



The Long-Haul Parcel Transportation Problem : Path-based Models and Divide-and-Conquer Algorithms

Camille Gras

► To cite this version:

Camille Gras. The Long-Haul Parcel Transportation Problem : Path-based Models and Divide-and-Conquer Algorithms. Software Engineering [cs.SE]. Université Grenoble Alpes [2020-..], 2021. English. NNT : 2021GRALI073 . tel-03462717

HAL Id: tel-03462717

<https://theses.hal.science/tel-03462717>

Submitted on 2 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : **GI : Génie Industriel : conception et production**

Arrêté ministériel : 25 mai 2016

Présentée par

Camille GRAS

Thèse dirigée par **Van-Dat CUNG**, Professeur, Grenoble INP-UGA,
et codirigée par **Alantha NEWMAN**, Chargée de Recherche, CNRS,
Nathalie HERR, Ingénieur docteur, Probayes
et **Nicolas TEYPAZ**, Ingénieur docteur, Probayes

préparée au sein du **Laboratoire des Sciences pour la Conception,
l'Optimisation et la Production de Grenoble**
dans l'**École Doctorale I-MEP2 - Ingénierie - Matériaux, Mécanique,
Environnement, Énergétique, Procédés, Production**

Problème de transport long-courrier de colis : modèles orientés chemins et algorithmes diviser-pour-régner

The Long-Haul Parcel Transportation Problem: Path-based Models and Divide-and-Conquer Algorithms

Thèse soutenue publiquement le **14 octobre 2021**
devant le jury composé de :

Monsieur Van-Dat CUNG

PROFESSEUR, Grenoble INP-UGA, Directeur de thèse

Madame Marie-Ange MANIER

MAITRE DE CONFERENCE, Université de Technologie de Belfort Montbeliard,
Rapporteuse

Monsieur Gautier STAUFFER

CHERCHEUR HDR, Kedge Business School, Rapporteur

Monsieur Axel PARMENTIER

CHARGE DE RECHERCHE, Ecole des Ponts et Chaussées, Examineur

Madame Marie-Laure ESPINOUSE

PROFESSEUR, Université Grenoble-Alpes, Examinatrice

Monsieur Khaled HADJ HAMOU

PROFESSEUR, INSA Lyon, Président

Madame Alantha NEWMAN

CHARGE DE RECHERCHE, CNRS, Encadrante

Madame Nathalie HERR

INGENIEUR DOCTEUR, Probayes, Encadrante

Monsieur Nicolas TEYPAZ

INGENIEUR DOCTEUR, Probayes, Encadrant



Remerciements/Acknowledgments

Mes premiers remerciements vont à Probayes et à mes encadrants, Nathalie Herr et Nicolas Teypez. Merci de m'avoir présenté ce sujet de stage et d'avoir porté ce sujet de thèse. Merci à tous les collègues de Probayes pour les moments de convivialité partagés et notamment les repas au LRP. Un merci tout particulier à Hugues, Ouadie, Joseph, David, Ann-Sophie et Sylvain qui ont été là pour me conseiller dans les moments difficiles.

Je tiens à remercier l'Association Nationale Recherche Technologie qui a soutenu financièrement mon travail dans le cadre de la bourse CIFRE n°2019/0351.

Ensuite, je tiens à remercier mes rapporteurs, Marie-Ange Manier et Gautier Stauffer pour leur lecture attentive du manuscrit et leurs précieuses remarques. Ainsi que mes examinateurs Axel Parmentier, Marie-Laure Espinouse et Khaled Hadj Hamou qui me font l'honneur d'être dans mon jury. Sans oublier mes deux tuteurs académiques, Vandat Cung et Alantha Newman, qui ont su m'aider à grandir en tant que chercheuse. Je remercie tous les membres du laboratoire G-SCOP pour leur accueil. Merci aussi aux amis, Cléopée, Luc, Lucie, Cléa, Florian, Pascal, Asiye, Bilge, Maud, Valentin, Astor et Thomas pour les bons moments passés ensemble.

Je tiens également à remercier mes parents Sylvie et Hervé, et mes soeurs, Delphine et Héloïse, pour leur précieux soutien et leurs encouragements. Je remercie finalement Benjamin qui m'a accompagnée au quotidien pendant ces trois années. Je le remercie de son amour, de sa confiance et de son soutien inconditionnel.

Abstract/Résumé

Abstract

With the advent of e-commerce, many studies have been carried out on urban logistics and last-mile delivery. In this work, we optimize another stage of parcel delivery, the long-haul parcel transportation. The long-haul transportation is from the collection sorting centers to delivery depots. We do not consider how the parcels are brought from their departure post office to their collection sorting center, neither how they are transported to the post offices and then to individuals. The Long-Haul Parcel Transportation Problem (LHPTP), which we formally define, is a Service Network Design with Asset Management problem which integrates the sorting operation allowing a better consolidation of parcels in containers. The LHPTP is a tactical design optimization problem which consists in defining an annual transportation plan composed of fixed links, based on mid-term volume forecasts, while minimizing the total cost. This cost is composed of the logistics cost and the transportation cost. The parcel transportation is made with two types of vehicles (trucks with one or two containers) which are balanced over the network on a daily basis with the management of empty containers. The transportation is optimized over a two-level hybrid hub-and-spoke network at the scale of a country. Indeed, this industrial problem originates from a postal company and their datasets permit to address realistic size data (around 225 sites with 2500 demands). The complexity is increased by the fact that a single demand (origin, destination, number of parcels) can be routed over multiple paths simultaneously. Thus the number of possible transportation plans explodes.

We propose a path-based Mixed Integer Linear Program (MILP) for the LHPTP and two divide-and-conquer algorithms exploiting this model to create better transportation plans. The first algorithm, the k -Clusters Algorithm, optimizes the LHPTP over a set of sites which are clustered beforehand. We test classical clustering techniques (spectral clustering, hierarchical clustering, k-means and random) using appropriate similarity functions (demand-based and distance-based) in order to see if it impacts the results. The original problem is divided into intercluster and intracluster subproblems which are solved with the MILP. Then the solutions of the subproblems are merged. The results obtained are compared to those obtained with a direct use of the MILP without clustering. It shows that certain properties of the clustering, such as obeying the hierarchy of the sites in the current postal network, have the most impact on the results.

Thus we design a second algorithm, the Hierarchical Algorithm with Aggregation

of Demands which exploits the two-level structure of the network. Its performance is related to the value of a truck filling rate threshold. The demands above this threshold can be routed directly while the ones below this threshold have to follow the hierarchical structure of the network. The routing of the two types of demands is optimized, first separately and then together in a multi-step process in which the subproblems are solved with the MILP. Various threshold values are tested to find out which one is the best, in terms of solution quality obtained and computational time. These tests show that the better filling rate do not result in the cheaper transportation plan in our case study. Moreover, the Hierarchical Algorithm allows to have clearly better transportation plans than the ones applied on the ground, the ones obtained via a direct use of the MILP and even the ones obtained with the k -Clusters Algorithm. Finally we implement these algorithms and present computational results. This shows that the divide-and-conquer paradigm is effective on service network design when addressing a large-scale industrial problem.

Keywords: Operational Research, Long-Haul Parcel Transportation, Network Design, Divide-and-Conquer, Tactical planning

Résumé

Avec l'essor du e-commerce, de nombreuses études ont été menées sur la logistique urbaine et la livraison du dernier kilomètre. Nous optimisons ici une autre étape de la livraison des colis : le transport long-courrier. Il a lieu entre les centres de tri de collecte et les dépôts de livraison. Ni la manière dont les colis sont acheminés de leur bureau de poste de départ à leur centre de tri de collecte, ni comment ils sont transportés vers les bureaux de poste puis aux particuliers ne sont pris en considération. Le problème du transport long-courrier de colis (PTLCC), défini formellement, est un problème de conception de réseau de services avec gestion des actifs. Il intègre l'opération de tri permettant une meilleure mutualisation des colis dans les conteneurs. C'est un problème tactique d'optimisation qui consiste à définir un plan de transport annuel composé de liaisons fixes, basé sur des prévisions de volumes à moyen terme, dont on minimise le coût total. Ce coût est composé du coût logistique et du coût de transport. Le transport de colis se fait avec deux types de véhicules (camions à un ou deux conteneurs) qui sont équilibrés chaque jour sur le réseau grâce à la gestion des conteneurs vides. Le transport est optimisé sur un réseau hybride hub-and-spoke biniveau à l'échelle d'un pays. En effet, ce problème industriel provient d'une entreprise postale et leurs ensembles de données sont de taille réaliste (environ 225 sites avec 2500 demandes). Une même demande (origine, destination, nombre de colis) peut être acheminée sur plusieurs chemins simultanément ce qui augmente la complexité du problème. Ainsi, le nombre de plans de transport possibles explose.

Nous proposons un programme linéaire mixte (PLM) orienté chemin pour le PTLCC et deux algorithmes diviser-pour-régner exploitant ce modèle pour créer de meilleurs plans de transport. Le premier algorithme, l'algorithme k -Clusters, optimise le PTLCC après avoir regroupé les sites du réseau en clusters. Nous testons des techniques classiques de clustering (clustering spectral, clustering hiérarchique, k -means et aléatoire) en utilisant des fonctions de similarité appropriées (basées sur les demandes et sur les distances) pour étudier l'impact sur les résultats. Le problème d'origine est divisé en sous-problèmes intercluster et intracluster résolus avec le PLM. Les solutions des sous-problèmes sont ensuite fusionnées. Les résultats obtenus sont comparés à ceux obtenus avec une utilisation directe du PLM sans clustering. Ces tests montrent que le respect de la hiérarchisation des sites du réseau postal est la propriété qui a le plus d'impact sur les résultats.

Ainsi, nous concevons un deuxième algorithme, l'algorithme hiérarchique avec agrégation de demandes qui exploite la structure à deux niveaux du réseau. Ses performances sont liées à un seuil du taux de remplissage des camions. Les demandes au-dessus de ce seuil peuvent être acheminées directement. Celles en dessous de ce seuil doivent suivre la structure hiérarchique du réseau. L'acheminement des deux types de demandes est optimisé, d'abord séparément puis conjointement via plusieurs étapes dans lesquelles les sous-problèmes sont résolus avec le PLM. Différents seuils sont testés pour déterminer lequel donne les meilleures solutions et temps de calcul. Ces tests montrent qu'un meilleur taux de remplissage n'aboutit pas à un plan de transport moins cher dans notre cas. De plus, l'algorithme hiérarchique permet d'avoir des plans de transport nettement meilleurs que ceux appliqués sur le terrain, ceux obtenus via une utilisation directe du PLM et même

ceux obtenus avec l'algorithme k -Clusters. Enfin, nous implémentons ces algorithmes et présentons les résultats numériques. Cela montre que le paradigme diviser-pour-régner est efficace pour la conception de réseau de services lorsqu'il s'applique à un problème industriel de grande taille.

Mots-clés : Recherche Opérationnelle, Transport long-courrier de colis, Conception de réseau, Diviser-pour-régner, Planification tactique

Contents

Remerciements/Acknowledgments	iii
Abstract/Résumé	v
Contents	ix
Introduction	1
1 Industrial Context	1
2 General Problem Description	3
3 Scope of the Study	4
4 Algorithmic Approaches	5
5 Manuscript Organization	6
1 Background and Previous Approaches	9
1 The Wide Spectrum of Parcel Transportation Problems	12
1.1 Transportation Mode	14
1.2 Transportation Network	15
1.3 Sorting and Other Logistical Aspects of the Parcel Transportation . .	16
1.4 Optimization Objectives	17
1.5 Overview of Computational Approaches	18
1.6 Synthesis	18
2 Network Design in Freight Transportation Problems	19
2.1 Point-to-Point Delivery Problems	19
2.2 Service Network Design Problems	20
2.3 Service Network Design with Asset Management	22
2.4 Trucks with Twin-Trailers	23
2.5 Synthesis	24
3 Design of Hub-and-Spoke Networks for Freight Transportation	25
3.1 Hub-and-Spoke Network	25
3.2 Hub-and-Spoke Network Design	27
3.3 Hub Location Problem	28
3.4 Synthesis	29

4	Linear Programming and MILP Decomposition	30
4.1	Mixed Integer Linear Programs	30
4.2	Classical Optimization Techniques	31
4.3	MILP Decomposition via Divide-and-Conquer Paradigm	32
5	Conclusion	33
2	Problem Description	35
1	Application Framework	36
1.1	Long-Haul Transportation	36
1.2	Parcels and Demands	37
1.3	Sites	37
1.4	Vehicles and Empty Repositioning	39
1.5	Sorting and Consolidation	39
1.6	Operational Paths	41
2	Optimization Problem	44
2.1	Input: Parameters of the Optimization Problem	44
2.2	Objective	45
2.3	Constraints	45
2.4	Output: Transportation Plan	46
2.5	Key Performance Indicators	47
3	Formal Definition of the LHPTP	48
3.1	Network Provided	48
3.2	Graph Definition	49
3.3	Path-Based MILP Formulation	50
4	Conclusion	51
3	Case Study Analysis	53
1	Datasets	54
1.1	Content of the Datasets	55
1.2	Configuration of Sites	56
1.3	Demand Profile	57
1.4	Distances in the Network	59
2	Optimization Strategies for Using the Network	61
2.1	Computational Aspects	62
2.2	Current Operational Strategy	64
2.3	Addition of New Paths to the Current Operational Strategy	65
2.4	Impact of the Soft Business Constraints on the Size of the MILP For- mulation	67
2.5	Synthesis	69
3	Exact Approach: Direct Solving with MILP	69
3.1	Application of the MILP on the LHPTP	70
3.2	Limits of LHPTP-MILP on our Case Study	73
4	Conclusion	74

4	Cluster-based Approach for Solving the LHPTP	77
1	Background and Previous Work	79
1.1	Clustering Algorithms	79
1.2	Cluster-Based Approaches in Transportation Optimization	82
2	k -Clusters Algorithm	83
2.1	Approach to Solving the LHPTP	83
2.2	Detailed Steps of the k -Clusters Algorithm	85
3	Applying the Clustering Methods on our Data	87
3.1	Input Data Provided to the Clustering Algorithms	88
3.2	Input Data for Each Clustering Algorithm	90
3.3	Output Clusterings	91
3.4	Confirmation of the Catchment Areas Utility	95
3.5	Structural Properties of the Data Revealed via Clustering	96
3.6	Synthesis	98
4	Simulations and Results	98
4.1	Simulation Parameters	98
4.2	Evaluation of Results	99
4.3	Synthesis	102
5	Conclusion	103
5	Hierarchical Approach: Algorithm with Aggregation of Demands	105
1	Background and Previous Works	107
2	A Hierarchical Algorithm with Aggregation of Demands	107
2.1	Approach to Solving the LHPTP	108
2.2	Detailed Steps of the Algorithm	110
3	Simulations and Results	114
3.1	Simulation Parameters	114
3.2	Evaluation of Results	115
3.3	Synthesis	117
3.4	Comparison of the k -Clusters Algorithm and the Hierarchical Algorithm	118
4	Conclusion	120
	Conclusion and Perspectives	123
1	General Synthesis	123
2	Perspectives	125
	Glossary	129
	References	135
	Résumé long	149

Introduction

1 Industrial Context

With the growing trend of e-commerce over the past two decades [MSB⁺14], parcel delivery has become an important focus of postal companies. In 2020, over two billion people purchased goods or services online [Sta21]¹. This growth trend is even more extreme due to the pandemic, pushing people to purchase more items online and generating an even greater need for efficient parcel delivery. This provides motivation for parcel delivery companies to constantly adapt and optimize their transportation networks. Optimizing transport and logistics operations for postal delivery is therefore crucial from both an economic and a quality of service point of view.

This CIFRE PhD is done at Probayes² which is a French company of 55 employees with a very strong R&D-oriented team of Data Scientists, with in-depth expertise in Data Mining, Machine Learning and Operations Research. In 2016, Le Groupe La Poste³ bought the company in order to build a center for artificial intelligence research to meet the need for internal transformation and optimization but also to develop new services for their customers. This led to the creation in 2017 of the project *Optimnet*⁴, which aims at optimizing parcel transportation. *Optimnet* is “a decision support solution, based on combinatorial optimization and predictive analysis techniques to provide customized and optimized transportation plans” to the postal company which uses it⁵. The tool *Optimnet* aims to help “transform the postal network tackled, to be more responsive in times of high activity, to optimize delivery times, to reduce costs while preserving the quality of service and to reduce the CO₂ emissions of parcel delivery”. In this thesis, we do not consider all of these objectives as we focus on a specific case study from a real customer of Probayes. The datasets provided by this customer are described in Chapter 3.

In this manuscript, we deal with the optimization of both parcels sent by companies and parcels sent by individuals. Thus, there are parcels from all the origin sites to all the destination sites. The parcels go through the process illustrated in Figure 1. Note that the process represented on the picture is the general process which as all processes can have exceptions. In the classical process, parcels are brought to the post offices by senders

¹<https://www.statista.com/topics/871/online-shopping/>

²<https://www.probayes.com/>

³<https://www.groupelaposte.com/en>

⁴<https://www.probayes.com/logistique-et-distribution/projet-optimnet/>

⁵<https://fr.calameo.com/read/006680797c45604ae82b9?page=1>

or postmen. Then in the collection stage they are transported to a fixed **sorting center** according to their departure post office. Next, the stage of long-haul transportation – that we optimize in this PhD thesis – brings the parcels to sites called **delivery depots**, potentially via an additional sorting center. The transportation stage from the delivery depot to the post office is the distribution. It can use an intermediate site. Finally, the delivery occurs from the post office to the recipient’s house.

