750 m
t_b	3.0 s	8.3 s	8.3 s	15.8 s	14.4 s
γ	0.70 ms^{-2}	0.75 ms^{-2}	0.75 ms^{-2}	0.58 ms^{-2}	0.69 ms^{-2}
T_{out}	1154 s	1474 s	1474 s	1564 s	1844 s

Table A.1 – Parameters for all types of trains

Moreover, we use $g = 9.81$ ms^{-2}. The length of the Tunnel L is 50,475 km, the first fire station s_1 is 17,160 km and the second one s_2 is 44,575 km.

A.3.2 FOCA

We present in Table A.3 the security headways for direction FOCA (from Folkestone to Calais) for all the pairs of trains. The leading train is indicated in the first column and the following one in the first line.

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0'53"	0'31"	0'31"	0'32"	0'40"
PAX shuttle	6'07"	1'28"	1'28"	1'42"	1'50"
HGV shuttle	6'07"	2'35"	2'35"	1'42"	1'50"
MA100	13'2"	8'14"	8'14"	1'57"	6'43"
ME120	8'13"	3'26"	3'26"	1'06"	1'56"

Table A.2 – Security headways for direction FOCA

However, because of technical reasons, Eurotunnel cannot detect before the entrance gate of the tunnel if a shuttle if of type PAX or HGV. Thus, the gap in case of a fire (see Section A.1.2) also concerns pairs of trains where the leading train is a PAX shuttle. We use then the following headways :

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0'53"	0'31"	0'31"	0'32"	0'40"
PAX shuttle	6'07"	2'35"	2'35"	1'42"	1'50"
HGV shuttle	6'07"	2'35"	2'35"	1'42"	1'50"
MA100	13'02"	8'14"	8'14"	1'57"	6'43"
ME120	8'13"	3'26"	3'26"	1'06"	1'56"

Table A.3 – Security headways for direction FOCA used by Eurotunnel

A.3.3 CAFO

We present in Table A.5 the security headways for direction CAFO (from Calais to Folkestone) for all the pairs of trains. The leading train is indicated in the first column and the following one in the first line.

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0'53"	0'48"	0'48"	0'48"	0'48"
PAX shuttle	6'22"	1'23"	1'23"	1'21"	1'21"
HGV shuttle	6'22"	2'49"	2'49"	1'21"	1'21"
MA100	12'21"	7'34"	7'34"	1'30"	5'59"
ME120	7'48"	3'02"	3'02"	1'15	1'18

Table A.4 – Security headways for direction CAFO used

As for direction FOCA, technical reasons lead to the following headways used by Eurotunnel:

	Eurostar	PAX shuttle	HGV shuttle	MA100	ME120
Eurostar	0’53”	0’48”	0’48”	0’48”	0’48”
PAX shuttle	6’22”	2’49”	2’49”	1’21”	1’21”
HGV shuttle	6’22”	2’49”	2’49”	1’21”	1’21”
MA100	12’21”	7’34”	7’34”	1’30”	5’59”
ME120	7’48”	3’02”	3’02”	1’15”	1’18”

Table A.5 – Security headways for direction CAFO used by Eurotunnel

Bibliography

[1] Arjang Assad. Models for rail transportation. *Transportation Research Part A: General*, 14(3):205–220, 1980.

[2] Lineu Barbosa and Moshe Friedman. Deterministic inventory lot size models – a general root law. *Management Science*, 24(8):819–826, 1978.

[3] Eva Barrena, David Canca, Leandro Coelho, and Gilbert Laporte. Exact formulations and algorithm for the train timetabling problem with dynamic demand. *Computers & Operations Research*, 44:66–74, 2014.

[4] Moshe Ben-Akiva and Michel Bierlaire. Discrete choice methods and their applications to short term travel decisions. *Handbook of Transportation Science*, 23:5–33, 1999.

[5] Moshe Ben-Akiva and Steven Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

[6] Dimitri Bertsekas. *Nonlinear Programming*. Athena scientific Belmont, 1999.

[7] Dimitris Bertsimas. Joint Network Pricing and Resource Allocation. Technical report, 2002.

[8] Juan José Miranda Bront, Isabel Méndez-Díaz, and Gustavo Vulcano. A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3):769–784, 2009.

[9] Luce Brotcorne, Fabien Cirinei, Patrice Marcotte, and Gilles Savard. An exact algorithm for the network pricing problem. *Discrete Optimization*, 8(2):246–258, 2011.

[10] Luce Brotcorne, Martine Labbé, Patrice Marcotte, and Gilles Savard. Joint design and pricing on a network. *Operations Research, volume=56, number=5, pages=1104–1115, year=2008, publisher=INFORMS.*

[11] Michael R Bussieck, Thomas Winter, and Uwe T Zimmermann. Discrete optimization in public rail transport. *Mathematical Programming*, 79(1-3):415–444, 1997.