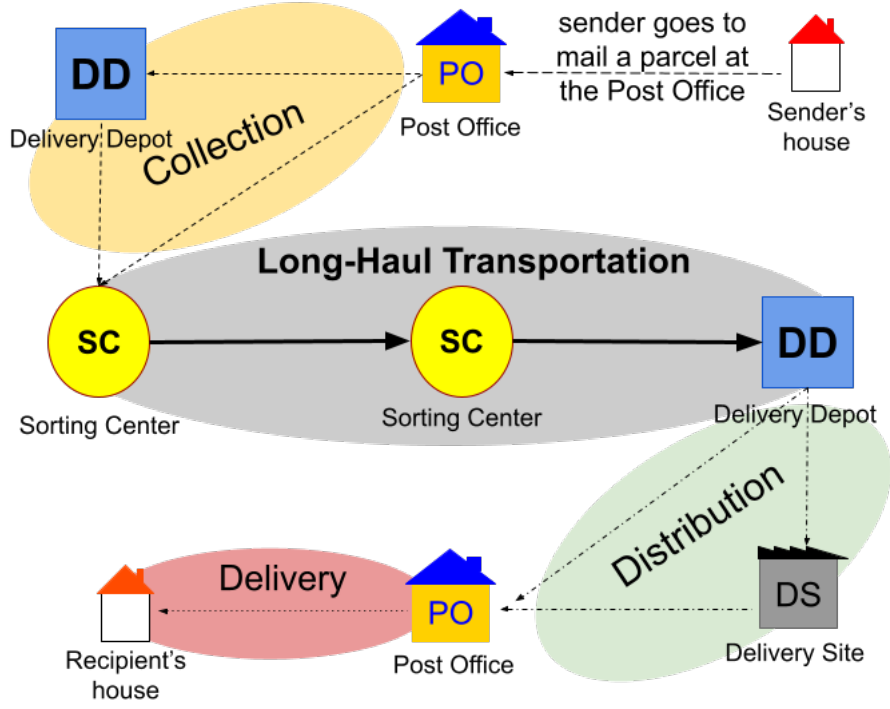


Figure 1: Path of a parcel from sender to recipient.

In the collection phase, parcels are gathered in a fixed Sorting Center associated with their departure Post Office (with some steps that we do not address here). The long-haul transportation is from the collection sorting center to the delivery depot associated with the final destination (more accurately, the final destination is associated with a post office which is associated with a delivery depot). In the complete transportation process presented in Figure 1, the long-haul transportation is from a sorting center to a sorting center to a delivery depot. But in this manuscript, we offer other options. The parcel can go directly from the collection sorting center to the delivery depot (when there are many parcels which require this). Or the parcel can go from the collection sorting center to two consecutive sorting centers to the delivery depot (when there are few parcels from this collection sorting center to this delivery depot). What happens precisely in the different sites is explained by Launay [Lau18].

2 General Problem Description

The objective of this work is to design an optimized transportation service network on a tactical level. There are three planning levels in transportation optimization [Cra00, MW84]. The **strategic** level deals with long-term optimization (acquisitions of vehicles, construction of roads or buildings etc.). The **tactical** level treats medium-term optimization (effective use of the network). Finally, the **operational** level concerns the short-term optimization (schedules for services, crews, maintenance activities etc.). We optimize parcel transportation on the tactical level for given demands which is medium to long term optimization. In this manuscript, we study this challenging industrial problem and design algorithms that can tackle it efficiently but could also be applied to other similar freight transportation problems in a bottom-up generalization approach.

The postal company for which we optimize the network is providing demand forecasts which represent an average day associated to average volumes to be distributed for six different years. Thus the demand forecast is not part of the optimization process we treat.

Our goal is to optimize the transportation of parcels from their collection sorting centers to delivery depots. We optimize only the long-haul transportation stage [Cra03, vHMS19] of parcel transportation, thus we refer to this problem as the **Long-Haul Parcel Transportation Problem** (LHPTP). We now give a comprehensive description of the LHPTP while the formal definition is given in Section 1 of Chapter 2. Assuming that a daily demand forecast is provided, in which parcels are sent to and from a predetermined set of sites within France, our objective is to find a daily transportation plan for all the parcels (resulting in their distribution and delivery) that minimizes transportation and logistics costs. In the LHPTP, all the parcels have to be delivered in one or two days. The capacity of the vehicles used has to be respected and these vehicles must be returned to the sorting centers in a manner that accommodates the transportation plan of the next day. This is called vehicles balancing.

The costs encountered in long-haul parcel transportation are transportation costs and costs of logistics operations. In this problem, transportation demands are expressed by the triple (origin, destination, number of parcels). The road network is composed of possible origin, delivery and intermediate sites and links between them is known and given as part of the input. Our goal is to optimize the service network on the given road network while considering simultaneously two aspects:

- We select which connection between sites – truck routes – we activate on the road network. We also have to manage trucks on these connections: We choose the number of trucks for each route assuming that trucks are available on each site when needed.
- We fix parcel paths and their assignment to trucks. Note that the parcels are not interchangeable (unlike, for instance, a commodity which can come from various origin sites).

A key aspect of our problem is that due to the allowed sorting operation (which induces a fixed cost), a parcel need not be routed in a single truck. In fact, a parcel can be assigned to at most three different trucks over the course of its transportation path and can cross at most two intermediate sites on its path from origin to destination. Indeed, to devise a more efficient and cost-effective transportation plan, parcels can be sorted according to their destinations and assigned to other trucks in the sorting centers. Moreover, parcels belonging to the same demand need not be routed over the same path. In other words, a demand may be routed over several paths, which is called disaggregate shipping [LMS90]. At the same time, parcels from different demands can also be gathered or **consolidated** in the same truck. For instance, if two demands can share a common section in their respective paths and each of these demands has a volume of 50% of the vehicle capacity, then it is possible to run three vehicles rather than four on the common section of the route. One of the three vehicles contains the mix of the two demands (50% + 50%), and the two other vehicles, which do not contain mixed demands, do not need a sorting operation at a site and can take different routes. The usage of multiple paths and consolidation are enabled by sorting operations and results in better solutions, but it adds to the computational difficulty of the problem.

3 Scope of the Study

Here, we aim to clarify the scope of the LHPTP. It is a deterministic problem, as the demand forecast is part of the input. Therefore we will not discuss dynamic variants of this problem. As we said earlier, we consider only the long-haul transportation of parcels. We do not optimize distribution and delivery as depicted on Figure 1 which is usually called last-mile delivery, so we will not talk about urban logistics. Nor do we optimize pick-up (first-mile) delivery. In this work we address road transportation optimization. However, as some ideas from air transportation or multimodal transportation can be useful, they will be discussed in Chapter 1.

We consider both trucks and parcels point of views, therefore it is not a classical flow optimization problem as we deal with two interdependent flows. Moreover parcel delivery deals with point-to-point demands. Even if some analogy can be done with the supply chain optimization, this is a big difference between parcel transportation and supply chain optimization, and thus we will not talk about it in this manuscript. Furthermore, the supply chain optimization takes into account the operational level and its interaction with the tactical level which is not the same in our case. The demand we treat are from a zip code to another zip code; it is not like in freight transportation in which products can come from different plants or warehouses to serve one customer. The parcel flow is splittable: One demand from a site to another can use simultaneously multiple trucks which can use multiple routes.

In this manuscript, the objective of the problem is to minimize the costs for the postal company. We do not consider here the quality of service (next day delivery) in the objective, but the delivery is guaranteed in one or two days as we only model these

possibilities. Thus we do not do what is called multi-objective optimization.

As we place ourselves at the tactical level, we do not study the planning of the sorting operations inside the sites; rather, we treat the sorting as a blackbox operation as it takes place at the operational level. For more details on the optimization of parcel sorting on sorting center, one could refer to Clausen et al. [CDB⁺15] or Duman [Dum07]. The planning for the trucks also takes place at an operational level and therefore is not part of the LHPTP. We count the number of trucks on each arc, but we do not plan their tours nor the tours of the drivers. The company for which we design the network uses third-party providers thus does not address this problem. The containers used to contain the parcels (also called swap bodies) are balanced as they belong to the postal company, but we do not make planning for the containers, they are interchangeable (container balancing is analogous to the previously discussed vehicles balancing).

4 Algorithmic Approaches

The postal system, by nature, is built on a divide-and-conquer principle [Knu99]. Indeed, Knuth explains that a large collection of letters can be sorted into separate bags for different geographical areas; each of these bags then contains a smaller number of letters that can be sorted independently of the other bags, into finer and finer geographical divisions. (Bags of letters can be transported nearer to their destinations before they are sorted further, or as they are being sorted further.) This divide-and-conquer approach is based on some natural or imposed hierarchical structure of the relevant network. For example, in our network, there is a hierarchy of sites: The sorting centers are at a regional level, the delivery depots at a finer level and the post offices at an even finer level.

We follow this general intuition. We identify hierarchical structure in our network, and our two algorithmic approaches are essentially divide-and conquer-algorithms that exploit this structure. Our approaches are based on using combinatorial algorithms to solve a Mixed Integer Linear Program (MILP, defined in Chapter 1 Section 4.3), which are too large to be solved by simply plugging into a solver. Here, we give a brief overview of our algorithmic approaches, since this is one of the main components of this thesis. This high-level description can be understood even without the precise problem description (found in Chapter 2).

The first algorithm we propose is based on dividing the sites into clusters. We cluster the sites so that a parcel can be routed either inside a cluster (intracluster problems) or between clusters (intercluster problem). This division results in subproblems of small size which can be solved with an exact method such as an MILP. Then we solve a global MILP carefully designed to combine the locally optimized solutions in an appropriate way. Clustering-based approaches have been applied to other transportation problems such as Vehicle Routing Problems (VRP) which face a similar hurdle with scalability issues. These previous works tailor their clustering approaches specifically to their objective or problem data. Our approach is different since we apply well-known, off-the-shelf, clustering algorithms to obtain smaller size subproblems, which we can then solve rel-

actively quickly and/or in parallel. We compare these clustering methods to determine if they have an impact on the results obtained with the approach we propose and if so, which one works best on our data instances provided by a postal company. We note that, as is typically the case when clustering algorithms are used in a black-box manner, the difficult aspect is to map our problem instance to the input of these algorithms. Specifically, the problem of determining an accurate distance function and/or similarity function to obtain meaningful and useful clusters is non-trivial. This algorithm allows to check first if divide-and-conquer approaches are promising on the LHPTP and secondly to find out how we should divide. Thus we tested the different clustering methods with various parameters. We learned that respecting the hierarchy of sites is probably the most important aspect for a clustering, even if some shortcuts (that bypass this hierarchy) are sometimes efficient to save money.

The second algorithm we propose exploits the most obvious hierarchical aspects of the network. Since we are working on a network that is naturally a two-level hierarchical transportation network (e.g., containing three types of sites) and since the results of the previous algorithm confirm the importance of respecting the hierarchy of sites, we want to exploit the hierarchical structure further when designing an algorithm for long-haul parcel transportation. On the one hand, strictly enforcing that the demands follow the network structure (e.g., only use paths in the hierarchical network) would possibly limit some paths resulting in suboptimal routings. On the other hand, the simplicity of the network structure is useful because it allows to reduce the number of (operational) paths in the network. Thus, our research goal is to understand and find a good trade-off between the simplicity of hierarchical network design and the optimality of the solutions produced by our algorithms. In fact, we add shortcut/direct paths for some demands, as this results in avoiding some set of suboptimal routings, and does not complicate our algorithm too much. To discriminate the demands, we introduce a threshold on the truck filling rate. Indeed, it seems natural to maximize the truck filling rate and minimize the number of parcels sorted. The truck filling rate threshold will help to find a good trade-off between the benefits of routing demands directly and the disadvantages of sending suboptimally filled trucks. Roughly speaking, the demands above this threshold can be routed directly while the ones below this threshold follow the hierarchical structure of the network. Furthermore, we remark that in the first algorithm, the intracluster subproblems are solved before the intercluster problem. This leads us to build the second step (intercluster step) of the solution on top of a first-step solution that is already possibly not optimal and could amplify errors from the first level. We note that in the second approach, we build a solution the other way around.

5 Manuscript Organization

In Chapter 1, we give background and present previous works in the domain of parcel transportation. We present standard problems close to the LHPTP in order to position it in the literature. First, the characteristics and inherent issues of parcel transportation

problems are presented. A wide spectrum of parcel transportation problems is described in order to be able to position the LHPTP precisely within this domain. Then network design problems related to the LHPTP are explored: Point-to-point delivery problems, Service Network Design problems, Service Network Design problems with Asset Management and network design problems with twin-trailers are introduced. Since the network on which the LHPTP occurs is a hub-and-spoke network, this type of network and the specifics of its design are introduced. We also give some background for linear programming to allow a reader unfamiliar with this topic to understand the rest of this manuscript.

Then we give an extended description of the LHPTP in Chapter 2. We give the application framework of this industrial problem. After this, the optimization problem is described: Its inputs, its outputs and its constraints are detailed. Finally, a formal definition of the LHPTP is given, including: The detailed presentation of the two levels of the network on which the parcel service is designed, the description of the different graphs which depict the LHPTP, and an MILP formulation for this optimization problem.

In Chapter 3, we present and analyze the case study and datasets provided to us by a postal company on which we test our algorithms. These datasets cover a six year period and thus they demonstrate the evolution of the postal network. We start by analyzing their structure to understand this evolution. Then, we discuss the impact of two soft constraints on the size of the MILP formulation. Indeed, as they are not strong constraints, we do not have to strictly respect them, and this might be useful in our case study. On the datasets we have, the MILP presented in the previous chapter allows to find feasible solutions but their quality depends heavily on the strategy we use (e.g., whether we use the soft constraints or not, the possible paths that we propose, etc.) and on the computation time. Thus we want to design heuristics able to find better solutions than the ones generated by an MILP solver in a limited time.

As the MILP works well on small instances, we aim to propose a heuristic approach based on a divide-and-conquer paradigm and clustering techniques which exploit an existing MILP (or other methods) for computing a solution. In Chapter 4, we divide the problem into k clusters of sites. Thus parcels are either routed inside a cluster or between clusters. The intraccluster problem are small enough to be tractable by our MILP formulation, but addressing well the intercluster problem is more technical. We try to divide the problem into various numbers of clusters: two, three and four clusters in order to find out which option works best on our data instances. We also test well-known clustering methods to determine if it impacts the results obtained with the k -Clusters Algorithm and to determine which properties of the clustering are needed to have good results. To do so, we start by giving some background on standard clustering techniques and discuss their use in transportation optimization. Then we propose an algorithm that combines clustering techniques with our MILP to obtain a feasible solution to the LHPTP that is better than the solution obtained using the MILP alone. This algorithm is called the k -Clusters Algorithm. We also explain the technical details (e.g. similarity function, input, etc.) of the clustering methods we use as well as how these methods are applied in our algorithm. Applying these clustering methods on the datasets highlights structural

properties to add to the one presented in Chapter 3. Finally, we present the results of this first divide-and-conquer algorithm.

In the second algorithm, we take advantage of the hierarchical structure of our network by solving the problem from sorting center to sorting center first, and then extending this partial solution to send parcels to the delivery depots. This algorithm is presented in Chapter 5 and is called the Hierarchical Algorithm with Aggregation of Demands. A feature of this algorithm is that trucks filled more than a truck filling rate threshold are sent directly (bypassing all sorting operations), rather than only those that are fully filled. As we exploit the two-level structure of the network, we give some background about this type of network. We test various values for the truck filling rate threshold in order to measure its impact on the quality of the solutions. This threshold helps to find a good trade-off between the benefits of routing demands directly and the disadvantages of sending sub-optimally filled trucks. The results of this algorithm on our case study are presented. Then, a comparison of the results of the two divide-and-conquer algorithms is given.

Finally the Conclusion contains the conclusions and perspectives of this work.

In this PhD thesis, specific vocabulary is introduced. Thus we provided a glossary (see Chapter 2) at the end of the manuscript. The words in the glossary are in bold in the text of the manuscript the first time they are defined in order to help the reader to spot the definitions. Also some key ideas are underlined in the text to help the reader.

Chapter 1

Background and Previous Approaches

Contents

1	The Wide Spectrum of Parcel Transportation Problems	12
1.1	Transportation Mode	14
1.2	Transportation Network	15
1.3	Sorting and Other Logistical Aspects of the Parcel Transportation	16
1.4	Optimization Objectives	17
1.5	Overview of Computational Approaches	18
1.6	Synthesis	18
2	Network Design in Freight Transportation Problems	19
2.1	Point-to-Point Delivery Problems	19
2.2	Service Network Design Problems	20
2.3	Service Network Design with Asset Management	22
2.4	Trucks with Twin-Trailers	23
2.5	Synthesis	24
3	Design of Hub-and-Spoke Networks for Freight Transportation	25
3.1	Hub-and-Spoke Network	25
3.2	Hub-and-Spoke Network Design	27
3.3	Hub Location Problem	28
3.4	Synthesis	29
4	Linear Programming and MILP Decomposition	30
4.1	Mixed Integer Linear Programs	30

4.2	Classical Optimization Techniques	31
4.3	MILP Decomposition via Divide-and-Conquer Paradigm	32
5	Conclusion	33

In this chapter, we give some background and present the previous approaches for parcel transportation optimization problems in the literature. As stated in the Introduction, we study the parcel transportation optimization on the long-haul section of a parcel trip, which is called the Long-Haul Parcel Transportation Problem (LHPTP). This problem is the focus of this thesis, but we defer the precise definition until the next chapter, because we want to first present the industrial/practical context before the problem definition in order to best motivate the choice of variables and constraints that constitute our model and ultimately define the problem. To present this context, we first introduce the industrial problems and gradually increase the level of abstraction. Indeed, we present parcel transportation, service network design, hub-and-spoke networks and linear programming vocabulary.

In the LHPTP, all the parcels have to be delivered in less than 48 hours, the capacity of the vehicles used has to be respected and these vehicles must be balanced (e. g. returned to the sorting centers in order to be able to deliver the parcels the next day). This precise set of constraints is not found in one standard optimization problem in the literature. Thus we will both present parcel transportation optimization problems and the multiple standard problems which are related to it.

Section 1 introduces the parcel transportation problems, their vocabulary and inherent issues. The characteristics of these types of problems are introduced and studied. We choose to survey many related problems in the general area of parcel transportation (not just long-haul) to have a clear view of the wide spectrum of parcel transportation problems and to be able to position the LHPTP precisely within this domain. Section 2 presents network design problems, which are related to the LHPTP. The LHPTP has point-to-point demands, a design-balance constraint and we design a service network for trucks with twin-trailers. Thus problems having these characteristics in the literature are introduced and analyzed. Well-defined (more general) problems such as Service Network Design problems, and Service Network Design problems with Asset Management are introduced and analyzed. In addition, somewhat more specific point-to-point delivery problems and network design problems with twin-trailers found in the literature are also presented. Since the network on which the LHPTP occurs is a hub-and-spoke network, this type of network and the specifics of its design are described in Section 3. Finally in Section 4, linear programming techniques and their use combined with divide-and-conquer are reviewed. Note that in the domain of parcel/freight transportation, academic publications have quite diverse formats. In other words, for some papers the main contribution is the problem presented, while for others it is their model or the algorithm introduced or the method to solve a problem. For each paper we cite, we try to describe its main contribution, which is why our presentation may seem to be a bit heterogeneous.

1 The Wide Spectrum of Parcel Transportation Problems

Nearly all countries of the world have a mail service and a parcel service [BDBP20]. Usually parcel services have a universal service obligation [CK98]: Each citizen of the country is able to receive and send both mail and parcels. These services are of large-scale and therefore the delivery costs are of large-scale. Optimizing these services can allow the companies that provide them to achieve economies of scale. Therefore, lots of studies on this subject have been made. But all the optimization problems formalized in these studies have different characteristics (see Table 1.1). For instance, companies do not use the same type of vehicles, as this is strongly related to the country configuration and size: Some “small” countries can perform next day delivery via trucks while larger countries need planes or trains. The network configurations follow more or less always the same scheme (presented in the Introduction), it is divided into four main transportation stages: Collection, Long-Haul Transportation, Distribution, Delivery (see Sebastian [Seb12]). In the LHPTP we deal with long-haul transportation. In the papers presented in this section, the problems deal with the optimization of at least one aspect of the long-haul transportation, and sometimes the problems deal with the optimization of long-haul transportation including some other aspects of the process.

	air transportation	rail transportation	road transportation	homogeneous fleet	limited fleet	origins \neq destinations	2-level network	vehicle balancing	bulk transportation	disaggregate shipping	sorting capacity	D+1 optimization	fixed hubs
LHPTP			x			x	x	x	x	x			x
Barnhart et al. [BKKW02]	x			x	x			x				x	x
Batista [Bat16]			x	x	x			x					\
Baumung and Gunduz [BG15]			x	\	\				x	x	x		
Baumung et al. [BGMS15]			x	\	\	x			x	x	x	x	
Ben Ayed [BA11]	x		x										x
Bruns et al. [BKS00]		x	x										
Cohn et al. [CRWM07]			x					x					x
Grünert and Sebastian [GS00]	x		x		x			x	x	x	x		
Jansen et al. [JST ⁺ 04]		x	x		x			x			x		x
Kim et al. [KBWR99]	x			x	x			x			x	x	x
Lee and Moon [LM14]			x	x				x			x		
Lin et al. [LZL20]	x	x	x								x		x
Meisen [Mei15]			x	\	\	x				x		x	
Ngamchai [Nga07]	x			x				x			x	x	x
Zäpfel and Wasner [ZW02]		x	x					x			x		
Zhang et al. [ZPM21]	x		x		x		x						x

Table 1.1: Characteristics of long-haul parcel transportation optimization problems

Nearly all the postal companies have a sorting operation (which groups parcels by destination) with a cost per parcel, and use some sites called hubs to consolidate demands. One exception to these rules seems to be the case of South Africa studied by Batista [Bat16]. Note that when a case in Table 1.1 is not relevant (no fleet or no hubs, there is a backslash in it).

Table 1.1 lists the characteristics of long-haul parcel transportation problems that we will introduce and discuss in the rest of this section in order to position the LHPTP. With respect to all of these characteristics, we can say the LHPTP is unique and new to the best of knowledge as it is the only one with this combination of characteristics in Table 1.1.

In Section 1.1, we present the transportation modes used in the general domain of parcel transportation. This corresponds to the five first columns of Table 1.1. In Section 1.2, we discuss common characteristics of road transportation networks (Columns 6 to 8 of Table 1.1). Then, in Section 1.3, we explain how the sorting operation properties define the parcel transportation problems (Columns 9 to 11 of Table 1.1). Finally, we

present in Section 1.4 the various optimization objective which can occurs in long-haul parcel transportation (last two columns of Table 1.1).

1.1 Transportation Mode

In long-haul parcel transportation, various transportation modes can be used depending on the choices made by a postal company. In this section, we introduce the long-haul parcel transportation problems which occur with air transportation, rail transportation and multimodal transportation. We compare the characteristics of these problems with those involving only road transportation. Maritime transportation is not deployed for parcel transportation within a country to the best of our knowledge.

One of the main studies on parcel transportation by air is the one of Kim et al. [KBWR99], which concerns UPS in the USA. Their objective is to find the cost minimizing “movement” of packages from their origins to their destinations, given very tight service windows, limited package sorting capacity, and a finite number of ground vehicles and aircraft. Their network of sites are already given. They develop a route-based model for large-scale transportation service network design problems with time windows. In their paper, valid inequalities and reduction methods are introduced to fit their problem and make it feasible for a MILP solver despite of its poor linear programming bounds. The heuristic of Kim et al. [KBWR99] is improved by Barnhart et al. in [BKKW02]. In their article, Barnhart et al. present a heuristic approach for the express shipment delivery problem in the USA (8 hubs, 258 locations and 54,563 demands) ¹. It divides the service network design problem into two subproblems: route generation and shipment movement. They alternate the resolution of each subproblem until they have a good solution, which they define as each demand being routed on a single path. This heuristic, which could be seen as a route-and-flow-generation algorithm. This is not the case of the LHPTP since it allows disaggregate shipping.

Zhang et al. [ZPM21] focus on consolidation when optimizing courier service network efficiency in the USA. They present an Integer Linear Program (ILP) to enhance consolidation for courier service network design. Finally, on the topic of parcel transportation by plane in the USA, Ngamchai [Nga07] studies parcel transportation by plane in the USA, which can be compared in terms of travel time to transport by trucks in France based on the relative sizes of the countries. However, the problem studied by Ngamchai can benefit from time lag which allows facilities to be open at different moments (because the metropolitan USA covers four times zones), which is not the case in our problem, as we deal with road transportation which do not change time zone.

Long-haul transportation of postal and express shipment in Germany (14 hubs and around 100 depots) is studied by Grünert and Sebastian [GS00]. They divide their study into three subproblems: Air transportation, ground transportation and scheduling. They solve each subproblem separately and merge the results. They call their problem the general pickup-and-delivery with hubs problem.

¹We mention such details when they are available in the article and seem relevant.

Another aspect of parcel transportation is combining different modes of transportation such as rail transportation, air transportation and road transportation. This was recently studied in the case of courier delivery in China (32 sorting centers, 36 local distribution centers and 152 delivery depots) [LZL20]. This was also studied by Jansen et al. [JST⁺04] who work for Deutsche Post Transport. They study the operational planning of a large-scale multi-modal transportation system which combines trucks and trains. They take into account the repositioning of containers as we do.

For the rest of this manuscript, we discuss mainly road transportation models, although sometimes the models involve multi-modal transport or more abstract models. In the LHPTP, the transportation of parcels is made via two types of vehicles: trucks with one or two containers. Zäpfel and Wasner [ZW02] consider these two types of vehicles. They optimize the parcel delivery network in Austria (10 depots and one hub compared to 18 sorting centers and hubs and more than 200 depots in our case) using an MILP. To build the right model, they investigate different types of networks. They show that a pure hub-and-spoke network is not optimal and study the addition of direct paths between collection points and dispersion points to the pure hub-and-spoke network. This encourages us to consider these types of paths in our problem. Cohn et al. [CRWM07] also have these two types of vehicles. This means that in their problem and in the LHPTP, we have an **heterogeneous fleet**. Moreover, in the LHPTP, unlike [BKKW02, Bat16, GS00, JST⁺04, KBWR99, ZPM21], the fleet size is not limited (a third-party logistics provider supplies trucks).

1.2 Transportation Network

In postal and parcel transportation, both origin and destination of parcels are known and given. Demands for delivery are **point-to-point** (see Section 2.1 for more details about point-to-point problems). In the network provided in our case study, the origin sites and the destination sites of the long-haul transportation are not the same sites. This is very rare in the literature as only Meisen and Baumung et al. [BGMS15, Mei15] seem to have different types of sites for the origin sites and the destination sites.

The LHPTP uses a two-level hub-and-spoke network (see Section 3 for a definition of this type of network and Chapter 5 Section 1 for a discussion of two-levels networks). The only long-haul parcel transportation problem in the literature to the best of our knowledge which has such a two-level network is combining air and road transportation [ZPM21]. In a pure road transportation network, many parcel transportation problems have two types of sites but in the LHPTP we have three types of sites: the delivery depots, the sorting centers, and the **inner-hubs** which is the name we give to the sorting centers between the collection sorting center and the last sorting center. We are not aware of any network with three types of sites linked by two levels in the long-haul transportation segment. Some networks are described as two-level networks (like [BG15, BGMS15, BKS00, JST⁺04, LM14, LZL20, Mei15]), but only the core level transportation is optimized.

The Swiss parcel delivery services restructured by Bruns et al. [BKS00] takes place on a network close to our network in terms of its daily functioning. Indeed, if we forget the

fact that we have one more level in our network, the networks have the same structure. They have some fixed locations for the sorting centers and want to locate the distribution sites and determine how many of them there should be. Even if they can relocate the existing facilities, this work had been made by a private firm and is not the main topic of the article. They emphasize on how they compute the costs. Their objective is to design an optimal network with respect to cost. However, some of their findings are interesting for our problem: For instance, they prove that the use of the railway is not competitive because the reloading costs a lot. As the costs are not very precise, they do a sensitivity analysis to check that their results will still be relevant if the costs change a bit. It gives them an insight of the cost structure.

As in the LHPTP we design a daily transportation plan, the vehicles have to be balanced over the course of a day for each site; we need to return empty vehicles to appropriate locations. This constraint is called vehicle balancing. It is quite classical as it appears often [BKKW02, Bat16, CRWM07, GS00, JST⁺04, KBWR99, LM14, Nga07, ZW02]. Vehicle balancing in parcel transportation is a major concern of Cohn et al. [CRWM07]. They study small package delivery in the US freight transportation network (for UPS). They state that in the optimization of small package transportation the planning process is often decomposed into sequential subproblems and this decomposition degrades solution quality. Therefore, they combine in the problem they investigate both load matching and routing and equipment balancing (LMREB). They propose an arc-based model on a time-space network which is a LP with non-linear cost structure. This LP is too large to be solved on realistic datasets, so they present a cluster-based approach which reduces it. This cluster-based approach removes equivalent movements which are movements located on a "cluster" of sites. They also use column generation. More general considerations about vehicle balancing will be presented in the Section 2.3 about Service Network Design with Asset Management.

1.3 Sorting and Other Logistical Aspects of the Parcel Transportation

In the LHPTP, parcels are transported in bulk in containers (like in [BG15, BGMS15, GS00]). The main other options being when there are compartments in the vehicles which permit to deliver parcels to different delivery depots consecutively. This is not the case in the LHPTP: Each container is completely unloaded in each site it goes through.

In the LHPTP, we do not consider **sorting capacity** in the sorting centers, unlike in [BG15, BGMS15, GS00, JST⁺04, KBWR99, LM14, Nga07, ZW02]. The sorting capacity is guaranteed by the soft constraint of the catchment areas (like Bruns et al. [BKS00]). This constraint means that each delivery depot is assigned to a sorting center and that parcels whose destination is this delivery depot should be sorted in this sorting center and not in other sorting centers.

Disaggregate shipping is a term defined by Leung et al. [LMS90] which states that all the parcels from one origin to a destination do not have to use the same path in the network. Not all the authors clearly mention if they allow this option or not as it seems

that it is supposed to be obvious when it is allowed and it is not allowed (e.g., for purposes of traceability). In the LHPTP, we allow disaggregate shipping as [BG15, BGMS15, GS00, Mei15]. Baumung et al. [BGMS15] introduce the structure of the German postal network for letter mail and for parcel mail, which is like ours. They present two models: The location/allocation model and a model which adds service quality constraints (next day delivery for 90% of parcels). This last model cannot be solved to optimality using solvers. Therefore, they developed metaheuristics to solve it. The solution of the hub location problem (first model) is the key to the strategic optimization of the overall network in the case when subsets of sorting centers are used as hubs in the long-haul transportation subnetworks. Baumung and Gunduz [BG15] focus on consolidating residual volumes to optimize standard parcel distribution tested on the data of Australia Post (from 10 to 50 sorting centers). They call their problem the Residual Volumes Hub Location Problem. The main idea of their heuristic is to route the small amounts of parcels that remain after full containers have been routed (called residual volumes). They present an MILP and valid inequalities which improve its resolution.

1.4 Optimization Objectives

While the main objective of long-haul parcel transportation is the cost minimization of the total transportation and sorting operations, there are other objectives which can occur.

The **quality of service** is one of the classical objectives in parcel transportation. It consists of delivering most of the parcels within 24 hours and is called D+1 delivery (or next day delivery). Generally, in road transportation, a D+1-ratio is given by the postal company [BGMS15, Mei15]. Meisen [Mei15] works on D+1 delivery of parcels in Germany (33 hubs and 200 depots). He introduces an MILP which can be solved by introducing some clever limitations of the solution space (e. g., valid inequalities). Here, the D+1 delivery is the objective of the optimization process. He studies the transportation costs variations according to the D+1-ratio guaranteed to the customer. In this manuscript, we do consider time delivery constraint within 24 hours and 48 hours but without ensuring a D+1 quality of service. Note that some countries are small enough that the D+1 delivery is not an optimization objective. While for bigger countries, the air transportation is required to ensure next day delivery [BKKW02, KBWR99, Nga07], which is why they address the problem of express shipment.

Some issues, as the objectives of Batista [Bat16] and Ben Ayed [BA11] are very rarely mentioned in the literature. Ben Ayed [BA11] is planning the parcel delivery scheme for an anonymous company, the security of the delivery (drivers and packages) is one of their focuses, along with the costs and the time of delivery. He uses a MILP to optimize the network (22 stations, 13 hubs). He develops the practical issues they met when applying their schedule, with the reluctance of workers to change methods. And he is the only one in the literature to develop this. Batista [Bat16] works on the case of parcels of South Africa Post Office, his objective is to reduce customer complaints. Thus he presents route optimisation techniques to improve the parcel service. The network is composed only of mail centers which are delivered by trucks (26 mail centers, 2500 operating retail outlets).

This appears to be a somewhat unique design for a postal network as they do not have distribution centers (or delivery depots as we call them) which would have reduced the number of origin-destination pairs through consolidation.

Finally, in some parcel transportation optimization problems, strategic hub location occurs. The Hub Location Problem (defined in Section 3.3) consists in picking some sites as hubs among a set of sites in a way which minimizes the transportation costs. Thus the parcel transportation problems can be divided between Hub-and-Spoke Network Design problems [BG15, BGMS15, BKS00, GS00, LM14, Mei15, ZW02] (defined in Section 3.2) and Service Network Design problems [BKKW02, BA11, CRWM07, JST⁺04, KBWR99, LZL20, Nga07, ZPM21] (presented in Section 2.2). In our optimization problem, we have some fixed hubs proposed but we can add new hubs if we want. It seems that very few countries have ever completely redesigned their postal networks with the exceptions of Slovakia [MMŠ15], Czech Republic [JL99] and South Korea [KLP14]. This will be detailed in Section 3.1 and Section 3.3.

1.5 Overview of Computational Approaches

With respect to the solving methods, most of the works presented in this section propose an MILP to address their parcel delivery problems [Bat16, BG15, BGMS15, GS00, LM14, LZL20, Mei15, Nga07, ZW02]. Integer Linear Programs are proposed by [BKKW02, BA11, CRWM07, KBWR99, ZPM21]. Zäpfel and Wasner [ZW02] and Baumung and Gündüz [BG15] have introduced valid inequalities to strengthen their models. Barnhart et al. [BKKW02], Kim et al. [KBWR99], Cohn et al. [CRWM07] and Grünert and Sebastian [GS00] present specially designed heuristics which divide the problem into subproblems solved separately. Note that this is not divide-and-conquer since in the divide-and-conquer paradigm, the original problem is divided into subproblems which are the same problem as the original one [CLRS09] while in these heuristics, the original problem is divided into different problems. Lin et al. [LZL20] use a Simulated Annealing algorithm to solve their problem. Local search algorithms are used in [BGMS15, BA11]. Genetic algorithms are used in [Nga07]. And a constructive heuristic is proposed by Zhang [ZPM21].

In this thesis, we will present an MILP to model the problem and we will also introduce MILP-based divide-and-conquer heuristics to solve the LHPTP. A reminder of the vocabulary of linear programming is given in Section 4.1.

1.6 Synthesis

In this Section, we have classified the long-haul parcel transportation optimization problems found in the literature according to the characteristics presented in Table 1.1. The first way to classify these problems is the transportation mode, which strongly impacts the optimization models proposed. The second way to distinguish these problems is the format of the network on which the parcel service is designed. The way the parcels are transported is also very important, as well as the optimization objective. With respect to

all of these characteristics, the LHPTP is unique with respect to its position in Table 1.1.

2 Network Design in Freight Transportation Problems

There is a family of problems called network design problems which is described by Magnanti et al. [MW84]. This type of problems tackles the three levels of decision-making that arise in transportation optimization [Cra98, Cra00].

Historically, the first problem of the network design family studied was the general Network Design Problem [JLK78]. In the 80's and 90's, the Hub-and-Spoke trend appeared and this type of network was broadly studied and used for transportation optimization (see Section 3.1). In 1986 Crainic defined the Service Network Design (SND) problems [CR86] in which transport services have to be defined and operated over a physical network. Service network design can be applied to various domains including public transportation [GH08] and freight transportation [Cra03]. We will focus on the latter in this manuscript. In freight transportation, SND is one of the two possible problem categories according to Steadieseifi et al. [SDN⁺14], the other one being the network flow planning. This latter problem addresses the optimization of the movement of commodities throughout the network which is already designed. For more details about multicommodity network flows, which are in general not NP-Hard problems, one could refer to Ahuja et al. [AMO88], or Tomlin [Tom66]. Note that the Service Network Design can use network flow as a subroutine while the reverse is not true.