[12] Valentina Cacchiani, Alberto Caprara, and Paolo Toth. A column generation approach to train timetabling on a corridor. *4OR: A Quarterly Journal of Operations Research*, 6(2):125–142, 2008.

[13] Valentina Cacchiani, Alberto Caprara, and Paolo Toth. Non-cyclic train timetabling and comparability graphs. *Operations Research Letters*, 38(3):179–184, 2010.

[14] Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737, 2012.

[15] Xiaoqiang Cai, C.J. Goh, and Alistair Mees. Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transactions*, 30(5):481–493, 1998.

[16] Alberto Caprara, Matteo Fischetti, and Paolo Toth. Modeling and solving the train timetabling problem. *Operations Research*, 50(5):851–861, 2002.

[17] Malachy Carey. Ex ante heuristic measures of schedule reliability. *Transportation Research Part B: Methodological*, 33(7):473–494, 1999.

[18] Jean-Francois Cordeau, Paolo Toth, and Daniele Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32(4):380–404, 1998.

[19] Roberto Cordone and Francesco Redaelli. Optimizing the demand captured by a railway system with a regular timetable. *Transportation Research Part B: Methodological*, 45(2):430–446, 2011.

[20] Robert Corless, Gaston Gonnet, David Hare, David Jeffrey, and Donald Knuth. On the Lambert W Function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.

[21] A.F. De Kort. Advanced railway planning using (max,+) algebra. *WIT Transactions on The Built Environment*, 50, 2000.

[22] Walter Erwin Diewert. Alternative characterizations of six kinds of quasiconcavity in the nondifferentiable case with applications to nonsmooth programming. In S. Schaible and W.T Ziemba, editors, *Generalized Concavity in Optimization and Economics*, pages 51–93. Academic Press, New York, 1981.

[23] Daniel Dooly, Sally Goldman, and Stephen Scott. TCP dynamic acknowledgment delay (extended abstract): theory and practice. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 389–398. ACM, 1998.

[24] Alexander Erdelyi and Huseyin Topaloglu. Using decomposition methods to solve pricing problems in network revenue management. *Journal of Revenue and Pricing Management*, 10(4):325–343, 2011.

[25] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2005.

[26] Matteo Fischetti, Domenico Salvagnin, and Arrigo Zanette. Fast approaches to improve the robustness of a railway timetable. *Transportation Science*, 43(3):321–335, 2009.

[27] Rob Goverde. The max-plus algebra approach to railway timetable design. *WIT Transactions on The Built Environment*, 37, 1998.

[28] Rob Goverde. *Punctuality of Railway Operations and Timetable Stability Analysis*. Trail, 2005.

[29] Valérie Guihaire and Jin-Kao Hao. Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251–1273, 2008.

[30] Susan Hallowell and Patrick Harker. Predicting on-time performance in scheduled railroad operations: methodology and application to train scheduling. *Transportation Research Part A: Policy and Practice*, 32(4):279–295, 1998.

[31] Pratt Hetrakul and Cinzia Cirillo. A latent class choice based model system for railway optimal pricing and seat allocation. *Transportation Research Part E: Logistics and Transportation Review*, 61:68–83, 2014.

[32] Andrew Higgins and Erhan Kozan. Modeling train delays in urban networks. *Transportation Science*, 32(4):346–357, 1998.

[33] Dennis Huisman, Leo Kroon, Ramon Lentink, and Michiel Vromans. Operations Research in Passenger Railway Transportation. *Statistica Neerlandica*, 59(4):467–497, 2005.

[34] Tijs Huisman and Richard J Boucherie. Running times on railway sections with heterogeneous train traffic. *Transportation Research Part B: Methodological*, 35(3):271–292, 2001.

[35] Laura Ingolotti, Antonio Lova, Federico Barber, Pilar Tormos, Miguel Angel Salido, and Montserrat Abril. New heuristics to solve the CSOP railway timetabling problem. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 400–409. Springer, 2006.

[36] Alan King and Tyrrell Rockafellar. Asymptotic theory for solutions in statistical estimation and stochastic programming. *Mathematics of Operations Research*, 18(1):148–162, 1993.

[37] Anton Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[38] Frank Geraets Leo Kroon, Dorothea Wagner, Frank Geraets, and Christos Zaroliagis. *Algorithmic Methods for Railway Optimization*. Springer, 2006.

[39] Leo Kroon, Rommert Dekker, and Michiel Vromans. Cyclic railway timetabling: a stochastic optimization approach. In *Algorithmic Methods for Railway Optimization*, pages 41–66. Springer, 2007.