In Section 2.1, we introduce point-to-point delivery problems which are historically the first problems addressed within the family of network design problems. Then, Service Network Design (SND) problems are presented in Section 2.2 and their extension with Asset Management (SNDAM) in Section 2.3. Finally, we give the specifics of Service Network Design with trucks twin-trailers in Section 2.4. Thanks to all these definitions, we will be able to position the LHPTP within the domain of Service Network Design in the synthesis at the end of the section.

2.1 Point-to-Point Delivery Problems

In the LHPTP, the objective is to deliver point-to-point demands in a network in which consolidation and sorting are available. Problems with point-to-point demands for delivery have been studied as **point-to-point delivery problems**. Li et al. [LMSL92] discuss the computational complexity of these types of problems. These problems involve shipping one item from each one of p sources to p destinations. The destinations might be prematched to sources (the fixed destination case, which occurs for parcel or passenger transportation), or a source's item might go to any destination (the non-fixed destination case which occurs in freight transportation). The networks can be directed or undirected. Up to K items at once they can share a truck on an arc, and costs are linear in the number of trucks used. The authors also consider the point-to-point connection

problems, which consists in finding a minimum cost arc subset connecting sources with destinations. They state that all variations of both problems are strongly NP-hard for all $K \geq 2$, but that there are polynomial algorithms in some cases (if p is fixed, or if the underlying network is an undirected grid with sources on one side, destinations on the other). The directed fixed destination version of this problem is close to our LHPTP but there are some differences: It does not have sorting costs, nor does it offer the possibility to do disaggregate shipping. Moreover, the path length is not limited in this problem. If the first difference do not change the complexity, the second and third ones might impact it.

Leung et al. [LMS90] also work on the point-to-point route planning problem. They have a delivery network with nodes of two types: terminal nodes and distribution centers (respectively our delivery depots and sorting centers). They decompose their optimization problem into two subproblems: An assignment problem with capacity constraints and a multicommodity flow problem. They present an MILP for the first subproblem. They do not discuss complexity even if they emphasize that the cost is not linear as it is a step function which depends on the vehicles.

After the end of the 90s, the term “point-to-point” was deprecated for the benefit of Service Network Design which covers both point-to-point demands and multi-origin or multi-destination demands. Nowadays, the term “point-to-point” is used to talk about networks and means there are direct connections between sites (see Figure 1.1a). For instance, Kim et al. [KLP14] distinguish the point-to-point network (with direct connections between all pairs of sites), the hub-and-spoke network, and hybrid networks which are a combination of both. These last two types of network will be discussed in Section 3.1.

2.2 Service Network Design Problems

In the LHPTP, we design a network for a parcel transportation service, thus it is a Service Network Design problem. The **Service Network Design (SND)** problem is an optimization problem, defined by Crainic and Rousseau [CR86], which addresses the tactical issues of selecting and scheduling the services to operate on a network as well as the logistics operations and fleet routing necessary for the service. A very complete description of Service Network Design in freight transportation is given in [Cra00]. SND in freight transportation has been surveyed by Crainic [Cra98] and Wieberneit [Wie08]. Models and methods for solving SND problems are presented extensively by Krishnan [Kri98]. Costa [Cos05] provides an overview for the fixed charge SND. It is a SND problem in which there are fixed costs when opening a link in the network. We already presented SND for parcel services [BKKW02, KBWR99, LZL20] in Section 1. We introduce in this section SND problems which concern freight transportation, the main differences being the transportation mode.

The Express Shipment SND is defined by Ruckle [Ruc18] as the problem of determining the lowest cost way to use a heterogeneous, multi-modal vehicle fleet to move a set of express shipment packages from their origin airports to their destination airports within

a domestic multi-hub-and-spoke network and within the guaranteed service periods. She proposes a model for this problem and approaches to solve it. A multi-hub version of ESSND with flexible hub assignment is studied in [PLT18]. An overview on Express Shipment Service Network Design is proposed by Taanman [Taa07]. In this paper, the focus is on express shipment delivery companies and the express shipment delivery problem. The solution frameworks that are used in practice by the 4 major leading companies (UPS, FedEx, DHL and TNT) on this area are reviewed. This problem is a SND problem for parcel transportation, but the air transportation do not have the same issues as road transportation, thus it cannot be applied as is to our problem even if there are similarities.

The Express Train Service Network Design Problem is studied by Liu et al. [LLWZ18]. They formulate it as a non-linear integer programming model whose goal is to find a service network and shipment routing plan at minimum cost. Lan et al. [LHX20] investigate the scheduled service network design problem in rail freight transportation and solve it thanks to a combination of Benders decomposition and column generation. More information on railway SND can be found in Zhu [Zhu11].

SND for an intermodal container network with flexible due dates/times and the possibility of using subcontracted transport is studied by Riessen et al. [RNDL15]. Their model uses a combination of a path-based formulation and a minimum flow network formulation. Two new features are integrated to the intermodal network-planning problem: Firstly, overdue deliveries are penalized and not prohibited, and secondly, the model combines self-operated and subcontracted services. They consider the network-planning problem at a tactical level as we do.

The integrated SND for a cross-docking supply chain network is presented by Sung and Song [SS03]. The problem is modeled as a path-based formulation for which a tabu-search-based solution algorithm is proposed. In supply chain optimization, the Logistics SND is defined and studied (see [DLPR18]) but this problem is too far to parcel transportation as in the LSND problem, customers demands may be offered in multiple origin locations.

Concerning long-haul freight transportation, an original new problem, the Service Network Design and Routing Problem, is treated by Medina et al. [MHLP19]. This problem integrates long-haul and local transportation planning decisions. In postal terms, they treat both long-haul transportation and delivery. They present both an arc-based and a route-based MILP formulations for their problem. They solve each with a dynamic discretization discovery algorithm. The scheduled service network design of long-haul freight transportation presented by Li et al. [LJL⁺21] is also unique in the sense that the synchronization departure time for commodity and vehicle at an arc is unknown in advance. The vehicles which deliver the commodities are rent and not balanced at all. It makes a huge difference with Service Network Design with Asset Management that takes into account the vehicle management.

2.3 Service Network Design with Asset Management

One of the main constraints of the LHPTP is the **design-balance constraint**. It states the vehicles have to go back at their starting points before the end of the day in order to be used again the next day. This leads to empty vehicle flows as once the parcel have been distributed in the delivery depot, the trucks are empty. In parcel services, the terms of “empty balancing” [GS00] and “repositioning” [JST⁺04] of the vehicles are used. In this thesis, we will use **empty repositioning** to talk about these vehicles movements. A taxonomy of empty flows and fleet management models is presented by Dejax and Crainic [DC87], they discuss the advantages of a hierarchically approach for the management of both empty and loaded vehicles in freight transportation.

Andersen et al. [ACC09a] introduce the **Service Network Design with Asset Management (SNDAM)** and clearly defines the design-balance constraint as an asset management constraint. Before this, the asset management was not always emphasized in the name of the problem: Barnhart et al. [BKKW02] explain that the SNDP, unlike the conventional network design problem (NDP), has an added degree of complexity in that the assets need to be balanced at the ends of the planning period for continuity in the service cycle. For them the NDP is what we will call SND and SNDP is what we will call SNDAM.

SNDAM has been studied by Andersen et al. [ACC09b] who present it as an extension of the design-balanced capacitated multicommodity network design problem. The MILP solver they use struggles to find solutions to the MILP presented, therefore, additional constraints are introduced to strengthen the relaxation and speed up the solution process. Heuristics based on decomposition of the problem into three main steps: construction of the network, filling vehicles with commodities and construction of the vehicle plannings are presented in Teypaz et al. [TSC10].

Multicommodity Capacitated Network Design is a variant of SNDAM. This problem consists in designing a service network with asset management and more than one commodity to route. A survey on MCND has been done by Gendron et al. [GCF99]. Chouman and Crainic [CC15] create a cutting-plane matheuristic² for the design-balanced capacitated multicommodity network design problem. The aim is to identify promising variables thanks to learning mechanisms embedded into the cutting-plane procedure in order to reduce the dimension of the problem instance and therefore make it addressable by a MIP solver.

The SNDAM is sometimes called Service Network Design with Resource Constraints [CHTV16, LDP⁺20]. This problem is addressed by Crainic et al. [CHTV16] who solve it with an approach that combines column generation, meta-heuristic, and exact optimization techniques. And, in the case of single-path SND (each commodity is delivered along a single path that prevents flow partition) by Li et al. in 2020 [LDP⁺20]. They introduce both node-arc and arc-cycle MILP formulations which are too large to be solved by a solver. They present a heuristic based on column generation to solve their MILP. Pedersen et al. call their variant of SNDAM, the Service Network Design with Asset-Balance

²A matheuristic is the interoperation of metaheuristic and mathematical programming [MF16].

Requirements [PCM09]. Both arc and cycle-based formulations for the new model are presented as well as a Tabu Search meta-heuristic framework for the arc-based formulation.

The SNDAM problem is more general than the LHPTP and it does not address some specific characteristics of the LHPTP. The parcels are routed in trucks but we also have to address the fact that they are sorted at the sites encountered during their trip and they are grouped in trucks.

2.4 Trucks with Twin-Trailers

The network design problems developed in Sections 2.1- 2.3 do not consider the same vehicles as we have: Trucks with one or two containers. The vehicles of this latter type can also be called trucks with twin trailers, or trucks with swap bodies. Here, we present network design problems which use this specific vehicle types.

As we do bulk transportation, each container is completely emptied at destination, thus two containers offer the possibility to distribute parcels in two destinations. Thus this specific fact is noteworthy.

Sebastian [Seb12] present the 2-container problem in which customers order a number of swap bodies. Each customer order is a request characterized by an origin, a destination, a pickup and a delivery time window. The problem is to assign requests (physically containers) to trucks, to route the trucks and to reposition the empty ones. Unlike him, we will not consider the containers and trucks separately in the LHPTP.

Huber and Geiger [HG14] study the Swap Body Vehicle Routing Problem. This problem is a VRP unlike the LHPTP, but they define the logistics operations which can be done with the vehicles that we manage. They present an Iterated Variable Neighborhood Search to solve their problem. The four operations they define one trucks with two swap bodies are:

- The park operation: when a truck with two swap bodies arrives at a site, leaves the second swap body in this site and then continues its path (the first swap body is the one near the trailer).
- The exchange operation: when a truck with two swap bodies arrives at a site, leaves the first swap body in this site and then continues its path.
- The pickup operation: when a truck with one swap body arrives at a site, picks up another swap body and goes out of the site with the two swap bodies.
- The swap operation: when a truck with at least one swap body arrives at a site, it leaves at least one swap body in the site and replaces it with at least one swap body met in this site.

In the LHPTP, none of these four operations are allowed.

Sun et al. [SZH15] study the case of postal express lines of a German express enterprise. They base their optimization on driving pattern of trucks, which is very rare in the

literature. To do that, they introduce three logistics operations which can occur thanks to swap bodies. The first one is the reciprocation: Sorting center A dispatches a truck which is loaded with outbound mails to sorting center B. After being unloaded, the truck returns with inbound mails (the outbound mails of sorting center B). The second one is the convection: Each sorting center dispatches a truck which is loaded with outbound mails to each other. After being unloaded, these trucks return to their sorting center respectively. The third and last one is the rendez-vous: Two trucks depart from their own sorting centers to an assigned point. When they meet at this point, the two trucks exchange their mails (usually swap containers directly) with each other and then return to their respective origins. They present an MINLP and a two-level tabu search procedure based on shipment grouping.

Eckstein and Sheffi [ES87] optimize the Group Line-Haul Operations for Motor Carriers Using Twin Trailers. They aim to minimize the transportation cost of the daily movement of trailers between a central breakbulk terminal and a set of end-of-line satellite terminals. But as each tractor can pull two trailers, there are many possibilities for creating tractor tours which perform the required pickup, delivery, and empty-balancing operations. They present an optimization approach based on a branch-and-bound framework, a Lagrangian relaxation for lower bounds and two upper-bound heuristics for solving this problem.

In the LHPTP, we have trucks with twin-trailers and the choice to use a truck with one container or two containers for each path. But we do not consider the routing of the trucks and the containers as a separate optimization problems and we do not consider other logistics operations than the sorting. Zäpfel and Wasner [ZW02] and Cohn et al. [CRWM07] also use double trailers and model it with two vehicle types as we do. Cohn et al. [CRWM07], who optimize transportation for small package carriers, emphasize that it gives a non-linear cost structure (with respect to the number of vehicles) as the cost of using a double trailer is cheaper than using two single trailer. This makes the MILP relaxation weak as it leads to a large number of nodes in the branch-and-bound tree.

2.5 Synthesis

In this Section, we have positioned the LHPTP among the family of network design problems. Indeed, the LHPTP is a service network design problem as we design a network for a parcel service. The LHPTP is also included in the subcategory of the SNDAM as it has a design-balance constraint to balance the vehicles and containers. Inside this category, the LHPTP can even be considered as a Multicommodity Capacitated Network Design problem as the vehicles have capacities. Therefore, even if the arcs of the network do not have capacities (the number of trucks is not limited) there are capacity constraints in the LHPTP.

Nearly all the papers cited here present an MILP and a heuristic to solve the problem as commercial solvers are only able to find high-quality solutions for small and medium size instances. Thus heuristics are necessary to solve large-size instances.

3 Design of Hub-and-Spoke Networks for Freight Transportation

Now, we look at the LHPTP from the point of view of type of network. We introduce hereafter the hub-and-spoke network on which we will design our parcel service. In Section 3.1, we will define the hub-and-spoke network and its variants. Then, we will describe the network design problem which occurs on this type of network in Section 3.2. Finally in Section 3.3 we will talk about the design of the hub-and-spoke network (after talking about the design on the hub-and-spoke network).

3.1 Hub-and-Spoke Network

In the LHPTP, the parcel service is designed on a network which is composed of two nested hybrid hub-and-spoke networks. Thus, we will characterize the hub-and-spoke network and its variants. A **hub-and-spoke network** has been defined by [BO99] as a network in which all links must either begin or end at a hub, the other extremities of the links being the spokes (see Figure 1.1b). The hub-and-spoke networks have been largely used for parcel services since Fedex used this type of network in the USA in the 80s [Che85, TH85]. Before that, due to the obligation to connect all sites with direct flights, the air network was a **point-to-point network** (see Figure 1.1a).

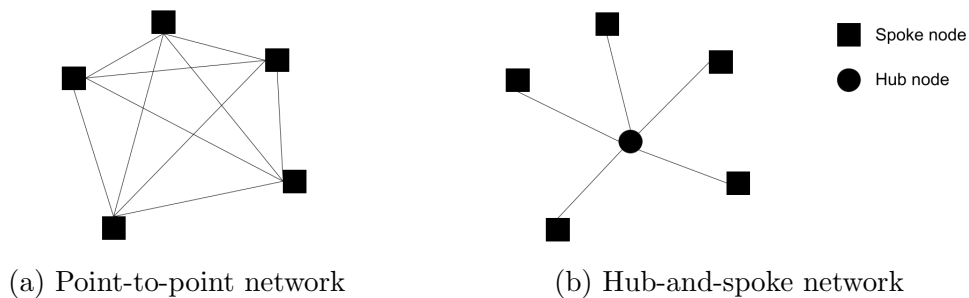


Figure 1.1: Comparison of point-to-point and hub-and-spoke networks

Hub-and-spoke networks in transportation are surveyed by Bryan and O’Kelly [BO99]. A framework to design hub-and-spoke networks in the case of air transportation is presented by Aykin [Ayk95]. O’Kelly [O’K98] analyses the hub-and-spoke network and explains why it is sometimes useful to add direct links. A hub-and-spoke network in which there are possibilities to bypass the hubs with direct links is called **hybrid hub-and-spoke network** [ZW02, ZWL07, LM14] (see Figure 1.2a) as opposed to a pure hub-and-spoke network (see Figure 1.1b).

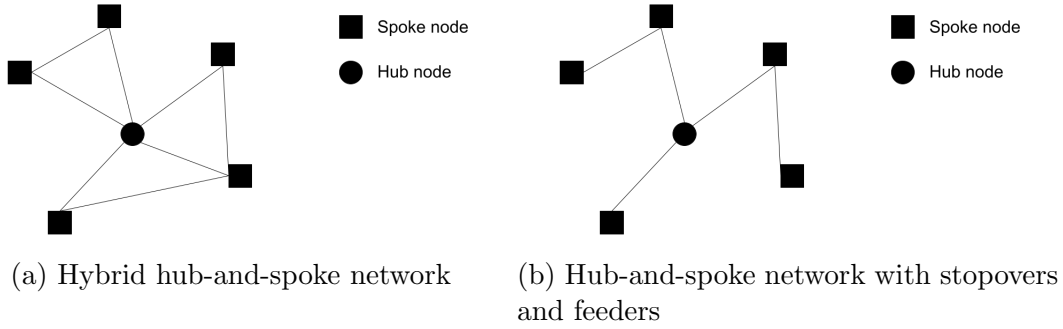


Figure 1.2: Comparison of hybrid hub-and-spoke network and hub-and-spoke network with stopovers and feeders

The **hub network with stopovers and feeders** is defined by Kuby and Gray [KG93] who study it on the case of Federal Express. This is another type of hub network in which the sites are connected by a hub, but these sites serve as stopovers in which demands from other sites can be consolidated in feeder flights (see Figure 1.2b). This network has three types of sites on two levels like the network in the LHPTP but no bypass is allowed. A comparison of pure hub-and-spoke network and hub-and-spoke network with Stopovers and Feeders on the case of Federal Express AsiaOne express network can be found in Lin et al. [LLL03].

Each one of the types of network presented above has its advantages and drawbacks. The question is which one is the most adapted to the considered problem. In the LHPTP, on one hand, we have sorting costs in the hubs, thus a pure hub-and-spoke network would be expensive; but on the other hand, we do not have enough parcels to use a pure point-to-point network. This is why we design a hybrid hub-and-spoke network to look for a good trade-off. However which direct lines are promising and should be included in the design process of the network? Zhang et al. [ZWL07] work on this question. They discuss if it is better to choose direct path or use hubs. They use a genetic algorithm on their problem which has 1 hub and 15 depots on one level. Kim et al. [KLP14] present the case of South Korean parcel delivery industry. They simulate various types of networks including point-to-point, hub-and-spoke, and multiple hub structure with parcel demand changes. They show that when demand is low the best type of network is a hub-and-spoke network, while when demand is higher, it is better to use direct paths. Zapfel and Wasner [ZW02] investigate the planning and optimization of hybrid hub-and-spoke transportation on the case of parcel delivery in Austria (1 hub and 10 depots). They have the same vehicle fleet as we do, but they have a one level network. And their demands are from depot to depot. They propose an MILP to solve their problem.

Lin and Chen [LC04] define a **hierarchical hub-and-spoke network** (see Figure 1.3b). In this type of network, each site is assigned to exactly one hub. It corresponds to our network without considering any inner-hub. They treat the time-constrained hierarchical hub-and-spoke network design problem as they consider simultaneously the time constraint and the design of routes, fleet size and schedules. Their case study is on the

network of Taiwan.

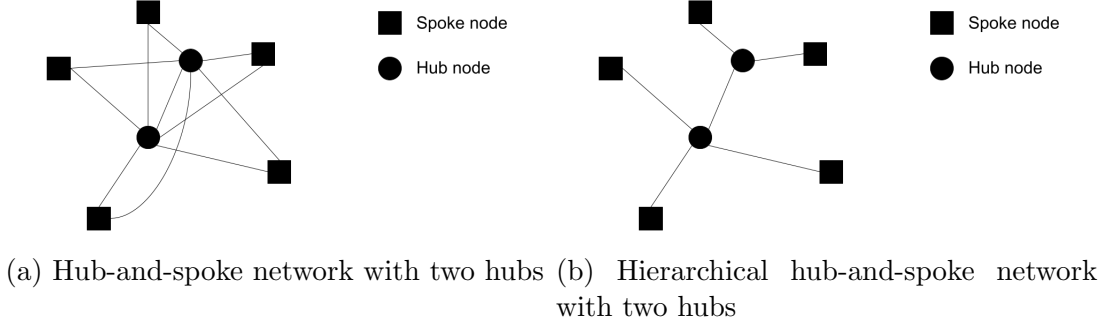


Figure 1.3: Comparison of hub-and-spoke network and hierarchical hub-and-spoke network

In the LHPTP, we have a two level hub-and-spoke network. None of the levels are hierarchical even if there is a soft constraint about hierarchy which will be discussed in Section 2 of Chapter 3. The imbrication of the two levels of a pure non-hierarchical hub-and-spoke network is presented in Figure 1.4a: The spokes of the inner level are the hubs of the outer level. For information about the hierarchical two-level hub-and-spoke network one could refer to Claes et al. [CHVG10]. The network on which we design our parcel service is represented on Figure 1.4b: There are two levels of non-hierarchical hub-and-spoke networks. The inner-level is hybrid while the outer level is a pure hub-and-spoke non-hierarchical network.

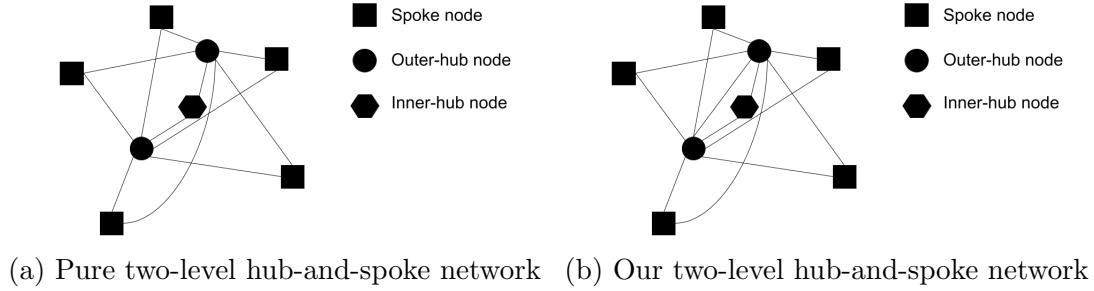


Figure 1.4: Comparison of a pure hub-and-spoke network and our hub-and-spoke network

3.2 Hub-and-Spoke Network Design

The problem of designing a hub-and-spoke network is the **hub-and-spoke network design** problem (or HSND problem), which is sometimes called hub network design problem [OM94] (HND). It involves four steps:

1. optimizing the locations of the hubs;

2. linking non-hub sites to hubs;
3. creating links between the hubs;
4. routing flows through this network.

For the LHPTP, we will focus only on the last three steps. The first one has already been done by transportation managers, but this is a soft constraint and we will discuss it in Section 2 of Chapter 3.

A review and synthesis of Hub-and-Spoke Network design is proposed by O’Kelly and Miller [O’K98]. Intermodal hub-and-spoke network design is studied by Meng and Wang [MW11]. They take into account multiple stakeholders, multi-type containers and container transfer processes at hubs. In some cases, the hubs can be capacitated, as it is the case for Serper and Alumur [SA16] who optimize the design of capacitated intermodal hub networks with different vehicle types. Their problem consists in determining the locations and capacities of hubs, which transportation modes to serve at hubs, allocation of non-hub nodes to hubs, and the number of vehicles of each type to operate on the hub network to route the demand between origin-destination pairs with minimum total cost. The generalized hub-and-spoke network design problem is studied by Lin et al. [LC08]. This problem combines the three types of hub-and-spoke networks: pure hub-and-spoke network, hub-and-spoke network with stopovers and feeders, and hub-and-spoke network with stopovers, feeders and center directs.

Wasner and Zapfel [WZ04] design for Austria (10 depots 1 hub) a delivery network such that 24-hour delivery service must be guaranteed. They present a multi-depot hub-location vehicle routing model for network planning of parcel service. They can choose the locations of the hubs and assess a path to the parcels. They raise the following questions: Why should it be necessary for network design to determine simultaneously the depot and hub locations and the routes? Thus they consider methods like the method Locate First/Route Second which decouples the problem into two sub-problems. After that, they do the two steps iteratively with feedback as one step influences the other. They consider closing some hubs or depots and opening some new ones, with both a costs aspect and a practical aspect.

3.3 Hub Location Problem

The optimization of the location of hubs in a hub-and-spoke network is the **Hub Location Problem (HLP)**. It can be incorporated to the HSND problem or be treated as a first step before a Location Routing Problem. The HLP is a Facility Location Problem known to be NP-Hard [GN11]. The HLP was originally formulated by O’Kelly [O’k87] as a quadratic integer programming model. Since Campbell [Cam94], it is referred as the *p*-hub median problem. Indeed, it is similar to the *p*-median problem defined by Hakimi [Hak65] as the problem of finding the locations of hubs which minimizes the total distance in the network. The *p*-hub median problem aims to select *p* hubs from a set of facilities and to allocate spoke nodes to hubs while optimizing the routing of freight

flows. The difference is that the p -median problem designs a hierarchical hub-and-spoke network (each spoke is linked to one and only one hub) while the p -hub median designs a non hierarchical hub-and-spoke network (each spoke is linked to at least one hub).

Reviews of the state of the art of HLP are presented by Alumur et al. [AK08] and Farahani et al. [FHAN13]. As for HLP in transportation networks, it has been studied by Gelareh and Nickel. [GN11]. For them, the HLP which occurs in transportation networks is the uncapacitated multiple allocation hub location problem. They propose a Benders decomposition approach to solve large instances for which the MILP they provide is not addressable by commercial solvers. They test it on the fixed cost values for Australian Post (AP) dataset they introduce.

Jablonský and Lauber [JL99] design the national postal distribution network in Czech Republic. They have to pick sites among the 69 local transit centres to make them become sorting centers and to determine the better number of sorting centers. The same problem is tackled by Madleňák et al. [MMŠ15]. In this last paper, they try to design a three levels network as in the LHPTP and define the types of the sites. Lee and Moon [LM14] work on the case of Korea Post (1 hub, 25 centers). They consider a hybrid hub-and-spoke network in which they select hubs thanks to an MILP which solves this Hub Location Problem and optimizes the transportation simultaneously. They do not optimize the transportation between sorting centers and depots. Hillebrandt [Hil15] investigates an original question about the design of a parcel service in Germany. She studies the re-optimization of an already existing network with a practical aspect: What happens to the nodes linked to a hub when this hub is replaced? She models the extra costs raised by the movement of hubs which is rare in the literature.

In the LHPTP, we design a hub-and-spoke network but we do not have a proper HLP as it is part of the optimization of the parcel service without being a separated problem.

3.4 Synthesis

In this Section, we define the hub-and-spoke networks and characterize the network of the LHPTP as a two-level non-hierarchical hybrid hub-and-spoke network which is hybrid on the inner-level only. The design of this type of network is an optimization problem known to be NP-Hard [EHJ⁺09] (both in the case of single and multiple allocation, for uncapacitated hubs). However, in the LHPTP, we do not locate the hubs. Some hubs are already provided and we have two strategies to test: Using these sites as inner-hubs or using all the sorting centers as inner-hubs (consequently rejecting this soft business constraint). When we use all the sorting centers as potential hubs and the best solution we have do not use some hubs, we say that these sorting centers are “simple” sorting centers while the ones used as inner-hubs are inner-hubs. We are assigning types to site, but this is not a HLP as it is a consequence of the optimization of the service. It is a HSND problem as some of the steps of HSND are tackled.

4 Linear Programming and MILP Decomposition

In the previous sections, we have presented parcel transportation problems, SND problems and HSND problems. For the first two categories of problems, the scheme is usually to write an MILP and to propose a heuristic to solve this model. In this thesis, we will present an MILP to model the LHPTP and we will also introduce MILP-based divide-and-conquer heuristics to solve it. In Section 4.1, we define what is an MILP and give the vocabulary and notions needed to fully understand the rest of this manuscript. Then, we give a quick reminder of the standard techniques used to solve MILPs in Section 4.2. Finally, in Section 4.3, we present the MILP decomposition techniques used in the literature.

4.1 Mixed Integer Linear Programs

In the previous sections of this chapter, we have seen that many authors use Linear Programs to model parcel transportation. In this section, we define this notion and its vocabulary. Linear Programming is a technique which appeared in 1947 [Dan83]. It allows to model and to solve a combinatorial optimization problems [Chv83]. A problem modeled in linear programming minimizes or maximizes an objective function under certain constraints. Each equation (objective function and constraints) is linear, that is, it can be interpreted as a linear function. Therefore, with $a_{i,j}, b_i, c_i \in \mathbb{R} \forall i, j \in \mathbb{N}$, a linear program is of the form:

$$\text{Min} \quad \sum_{j=1}^n c_j x_j \quad (1.1a)$$

$$\text{s.t.} \quad \sum_{j=1}^n a_{i,j} x_j \begin{cases} = \\ \leq \\ \geq \end{cases} b_i \quad \forall i = 1, \dots, m \quad (1.1b)$$

$$x \in X \quad (1.1c)$$

The vector x of dimension n corresponds to the variables of the linear program. The domain of definition of the variables is denoted X . The vector c corresponds to the vector of costs. In the case when X is the set of reals ($X = \mathbb{R}^n$), we are talking about **Linear Programming** (LP). If X represents the set of integers ($X = \mathbb{N}^n$) then it is an **Integer linear program** (ILP). When some variables are integers and others real numbers, we talk about **Mixed Integer Linear Programming** (MILP). This is the case for the model presented in Section 3.3 of Chapter 2. In the literature, the terms MIP and IP can also be found. The Mixed-Integer Programs regroup both MILP and MINLP (Mixed-Integer Non-Linear Programs). For clarity sake, we will not use MIP or IP. Some resolution techniques use the MILP relaxation. It is the same linear program as the original MILP except that the integer variables are real numbers: The integrality constraint of each variable is removed.

In transportation, LP can be **arc-based**, which means that there are variables which represent the flow on the arcs of the graph in which the transportation is optimized [GS00,

JST⁺04, Mei15], or **path-based** (sometimes route-based is also used [BKKW02, SS03]). This means that there are variables which represent the flow to optimize on paths or routes which are made of one or more arcs. In these models, solving the LP consists in selecting paths (or routes) among the possible paths. Note that in the literature, the path-based formulations are also called arc-chain formulations [FPL93] or arc-path formulations [Wan18], while the arc-based formulations are also called node-arc formulations [Wan18].

When a constraint has to be respected, it is called a **strong constraint**, as opposed to a **soft constraint** which can be represented with penalties in the objective function when it is not respected or can even be completely rejected.

4.2 Classical Optimization Techniques

An LP can be represented with algebraic notations:

$$\begin{array}{ll} \text{Min} & cx \end{array} \quad (1.2a)$$

$$\begin{array}{ll} \text{s.t.} & Ax \leq b \end{array} \quad (1.2b)$$

$$x \in X \quad (1.2c)$$

In this form, we can express the feasible solution space by a polyhedron:

$$P = \{x \in \mathbb{R}^n: Ax \leq b, x \in X\}$$

A solution $x^* \in P$ is optimal if and only if $cx^* \leq cx, \forall x \in P$.

The feasible **solution space** of a linear program is a polyhedron and the set of optimal solutions are on these faces. The simplex algorithm [Dan90] then makes it possible to explore the vertices of the polyhedron, more precisely the extreme points, until reaching an optimal solution if it exists.

In this manuscript, we will not use a tailor-made algorithm to solve our MILP but an MILP-solver: CPLEX (version 12.8) [Cpl09]. The name CPLEX is a word game built on the concept of a simplex algorithm written in C. So C-Simplex trained CPLEX. Nowadays, CPLEX does not use only the simplex algorithm but also other state-of-the-art techniques. We will use this MILP-solver as a black-box to solve some parts of the problems tackled here.

A key notion of linear programming is the notion of duality. Let us take again the algebraic form of the LP, called primal, with x the vector of n variables, c the vector of costs, A the matrix $m \times n$ constraints and b the column vector of the second side (see Equation (1.3)). For each constraint i ($i = 1, \dots, m$) a dual variable y_i is associated. The dual of the LP is then defined as in Equation (1.4).

<i>Min</i>	cx	(1.3a)	<i>Max</i>	yb	(1.4a)
<i>s.t.</i>	$Ax \leq b$	(1.3b)	<i>s.t.</i>	$yA^T \leq c$	(1.4b)
	$x \geq 0$	(1.3c)		$y \leq 0$	(1.4c)
Primal LP			Dual LP		

Comparison of the primal and dual LP

The strong duality theorem [Sch98, WN99] states that:
 If the primal problem (1.3) admits an optimal solution x^* , then the dual problem (1.4) admits an optimal solution y^* and $cx^* = by^*$. If the dual problem is unbounded, then the primal is not feasible.

Indeed, the solution to the dual problem provides a **lower bound** to the solution of the primal problem. The strong duality theorem implies a relationship between the primal and dual which is known as the complementary slackness. There are then primal-dual algorithms in order to solve a linear program, which solve both problems and test the optimality by measuring the **gap** between the two solution. This duality gap between the best solution found for the primal and the best solution found for the dual (called lower bound) is a good indicator of the quality of a solution thus we will use this indicator in the rest of this manuscript.

CPLEX solver, that we use as a black-box, gives a gap relative to the best possible optimal value along with the best solution it obtains using Branch-and-Cut [PR91]. The Branch-and-Cut algorithm, which is a branch-and-bound algorithm [LW66] which uses cutting planes to tighten the MILP relaxation, is solved via the simplex algorithm.

4.3 MILP Decomposition via Divide-and-Conquer Paradigm

As we have seen in the previous sections, in heuristics, MILP are often decomposed mathematically: Row generation (Benders decomposition [Ben62], Cutting-Plane Algorithm [Kel60], Branch and Cut [Mit02], etc.) or Column generation (Dantzig-Wolfe decomposition [DW60], Branch and Price [BJN⁺98], etc.). Sometimes, the problems are decomposed into subproblems of various types (e.g., Barnhart [BKKW02] dividing problem into route generation and then routing). According to Cormen et al. [CLRS09], a **divide-and-conquer** algorithm recursively divides a problem into two or more subproblems of the same or related type, until these become simple enough to be solved directly. The solutions to the subproblems are then combined (or merged) to give a solution to the original problem. With this definition, the divide-and-conquer paradigm appears to be rarely used in optimization. One example we did find is in scheduling optimization, chronological decomposition based on Divide-and-conquer methods are proposed by Kelly [Kel02].

In transportation optimization, the problems are sometimes divided into clusters to reduce their size. However, in these problems, only one optimization problem uses divide-

and-conquer paradigm combined with an MILP to solve the subproblems. Indeed, Patiño Chirva et al. [PCDCLS16] use a hybrid mixed-integer linear programming and clustering approach to optimize the selective collection services problem of domestic solid waste. This is not a SND problem as it is a vehicle routing problem (VRP). The idea to divide transportation problems into subproblems via cluster-based approaches has been explored in the literature. The definition of clustering, these cluster-based approaches and some clustering algorithms are presented in Chapter 4 Section 1.2.

Another way to divide the problem will be explored in Chapter 5, it consists in exploiting the hierarchical nature of our two level network to solve the two levels successively. The definition of two-level networks and a literature review about transportation optimization on this type of network will be done in Chapter 5 Section 1.

5 Conclusion

In this section, we compare and position the LHPTP to the most closely related problems in the literature and isolate its key features. The Long-Haul Parcel Transportation Problem (LHPTP) introduced in this thesis is a Service Network Design problem with Asset Management which takes place on a two-level non-hierarchical hybrid hub-and-spoke network which is hybrid on the inner-level only. Thus the LHPTP is also a Hub-and-Spoke Network Design problem. The LHPTP has distinct properties such as the fact that demands have both fixed origin and destination, and the sorting operation has a cost per parcel. It is a point-to-point delivery problem, with a sorting operation, which allows parcels with different destinations to be consolidated along their routes and which appears mainly in the postal transportation domain. The long-haul parcel transportation problems are really a family of problems based on various parameters (e.g., allowed logistic operations, limited fleet versus unlimited fleet, arc capacity, vehicle balancing constraints, vehicle types, etc.). None of the problems modeled in the literature is sufficient to model the specific problem that we want to solve nor to solve the variations that could arise based on the various parameters that we consider. In fact, we notice that the models presented in the literature related to this problem do not present a systematic approach to the general problem of long-haul parcel transportation. Moreover, the problem we tackle is an industrial problem containing specific strong constraints. Thus, in the next chapter (Chapter 2) after the presentation of the application framework of our problem, we introduce a specific MILP for the LHPTP.