[40] Leo Kroon, Dennis Huisman, Erwin Abbink, Pieter-Jan Fioole, Matteo Fischetti, Gábor Maróti, Alexander Schrijver, Adri Steenbeek, and Roelof Ybema. The new dutch timetable: The OR revolution. *Interfaces*, 39(1):6–17, 2009.

[41] Leo Kroon, Gábor Maróti, Mathijn Retel Helmrich, Michiel Vromans, and Rommert Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553–570, 2008.

[42] Leo Kroon and Leon Peeters. A variable trip time model for cyclic railway timetabling. *Transportation Science*, 37(2):198–212, 2003.

[43] Richard Larson and Amedeo Odoni. *Urban Operations Research*. 1981.

[44] Phillip Lederer. A competitive network design problem with pricing. *Transportation Science*, 27(1):25–38, 1993.

[45] Vassilissa Lehoux-Lebacque, Nadia Brauner, Gerd Finke, and Christophe Rapine. Scheduling chemical experiments. In *37th International Conference on Computers and Industrial Engineering,(CIE37)*, 2007.

[46] Christian Liebchen. Finding short integral cycle bases for cyclic timetabling. In *European Symposium on Algorithms*, pages 715–726. Springer, 2003.

[47] Christian Liebchen. Symmetry for periodic railway timetables. *Electronic Notes in Theoretical Computer Science*, 92:34–51, 2004.

[48] Christian Liebchen and Rolf Möhring. A case study in periodic timetabling. *Electronic Notes in Theoretical Computer Science*, 66(6):18–31, 2002.

[49] Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

[50] Karl Nachtigall. *A branch and cut approach for periodic network programming*. Inst. für Mathematik, 1994.

[51] Karl Nachtigall. Periodic network optimization with different arc frequencies. *Discrete Applied Mathematics*, 69(1-2):1–17, 1996.

[52] Karl Nachtigall. *Periodic network optimization and fixed interval timetables*. 1999.

[53] Karl Nachtigall and Stefan Voget. A genetic algorithm approach to periodic railway synchronization. *Computers & Operations Research*, 23(5):453–463, 1996.

[54] Michiel Odijk. A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B: Methodological*, 30(6):455–464, 1996.

[55] Michiel Odijk. *Railway timetable generation.* Delft University Press, 1997.

[56] Meritxell Pacheco, Shadi Sharif Azadeh, and Michel Bierlaire. A new mathematical representation of demand using choice-based optimization method. In *16th Swiss Transport Research Conference*, number EPFL-CONF-218673, 2016.

[57] E. Petersen and Taylor A. A structured model for rail line simulation and optimization. *Transportation Science*, 16(2):192–206, 1982.

[58] Tomáš Robenek, Shadi Sharif Azadeh, Yousef Maknoon, and Michel Bierlaire. Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables. *Transportation Research Part C: Emerging Technologies*, 75:228–253, 2017.

[59] Paat Rusmevichientong, David Shmoys, Chaoxu Tong, and Huseyin Topaloglu. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23(11):2023–2039, 2014.

[60] Paat Rusmevichientong, David Shmoys, and Huseyin Topaloglu. Assortment optimization with mixtures of logits. Technical report, School of IEOR, Cornell University, 2010.

[61] Alexander Schrijver and Adri Steenbeek. Timetable construction for railned. Technical report, CWI, Amsterdam, The Netherlands, 1994.

[62] Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, 1989.

[63] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory.* SIAM, 2009.

[64] Alexander Shapiro and Tito Homem-de Mello. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization*, 11(1):70–86, 2000.

[65] Kalyan Talluri and Garrett Van Ryzin. *The theory and practice of revenue management*, volume 68. Springer Science & Business Media, 2006.

[66] Kenneth Train. *Discrete choice methods with simulation.* Cambridge university press, 2003.

[67] Amos Tversky. Elimination by aspects: A theory of choice. *Psychological Review*, 79(4):281, 1972.

[68] Ann Van Ackere. The principal/agent paradigm: Its relevance to various functional fields. *European Journal of Operational Research*, 70(1):83–103, 1993.

[69] Marc Voorhoeve. Rail scheduling with discrete sets. *Unpublished report, Eindhoven University of Technology, The Netherlands*, 1993.

[70] Michiel Vromans, Rommert Dekker, and Leo Kroon. Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2):647–665, 2006.

[71] Michiel Vromans, Rommert Dekker, and Leo Kroon. *Reliability and heterogeneity of railway services*, volume 172. Elsevier, 2006.

[72] Jianxin Yuan. Stochastic modelling of train delays and delay propagation in stations. 2006.

[73] Xuesong Zhou and Ming Zhong. Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research*, 167(3):752–771, 2005.