As we said in Section 4.3, divide-and-conquer techniques are rarely applied to optimization problems. These techniques are efficient to tackle large-scale problems when it is easy to solve smaller version of the same problem. This is our case as we have an MILP which can solve quickly and exactly the small instances. Thus we will use the divide-and-conquer paradigm along with the MILP in Chapter 4 and Chapter 5.

Chapter 2

Problem Description

Contents

1	Application Framework	36
1.1	Long-Haul Transportation	36
1.2	Parcels and Demands	37
1.3	Sites	37
1.4	Vehicles and Empty Repositioning	39
1.5	Sorting and Consolidation	39
1.6	Operational Paths	41
2	Optimization Problem	44
2.1	Input: Parameters of the Optimization Problem	44
2.2	Objective	45
2.3	Constraints	45
2.4	Output: Transportation Plan	46
2.5	Key Performance Indicators	47
3	Formal Definition of the LHPTP	48
3.1	Network Provided	48
3.2	Graph Definition	49
3.3	Path-Based MILP Formulation	50
4	Conclusion	51

In this chapter, we define the Long-Haul Parcel Transportation Problem (LHPTP). As we said in the Introduction, the parcels go through lots of steps between the moment they are sent in a post office until they reach the mailbox of the recipient. The optimization problem we study is the optimization of the long-haul transportation of the parcels.

In Section 1, we describe the application framework of the problem. We define its key notions: the parcels, the sites, the vehicles used, the sorting operation, the consolidation of demands and the operational paths. Then the optimization problem is presented in Section 2 (e. g. its input, its objective, its constraints and its output). In Section 3, a formal definition of the problem is given. The network is described along with the two graphs which we use to model the LHPTP: the demand graph and the physical distance graph. Finally, a path-based MILP formulation for the LHPTP is introduced.

1 Application Framework

The LHPTP has a realistic basis as it comes from a case study provided by a postal company. However, we do not strictly enforce all their constraints in order to explore possibilities for finding improved solutions that are excluded by current restrictions on the network. Therefore the application framework we define here is a model which does not necessarily correspond exactly to today’s reality on the ground, it is inspired by it. We will discuss in Chapter 3 Section 2 which soft constraints we want to respect and which ones we want to release.

1.1 Long-Haul Transportation

Our objective is to optimize the transportation of parcels from their collection sites, known as sorting centers, to delivery depots. It is a **long-haul transportation** optimization problem. **Long-haul transportation** is defined as intercity transportation. It mainly concerns the movement of goods over relatively long distances, between terminals or cities. Goods may be moved by rail, truck, ship, etc., or any combination of modes [Cra03]. In our case, it is truck transportation.

In our optimization problem, we do not optimize the collection operations that consists in picking up the parcels and transporting them into the sorting centers which is the “first-mile” optimization problem. Nor do we address the distribution from the delivery depots to the final post offices which is the “last-mile” optimization problem. The Figure 2.1 distinguishes these three problems which are parts of parcel delivery. The wavy arrow represents the path we are optimizing, with zero to two sortings in sorting centers different than the origin one. Note that, the collection and the distribution parts can be modeled with the Vehicle Routing Problem, which is another well-known optimization problem.

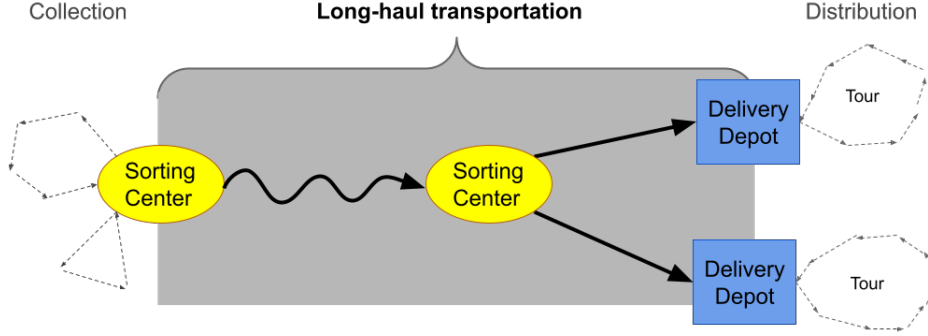


Figure 2.1: The postal network

1.2 Parcels and Demands

A **parcel** is a physical object which must be routed from a specified origin site to a designated delivery depot. We do not consider any volume nor weight for parcels, we assume all the parcels have the same size which is a global average size. This is a realistic assumption because each container can transport a significant amount of parcels. All the parcels which have the same origin and destination are grouped in a demand. We define a **demand** as an origin-destination-volume triple in which the volume is the number of parcels to deliver from the origin to the destination. We assume there are demands from all sorting centers to all delivery depots. Note that with this definition, there is only one demand for each sorting center to delivery depot pair. We do not currently focus on the demand forecast problem, since it is a distinct problem. As input, we are given demands for an average day. More details are given in Section 1.3 of Chapter 3.

1.3 Sites

In our network, we have two distinct sets of sites: each site is either a delivery depot or a sorting center but cannot be both.

The **sorting centers** are the starting points of demands in our network. Indeed, the parcels are collected from their senders and gathered in their local sorting centers. They are the place in which sorting (see Section 1.5) takes place and they are the departing points and the endpoints for vehicles.

The **delivery depots** are depots from which the parcels are dispatched to the post offices corresponding to their final destinations. They are the endpoints for parcels in our problem.

Among the sorting centers, some are also **inner-hubs** fixed by the transport managers. Inner-hubs are sorting centers which are on a path between two other sorting centers. Note that we do not restrict ourselves to the inner-hubs proposed by the transport managers as we sometimes consider all sorting centers as candidates to be inner-hubs.

We have around 15 sorting centers and 150 delivery depots. More details are given in Section 1.1 of Chapter 3.

Site Schedules

Both types of sites have opening hours. Sorting centers have three time slots in which the sorting operation can be performed (see Figure 2.2a). The first sorting (post-collection sorting) is mandatory and takes place before parcels leave their initial sorting center. A delivery depot can receive parcels during two time slots in the day that occur before the dispatching phases (see Figure 2.2b). Note that these times slots are not time windows: the vehicles arrive before the beginning of the time slot and leave the site at the end of the time slot.

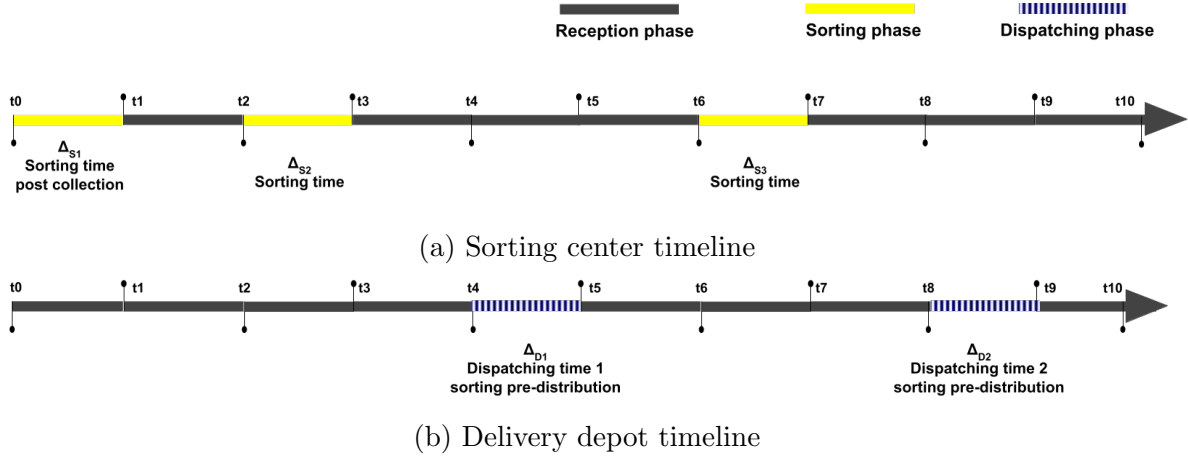


Figure 2.2: Daily operational organization

Catchment Area

In the LHPTP, we have a soft business constraint which assigns to each delivery depot an associated sorting center. It is usually the closest (with respect to physical distance) but not always as it has been determined by both history and transportation managers. The area which contains the delivery depots affiliated to a sorting center, including this sorting center, is called a **catchment area**. The sorting center to which a delivery depot is affiliated is called **associated sorting center**. This mapping is used in the current strategy in which the parcels are handled regionally: they are sent to the region containing their delivery depot, sorted in their associated sorting center and sent to their delivery depot.

If you already have posted a letter in France, you know that French mailboxes have two drop slots: one with the “region” you are in and the other called “other destinations”. The “region” you are in is the catchment area for the mail network you are in. The catchment area of the parcel network are not the same as the one for the mail network but this can give you the intuition of what is a catchment area.

1.4 Vehicles and Empty Repositioning

There are two types of **vehicles** we need to manage: trucks with one container and trucks with two containers (also called twin trailers). In our case study, one container can contain 2000 parcels. The trucks with two containers can contain twice the capacity of a container, in two separated containers. All sites can be served by the two types of vehicles and the choice of the more suitable ones on each transportation line is part of the decision problem. In our application framework, the number of vehicles or containers is not limited and can be sized appropriately in the optimization process to suit the needs as the postal company works with subcontractors. Indeed, we assume that these vehicles are rented, therefore the limit is their cost (which we aim to minimize), not their number. As all our vehicles are trucks, the travel time is the same for all the vehicles and does not depend on the vehicle type. The vehicle type changes only the capacity and the cost.

The trucks with two containers allow to deliver two sites in a row. This is called **double delivery**. The truck goes from the sorting center with parcels for depot 1 in the first container and parcel for depot 2 in the second container. It goes to depot 1 and distributes the parcels in the first container and then goes to depot 2 to distribute the parcels in the second container. It does not go through a sorting center between the two depots. The two delivery depots must be associated to the same sorting center.

As we design a daily transportation plan, we have to relocate the vehicles to be ready to for the next day. There is a need to send empty trucks back from delivery depots to sorting centers. This is called **empty repositioning**. Moreover, there is also a need to ensure that enough vehicles are available each day on each site to send the daily amount of parcels. This means that the number of outgoing and incoming trucks must be equal for each site in the course of a day. When some sorting centers send more parcels than what the delivery depots around them receive, there are more trucks going out of these sorting centers than trucks coming back from the delivery depots around them. Thus, there is a need to balance with empty trucks between sorting centers. The same issue exists for containers.

We have to remark that the vehicles are interchangeable. Indeed, we do not want to build a truck schedule, but a transportation plan (see Section 2.4).

1.5 Sorting and Consolidation

Sorting

In this case study, we deal with one logistics operation which takes place in the sorting centers: the **sorting**. When we sort parcels, all parcels in a container are grouped according to the next site served on their operational path. For instance, in Figure 2.3, the parcels in purple are for one destination and the parcels in yellow are for another destination. Note that, in the case of a truck with two containers, it is not possible to sort only one container. The two containers are sorted.

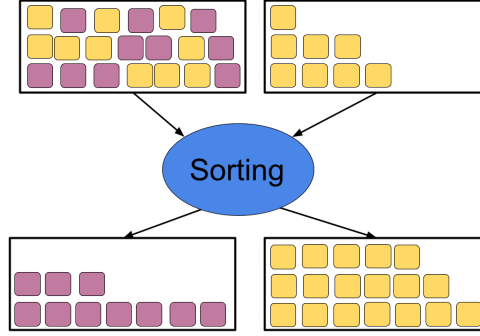


Figure 2.3: The sorting operation

The parcels can go through one to three sortings during their ride:

- **Sorting 1** - The first sorting allows to sort the freshly collected packages. It is mandatory and takes place in the starting sorting center. It is called **post-collection sorting**. In this application framework, it will take 3 hours. As this sorting is mandatory, we will not talk about it in the optimization process.
- **Sorting 2** - When parcels for different destinations are in the same truck or container, they have to be sorted a second time to be separated and sent to their respective destination. This second sorting always takes place in a sorting center. It takes 2 hours.
- **Hub sorting** - Before the sorting 2, an extra sorting can be done. But it can only be done in the inner-hubs (hence the name hub sorting). In this application framework, it will take 2 hours.

The time slots in which the sorting 2 and the hub sorting can take place are the same (see Figure 2.2a). Note that in this application framework, we do not consider any sorting capacity nor storage capacities in the sorting centers.

The constraint which states that the hub sorting is done in a inner-hub provided by the transport managers is a soft business constraint, as well as the constraint which states that the last sorting before the delivery is done in the sorting center associated to the delivery depot. We will discuss in Section 2 of Chapter 3 whether or not we will respect these constraints in our tests.

Consolidation

When a set of demands destined for different final sites are put in the same container it is called **consolidation**. For instance, if a sorting center B is between two sorting centers A and C , parcels from A for delivery depots 2 and 3 near B and delivery depots 4 and 5 near C can be put in a container from A to B . Then, in sorting center B , a sorting will separate the parcels for the delivery depots around C which will take a link from B to C

and the parcels for delivery depots near B which will be directly delivered from B (see Figure 2.4). In this case, demands from A to 2, 3, 4 and 5 are consolidated on the link from A to B . Demands from A to 4 and 5 are also consolidated on link from B to C and can be consolidated with demands from B to 4 and 5 on that link.

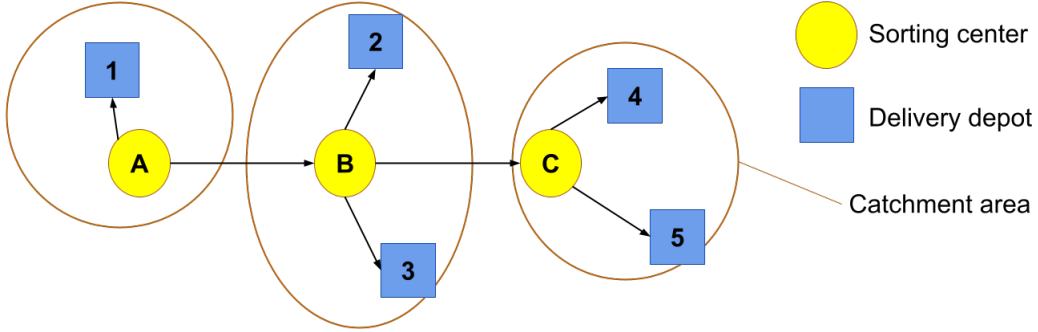


Figure 2.4: Consolidation of demands from A to 2, 3, 4 and 5

Consolidation of demands allows cost savings as it permits to better fill trucks and reduce their number on the network.

Bulk

All parcels are shipped in bulk in containers. Since a container cannot be partially unloaded, it carries parcels headed for one site, which is either an intermediate sorting center or a final delivery depot depending on the parcel's destination. This means that when there is consolidation between demands there is always a sorting operation which comes after: sorting centers are breakbulk terminals [Cra03].

1.6 Operational Paths

In the network, to go from a site to another, vehicles use links. Links can also be called arcs in a graph approach. We distinguish physical links from **operational links** which are associated with a schedule and a vehicle type (i.e., a vehicle with one or two containers). In the network, the links between sites (sorting centers or delivery depots) which are too long to be completed with respect to driving time regulations are removed from the network. Note that there is one special type of operational link which is not composed of one physical link: the double delivery. The double delivery is made via one link which starts in a sorting center and serves the two delivery depots successively. Thus technically it is constituted of two links, but this implementation trick avoids to consolidate parcels in a delivery depot which is forbidden. Note that the double delivery is allowed only in the same catchment area and that for two depots A and B , only one such double delivery is created (i.e., if the double delivery serves A and then B , then we do not allow a double delivery serving B and then A . We choose the shorter of the two possibilities in terms of travel time).

An **operational path** is a set of consecutive operational links between sites in which each site is associated with an operation (sorting or delivery) performed on a parcel flow at a time slot (see Figure 2.5).

In our problem, the time is represented by fifteen minutes intervals (e.g., the travel times are rounded at the next quarter of hour). The fixation of possible paths in terms of precedence force the parcels to arrive at a site before they leave it. It also forces parcels and trucks to wait until the end of the sorting operation to leave the sorting centers. According to the national regulation, drivers should make a break every 8 hours. This is directly taken into account when we create the set of possible operational paths.

As shown in Figure 2.5, each parcel can follow four types of operational path:

- a **direct path**: the parcel is sent directly from the sorting center to a delivery depot which is not in the catchment area of the origin,
- a **distribution path**: the parcel is sent directly from the sorting center to a delivery depot which is in the catchment area of the origin,
- a **path with one sorting**: the parcel undergoes a single sorting at the sorting center to which the delivery depot is associated (not necessarily an inner-hub),
- a **path with two sortings**: the parcel undergoes two sorting operations, one at an inner-hub and one at the sorting center to which its delivery depot is associated.

The constraint which states that the hub sorting is done in a inner-hub designed by the transport managers is a soft constraint: we can add new inner-hub if it is useful. The constraint which states that the last sorting before the delivery is done in the sorting center associated to the delivery depot is also a soft constraint: if it is better not to respect it, we can reject it. This will be discussed in Chapter 3 Section 2.

The objective of the LHPTP is to find at least one operational path for each demand in the transport network. This is what Leung et al. [LMS90] call *disaggregate shipping*: a demand can use multiple operational paths. The three possible paths format for parcels are represented by Figure 2.5. Note that the path types “direct path” and “distribution path” have the same format: They are made of only one operational link from a sorting center to a delivery depot. These links are called respectively **direct link** and **distribution link**.

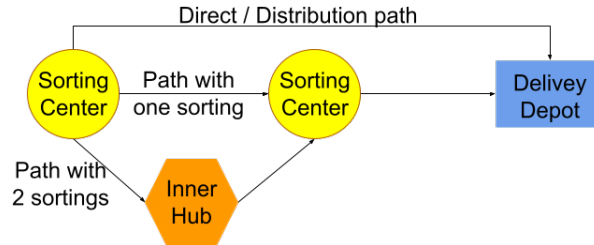
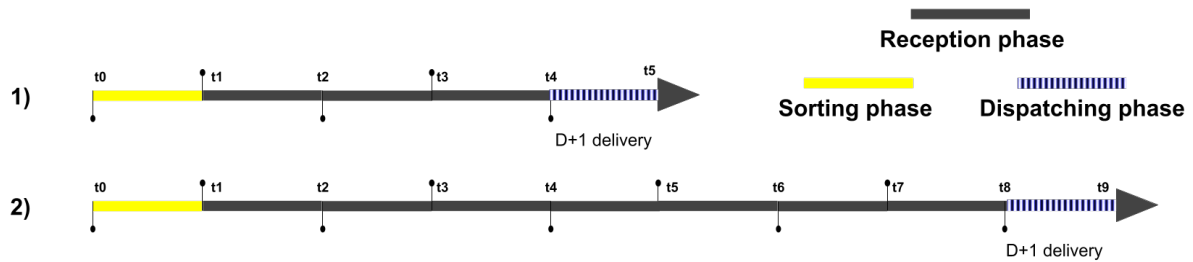


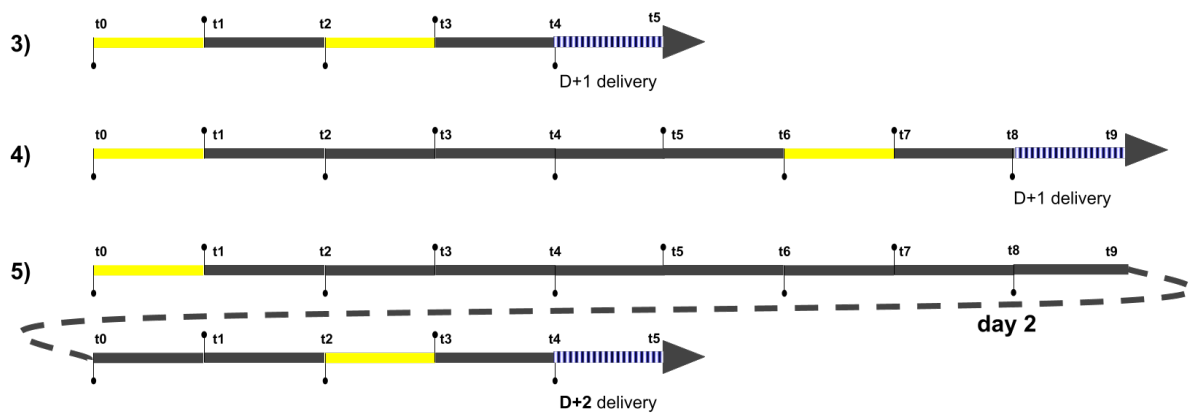
Figure 2.5: The three possible paths format for parcels

In Figure 2.5, the possibility of double delivery does not appear as it represents the paths from a parcel point of view. The double delivery concerns the truck point of view and therefore it does not appear here. Indeed, as more than one vehicle can be involved in the path of one parcel (it can change vehicle in a sorting center), the two aspects have to be distinguished. Note that as presented in Figure 2.5, each operational path is allowed to have at most two sorting operations after the first mandatory post-collection sorting.

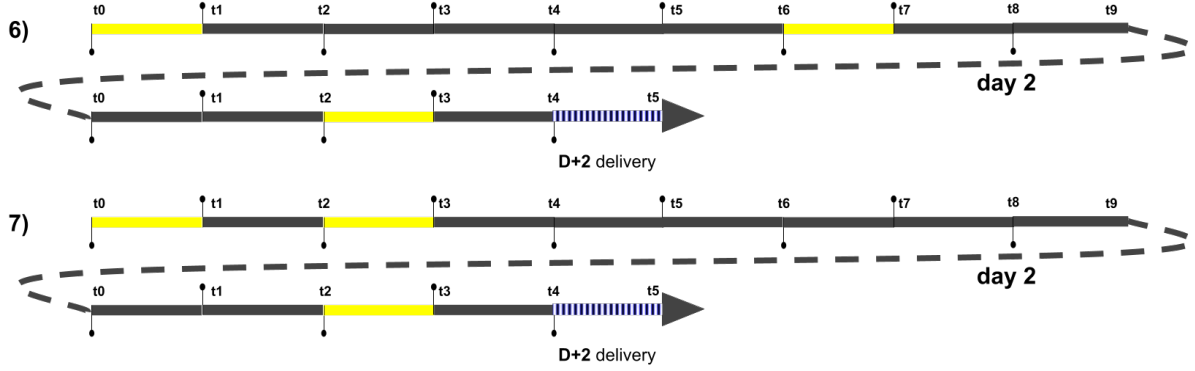
The paths can take one or two days to deliver the parcels. If the parcels are delivered in one day the path is called **D+1** as the delivery is done the day after the sending day D . The paths 1 to 4 in Figure 2.6 have a D+1 delivery. Otherwise it is a **D+2** delivery (like paths 5 to 7 in Figure 2.6). This figure represents all the path types allowed in the tests in this manuscript. It shows the use of the different timeslots for both types of sites. For instance, the paths 1 and 2 (see Figure 2.6a) differ because parcels using these paths do not arrive at the same time. The second path provides more time for transportation between sites than the first one and therefore starting from a fixed sorting center it allows to reach further delivery depots thus to reach more delivery depots.



(a) Direct or distribution paths



(b) Paths with one sorting



(c) Paths with two sortings

Figure 2.6: The possible paths types for a demand

In our case study, the paths 1 and 2 on Figure 2.6a are created according to the distance to go from the origin to the destination. If the vehicle can make the distance in the time allowed in path 1 then path 1 is created, otherwise, path 2 is created. Concerning Figure 2.6b, the same logic applies. When the path 3 can be done, paths 4 and 5 are not created. Otherwise if path 4 can be done, it is created and paths 3 and 5 are not. When the sites are too far apart to allow to use paths 3 and 4, path 5 is created. Paths 6 and 7 of Figure 2.6c are created according to the distance between the first and the second sorting center and the distance between the second and the last sorting center.

When trucks have delivered the parcels, they stop in delivery depots. They have to go back to a sorting center to be used the next day. This is called a **repositioning** as we said in Section 1.4.

2 Optimization Problem

In this section, we introduce the optimization problem of the Long-Haul Parcel Transportation Problem which takes place subject to the application framework described above.

2.1 Input: Parameters of the Optimization Problem

An input for the Long-Haul Parcel Transportation Problem is made of:

- a list of sites with their type, location and the travel times between each pair of sites,
- a list of demands (origin, destination, number of parcels) between these sites,
- the costs of all the logistics operations (sorting, transportation, entering sites, leaving sites, etc.).

The datasets of our case study will be introduced in Chapter 3 Section 1.

Example of Input on a Toy Problem

In this example, the demands and distances are as follows:

	Origin	Destination	Quantity
a	A	1	18
b	A	2	14
c	A	3	15
d	B	1	13
e	B	2	8
f	B	3	9

Table 2.1: Demands

Start \ End	A	B	1	2	3
A	0	460	108	427	495
B	464	0	520	98	80
1	109	516	0	/	/
2	430	99	/	0	/
3	501	80	/	/	0

Table 2.2: Distances

The graphs which represent Table 2.1 and Table 2.2 are introduced in Section 3.2. The output which corresponds to this input is in Section 2.4.

2.2 Objective

The objective of our problem is to minimize the delivery costs while delivering all the demands (see objective function (2.1a) of the MILP LHPTP-MILP given in Section 3.3). The cost of a solution is composed of:

- the transportation costs (tires, fuel, maintenance and toll),
- the salary expenses (salary and large displacement benefits),
- vehicle costs (insurance, taxes, etc...),
- unloading of containers costs,
- loading of containers costs,
- sorting costs (per parcels).

In practice, the salary expenses and the transportation costs are included in a kilometer cost, as it is directly linked to travel times. The other costs, which depends on the number of parcels and operations made on the parcels, are gathered in the cost of an operational path.

2.3 Constraints

Our model is composed of three major types of constraints: the delivery constraint, the capacity constraint and the design-balance constraint.

The main constraints are the **delivery constraints** (see constraints (2.1b)) which state that all the parcels of each demand have to be delivered.

Then, the **link capacity constraints** (see constraints (2.1c)) associate the number of parcels and the vehicles. It states that the number of parcels on each link is smaller

than the number that can be contained by the vehicles on this link. The capacities of the vehicles have to be respected but the number of vehicles on the links is not limited. There is a link capacity constraint for each vehicle type and each link in the network.

The **design-balance constraints** (see constraints (2.1d)) balance both containers and trucks as there is a constraint for each site for each vehicle type.

Two soft constraints are added to these three types of strong constraints: the inner-hubs and the catchment areas. Indeed, in the network provided, the hub sorting can be done only in an inner-hub fixed by a transportation manager but this is a soft constraint and we can add new inner-hubs. As for the catchment areas, the last sorting for parcels should be in the associated sorting center of the destination delivery depot. But this is a soft constraint and we can release it.

2.4 Output: Transportation Plan

With respect to the application framework described previously (see Section 1), our goal is to design a **transportation plan** which minimizes the costs for the postal company while delivering all the parcels in one or two days. A transportation plan is made of:

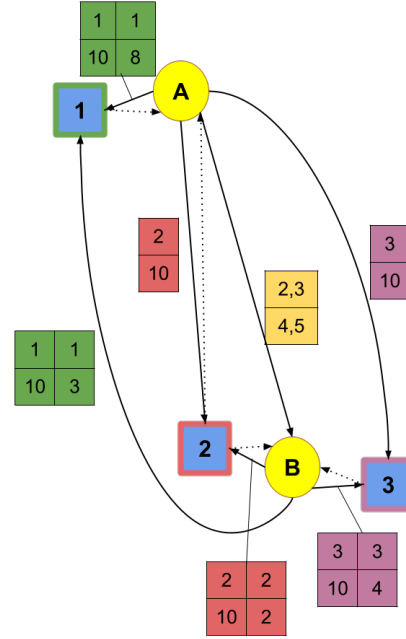
- at least one path for each demand (indeed, each demand can be split on multiple operational paths in the transportation plan),
- the number of vehicles of each type on each operational link,
- the total cost of the plan.

The key aspect here is that we have two flows to optimize simultaneously: the parcel flow and the truck flow.

Example of Output on a Toy Problem

Let us consider that the capacity for the truck is of 10 parcels. The optimal solution is represented below. In the picture a table is a truck with the two destinations of the two containers and the number of parcels in it. The dotted arrows represent the empty trucks sent back to sorting centers to balance both vehicles and containers for each site.

	Origin	Dest.	Path type
a	A	1	direct path
b	A	2	sorting at B and direct path
c	A	3	sorting at B and direct path
d	B	1	direct path
e	B	2	direct path
f	B	3	direct path



2.5 Key Performance Indicators

A transportation plan is characterized by indicators. These indicators include:

- the total cost of the transportation plan,
- the transportation costs of the transportation plan,
- the sorting costs of the transportation plan,
- the rate of parcels delivered inside the same catchment area in the transportation plan,
- the rate of parcels delivered on each type of path (direct, one sorting, two sortings) in the transportation plan,
- the rate of parcels delivered in one day or two days in the transportation plan,
- the filling rate of the trucks in the transportation plan,
- the filling rate of the trucks without counting the empty vehicles in the transportation plan,
- the number of operational links of each type (with a truck with one or two containers between sorting center and sorting center, sorting center and delivery depots etc...) used in the transportation plan,
- the number of constraints and variables of the model.

The objective of the optimization process is to propose transportation plans with all these characteristics to transportation managers to help them decide which one they want to implement in real life. According to the exchanges we had, the focus has been made on the minimization of the total cost, which is the objective we will pursue in the rest of this manuscript. That is why we will present the total cost of the transportation plan and the optimality gap (defined in Chapter 1 Section 4.2) each time we will present results. But some indicators remain critical like the filling rate of the trucks which is what the operational people see. Thus it impacts them psychologically as no one likes to send empty trucks. Therefore we will also discuss this indicator in the results. Finally, the last point observed by the transportation managers is the carbon cost of the transportation plan. It is computed from the carbon footprint which is computed from the total distance made by the trucks, the fuel consumption and the emission coefficient. We will not especially study the carbon cost in this manuscript as minimizing the transportation costs minimizes the carbon cost of the transportation plan.

3 Formal Definition of the LHPTP

In this Section we formally define the LHPTP described previously by presenting the network on which we optimize the long-haul transportation and the graphs and notations which model it. Finally, we introduce an MILP for this problem.

3.1 Network Provided

The parcels are delivered on a road network composed of two hierarchically nested hub-and-spoke networks (see Figure 2.7). The inner level of this network is made of sorting centers. The outer level of the network is composed of sorting centers and delivery depots.

The inner network, which connects the sorting centers, is a hybrid hub-and-spoke network (see Figure 2.7a). Indeed, the sorting centers either send parcels directly to other sorting centers or they can use an inner-hub to sort and/or consolidate demands in a hub sorting (see Section 1.5). So it constitutes a hybrid hub-and-spoke network in which the inner-hub are the hubs while the sorting centers are the spokes. Note that the inner-hubs are also sorting centers so they might be the spokes of other inner-hubs.

The outer network appears when we add the final destinations. The final destinations of parcels are the delivery depots (D-D on Figure 2.7b) which can be seen as the spokes in a hybrid hub-and-spoke network in which the outer-hubs would be the associated sorting centers. Indeed, parcels can be either sent directly from the origin sorting center or consolidated and then sorted at a sorting center before reaching their delivery depots.

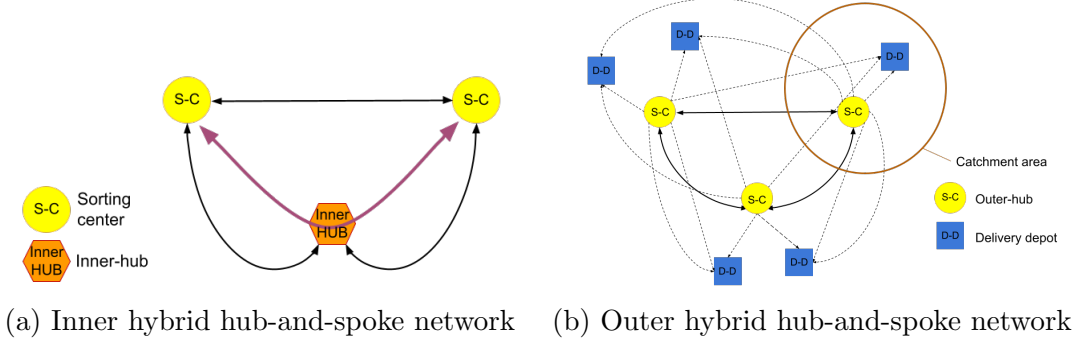


Figure 2.7: The two levels of the hybrid hub-and-spoke network

3.2 Graph Definition

To formally define our optimization problem we need two graphs, depicted in Figure 2.8. The first graph is the **demand graph** which is a directed bipartite graph with all arcs directed from sorting centers to delivery depots. The second graph is the **physical distance graph**. This is a directed graph that has two arcs in each direction between (i) each pair of sorting centers and (ii) each pair of sorting center and delivery depot. Each arc from point a to point b represents the real-life distance for a vehicle to travel from a to b . These graphs are represented as matrices and sometimes we refer to the **demand matrix** or **physical distance matrix**. We will use D^{dist} to refer to the physical distance matrix. An arc that connects any two points in the graph corresponds to a **physical link**. Note that the physical distance between two sites can slightly differ if it is the distance from a to b or from b to a as it is the distance in the road network, respecting the speed limits.

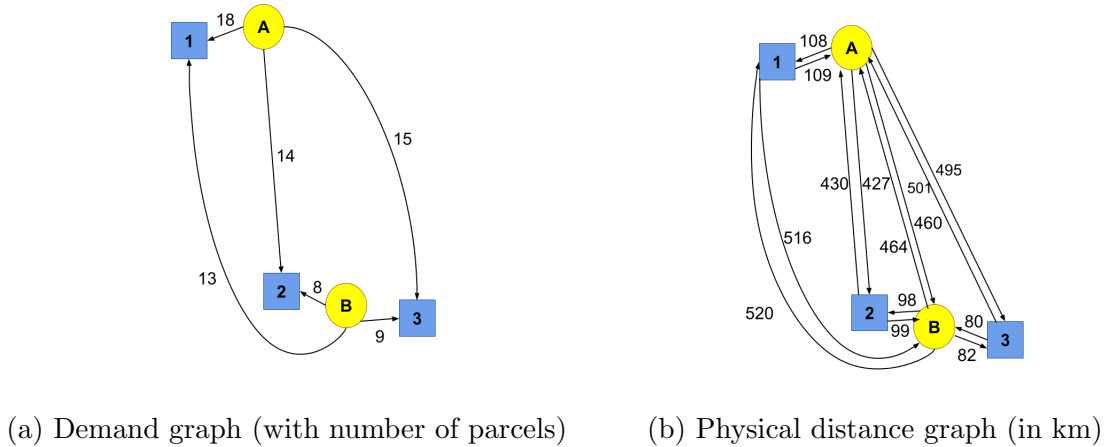


Figure 2.8: Examples of graphs

3.3 Path-Based MILP Formulation

We introduce here the MILP formulation which can solve the LHPTP in reasonable time when the size of the input is small enough. We propose a path-based model as the definition of parcel paths by operational people (see Section 1.6) was based on possible paths. Thus we design our model in a way which allow to easily monitor the possible paths we propose to the solver, especially when it comes to adding and/or removing some path types, in order to facilitates the exchanges with the transportation managers.

Note that an arc-based formulation has been provided by Meisen [Mei15] for a problem very close to the LHPTP (the differences between this problem and the LHPTP can be found in Table 1.1 in Section 1 of Chapter 1). He models his problem with a time-space network [ACC09a]. In this type of network, the vertices are replicated in each time period which increases the size of the network in the same way that the number of paths increase in a path-based model. In the rest of this manuscript, we will only use the path-based model.

Before introducing our MILP, let us define some notations:

- D is the set of demands,
- S is the set of sites composed of $S_{s,c}$ the set of sorting centers and $S_{d,d}$ the set of delivery depots,
- $S_{s,c}$ the set of sorting centers (origins of parcels) is partitioned into “regular” sorting centers plus “inner-hubs”,
- $S_{d,d}$ the set of delivery depots (destinations of parcels),
- L is the set of links between sites whose element $l_{i,j}$ is the link between site i and site j ,
- P_d is the set of possible operational paths for the demand d in D ,
- P_d^l is the set of possible operational paths using the link l in L for demand d in D ,
- V is the set of vehicle types.

The variables are of two types, for parcel flows and vehicle flows:

- $x_p^d \in [0, 1]$, with $d \in D$ and $p \in P_d$, represent parcel flows. It is the percentage of a demand d using an operational path p (geographical links combination with a specific vehicle and sorting operations at fixed dates).
- $y_l^{veh} \in \mathbb{N}$, with $l \in L$ and $veh \in V$, represent vehicle flows. They are integers which represent the number of vehicles of type veh on each operational link l between two sites.

Let us also consider:

- c_p the cost of using the operational path p (it includes the costs of loading, unloading and sorting),
- c_l^{veh} the cost of using the operational link l with vehicle veh (it includes the salary, transportation and vehicle costs),
- v_d the volume (number of parcels) of demand d ,
- C_{veh} the capacity of vehicle veh .

With these notations, our model, denoted LHPTP-MILP, can be described as follows:

$$\left\{ \begin{array}{ll} \min \sum_{d \in D} \sum_{p \in P_d} c_p x_p^d + \sum_{veh \in V} \sum_{l \in L} c_l^{veh} y_l^{veh} & (2.1a) \\ s.t. : & \\ \quad \forall d \in D, \sum_{p \in P_d} x_p^d = 1 & (2.1b) \\ \quad \forall l \in L, \forall veh \in V, \sum_{d \in D} \sum_{p \in P_d^l} v_d x_p^d \leq y_l^{veh} C_{veh} & (2.1c) \\ \quad \forall s \in S, \forall veh \in V, \sum_{i \in S} y_{l_{i,s}}^{veh} = \sum_{j \in S} y_{l_{s,j}}^{veh} & (2.1d) \\ \text{with: } x_p^d \in [0, 1] & (2.1e) \\ \quad y_l^{veh} \in \mathbb{N} & (2.1f) \end{array} \right.$$

LHPTP-MILP

4 Conclusion

In this chapter, we introduce the Long-Haul Parcel Transportation Problem (LHPTP). We present its application framework, with all the notions and constraints inherent to this case study. These notions include the demands, the sorting operation, the empty repositioning of vehicles, the consolidation of demands, and the operational paths (note that the key words of the manuscript are defined in the Glossary 2). We develop the optimization problem: its input, its objective, its constraints and its output. Given the sites (and their types), the distance and travel time between them, the demands, the capacity of vehicles, the costs of transportation and sorting, the goal is to design a daily transportation plan which allow to deliver all the demands from their origin sorting center to their destination delivery depots. The transportation plan has to respect the constraints of the problem: the delivery constraint, the capacity constraint and the design-balance constraint. Moreover, there are two soft business constraints: the hub-sorting

has to take place in the inner-hubs proposed by the transportation managers, and the last sorting has to be done inside the catchment area of the destination. Finally, we give a formal definition of the LHPTP. We described the two-level hub-and-spoke network on which the LHPTP takes place. We present the graphs which model the problem and propose an MILP formulation to optimize it. Note that this MILP can be adapted to solve other long-haul parcel transportation problems (e.g., with different vehicle types, without vehicle balancing, or with sorting capacity constraints, etc.).

Since a container can carry parcels with different origins and destinations, an optimal solution might not route each demand on its shortest path. Thus our parcel transportation problem is not simply a shortest path problem due to the usage of consolidation. Indeed, consolidation is a crucial strategy we use to reduce costs, but the usage of consolidation makes the problem computationally difficult.

Different sorting centers and different time slots can be used for the sorting operations. As we add the possibility of disaggregate shipping, there is a combinatorial explosion on the number of possible operational paths, even though this number is bounded, which prevent the MILP to give an optimal solution on realistic sized datasets. These realistic datasets and the combinatorial explosion are described in the next Chapter.

Chapter 3

Case Study Analysis

Contents

1	Datasets	54
1.1	Content of the Datasets	55
1.2	Configuration of Sites	56
1.3	Demand Profile	57
1.4	Distances in the Network	59
2	Optimization Strategies for Using the Network	61
2.1	Computational Aspects	62
2.2	Current Operational Strategy	64
2.3	Addition of New Paths to the Current Operational Strategy	65
2.4	Impact of the Soft Business Constraints on the Size of the MILP Formulation	67
2.5	Synthesis	69
3	Exact Approach: Direct Solving with MILP	69
3.1	Application of the MILP on the LHPTP	70
3.2	Limits of LHPTP-MILP on our Case Study	73
4	Conclusion	74

In this chapter, we present and analyze the case study and accompanying datasets on which we test our algorithms. These datasets have been provided to us by a postal company. We discuss the structure of these datasets on which we will solve the LHPTP. These datasets span a six year period and thus they demonstrate the evolution of the postal network. On these datasets, we found that the LHPTP-MILP yields feasible solutions but their quality depends heavily on the optimization strategy we use (e.g., the soft constraints we use or not, the possible paths that we propose or not etc.)

In Section 1, we present and discuss the datasets. Then, in Section 2, we discuss the impact of the two soft constraints introduced in the previous Chapter (Chapter 2) on the size of the MILP formulation. As a reminder, the first soft constraint is about doing the sorting in the sorting center associated to the delivery depot while the second one is related to using only the inner-hubs proposed by the transportation managers to do the hub sorting. In Section 3, we discuss the results obtained when solving the MILP only using an off-the-shelf solver. This motivates the objective of obtaining a better solution that goes beyond the limits of what can be achieved only using an MILP solver as a blackbox.

The model presented in this chapter has been presented at the ROADEF 2019 conference [GCH⁺19].

1 Datasets

In this section, we introduce the datasets used in this case study. We present exactly what data is contained in a dataset: For example, the sites, the catchments areas, the demands, etc. Moreover, we discuss their structural properties such as how the sites are geographically distributed, the balance of the demands, the size of the demands relative to container size, the configuration of the catchment areas, etc.

The actual data used for our tests are data provided by a postal company. They are based on midterm forecasts made from field data a few years ago. These demand forecasts are constructed by averaging daily data and the sites (e.g., sorting sites, catchment areas, delivery depots) are those which are projected to be the most practical for the time period covered by the demand forecast. Indeed the forecast was conducted before the pandemic, which strongly impacted the volumes of parcels sent in France, and exactly how this will affect the future demand has not yet been established. Nevertheless, the datasets we work with contain useful information about demand forecasts with respect to growth trends, balance, etc. These data provide instances of an accurate size with key characteristics of an actual network in terms of parcel volume and growth albeit created before the pandemic.

1.1 Content of the Datasets

We have six datasets which represent six different configurations of the network with logistics sites spread across mainland France. By configuration, we refer to the physical layout of the network, types of sites, etc. Properties of these configurations are described in Table 3.1. A dataset consists of:

- a list of sites with their type, location and the travel times and distances between each pair of sites,
- a list of demands (origin, destination, number of parcels) between these sites,
- the costs of all the logistics operations (sorting, entering sites, leaving sites, etc.).

Each of these configurations represents a step in the evolution of the network. This planned evolution of the network over time was made to meet the growing demand and the potential growing number of parcels to sort. We see a growing number of inner-hubs, which is sized to suit the needs. The number of depots decreases as the network is modernized and the old sites are replaced by fewer but larger sites. These six configurations considered together depict the evolution of the network as it was planned based on predictions, although it does not mean it will be applied as is.

Table 3.1: Datasets

Name of the dataset	Number of sites	Number of delivery depots	Number of sorting centers	Number of inner-hubs	Number of demands
Config. 1	292	279	13	2	3627
Config. 2	263	248	15	3	3720
Config. 3	256	239	17	4	4063
Config. 4	212	195	17	4	3120
Config. 5	173	156	17	4	2496
Config. 6	154	136	18	4	2312

For the first three configurations, all the sorting centers send parcels to all the delivery depots, thus the number of demands is $N_{\text{sorting centers}} \times N_{\text{delivery depots}}$. For the last three configurations, there is one sorting center which does not send parcels and is used only for sorting and hub sorting ¹. Its purpose is to allow for more consolidation of demands thanks to the sorting and hub sorting as it is an inner-hub proposed by the transportation managers. Note that in Table 3.1, the inner-hubs are counted twice: They are counted as sorting centers and as inner-hubs.

We will not present the costs of any operation here for confidentiality reasons, but as an indication, we can say that the kilometric cost for a vehicle is thrice the cost to sort one parcel in a sorting center. This means that sorting three parcels costs the same as routing one truck which can contain up to 2000 parcels for one kilometer. Routing

¹defined in Chapter 2 Section 1.5, it is the first of the two sortings in a path with two sortings

a truck on 527 km (the average distance to make for a demand) costs the same as sorting 1581 parcels, while the average number of parcels in a demand is 402 (figures for Configuration 6, see Sections 1.3 and 1.4). These costs encourage sending non-full trucks on short distance when it avoids sorting parcels. But on long distances, it encourages to consolidate demands more (even if this implies sorting) which increases the filling rate and minimizes the number of trucks. This hypothesis will be discussed in the results sections of the heuristics chapters (Section 4.2 of Chapter 4, and Section 3.2 Chapter 5). We will test other values for these parameters, which will potentially impact, even invert, the decisions taken in the optimized transportation plan.

For the rest of this chapter, the illustrations will be provided with respect to Configuration 6 as it is the smallest configuration and therefore the easiest to visualize.

1.2 Configuration of Sites

The sites are spread across the country. Figure 3.1 represents the sites in France for Configuration 6. The sorting centers are in blue and the delivery depots are in red.

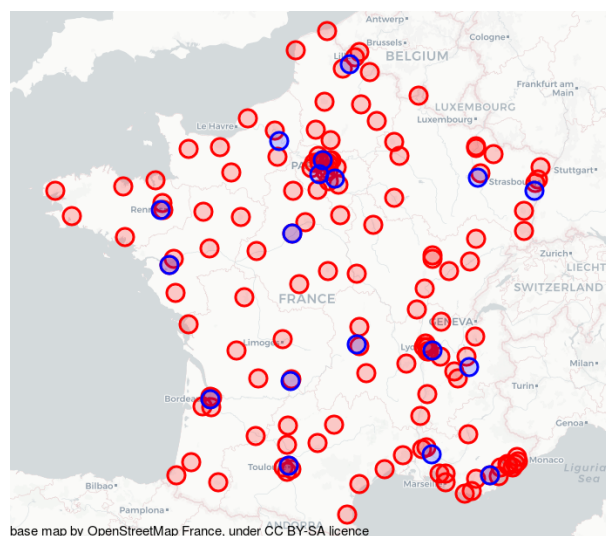


Figure 3.1: The sites in Configuration 6

Figure 3.2 represents the sites in France colored by catchment areas for Configuration 6. The sorting centers are represented by the larger, fully colored circles, while the delivery depots are smaller and lightly colored. We can see on Figure 3.2 that the distribution of sites in the catchment areas are based on distances to the associated sorting center, but also on travel times, which may be a function of the topography and the existing road network. The catchment areas are also based on other geographic considerations such as regional groupings of depots and volume considerations: The grouping of sites takes into consideration the sorting capacities of sorting centers.

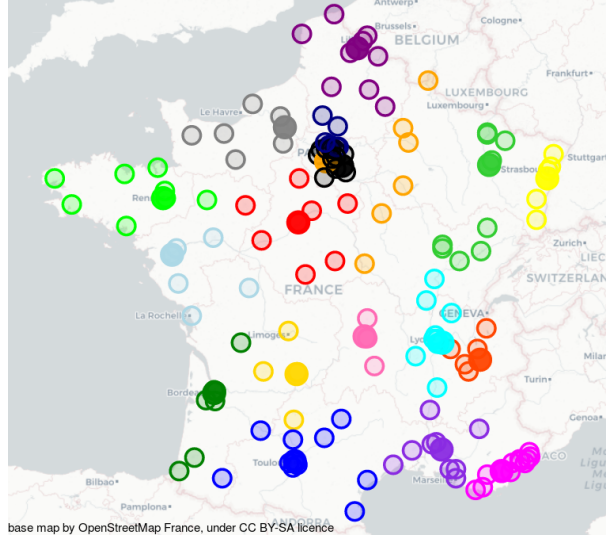


Figure 3.2: The catchment areas in Configuration 6

1.3 Demand Profile

We can see in Figure 3.3 a bar chart of demands ² according to the number of parcels they contain for the six configurations. The average demand volume (per demand) is 402 (mean of the means of the six configurations). The mean demand volume of the medians of the six datasets is 225 parcels. The mean of the standard deviations is 544 parcels.

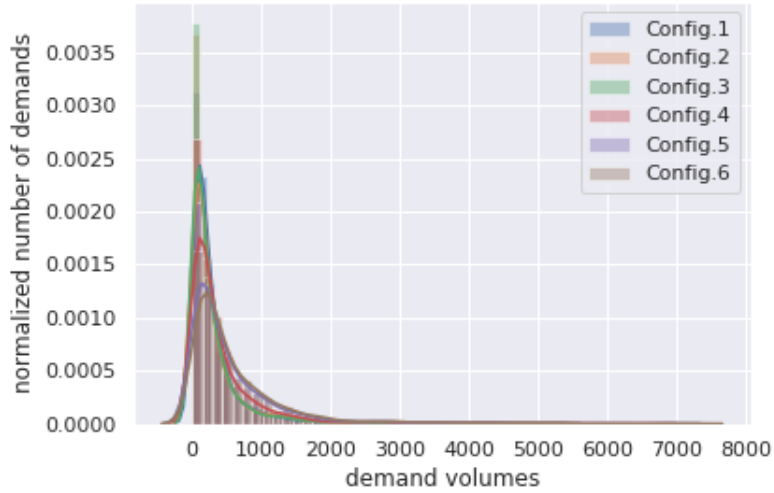


Figure 3.3: Distribution of demands with respect to their volume³

²A demand is an (origin, destination, number of parcels) triple, see chapter 2 for details.

³the volume of a demand is the number of parcels in this demand

In this case study, one container can contain 2000 parcels. We clearly see that most of the demands are under this value. This means that consolidation will be useful as most of demands do not fill a truck. Table 3.2 gives the number of demands which fill 60% and 100% of a container (which corresponds respectively to 1200 parcels and to 2000 parcels). We see it represents a small number of the total of demands.

Table 3.2: Number of large demands

Name of the dataset	Number of sites	Number of demands	Number of demands ≥ 1200 parcels (60%)	Number of demands ≥ 2000 parcels (100%)
Config. 1	292	3627	135 (3.72%)	42 (1.15%)
Config. 2	263	3720	167 (4.49%)	56 (1.51%)
Config. 3	256	4063	169 (4.16%)	53 (1.30%)
Config. 4	212	3120	209 (6.67%)	70 (2.24%)
Config. 5	173	2496	252 (10.1%)	81 (3.25%)
Config. 6	154	2312	278 (12.0%)	89 (3.85%)

We see that the distributions of demands in the six configurations have similar shapes with respect to their volume. We can see in Figure 3.4 that the difference between the configurations in terms of demands is the total number of parcels (i.e., total demand volume). Note that there is no contradiction with the number of demands decreasing in Table 1: There are less origin-destination pairs but the total number of parcels between these pairs grows. Indeed, the number of parcels was predicted to grow with an increase of 4.6% by year and this is what the dataset is based on. In practice, this growth was a bit higher (+6.6% for 2018 and +8.3% for 2019) for domestic parcels for all postal companies in France⁴.

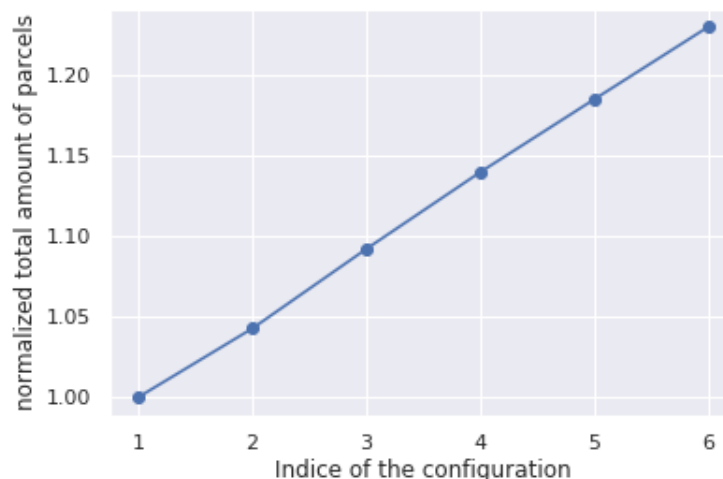


Figure 3.4: Relative growth of the number of parcels sent in the network

⁴ARCEP "Les marchés du courrier, du colis et des activités connexes en France" ISSN n°2258-3106

1.4 Distances in the Network

In this section, we will discuss the distances related to parcels and demands. Notice that parcels and demands are two different things. The number of parcels is the total demand volume, while a demand is an origin destination pair associated to a volume. Meanwhile, the number of demands is the number of origin destination pairs. The distance of a demand or a parcel is the distance between its origin and its destination. Recall that there are demands between all the pair of sites, but of various volumes and therefore, in some sense, we have two points of view: The number of parcels which has an economic meaning, and the number of demands which impacts the size of the model and therefore has consequences on the optimization obtained via an MILP.

First, let us discuss the distribution of distances related to demands (not parcels) for the six configurations. Figure 3.5 represents the distribution of distances when we count once each demand. Note that as we have demands between all sorting center delivery depot pairs, it is also the distribution of distances between the sorting center delivery depot pairs. The mean of the distances averaged over all demands (not parcels) is 526.97 km (mean of the means of the six configurations). The mean of the medians of the six configurations is 516.08 km. The mean of the standard deviations of the six configurations is 261.69 km. These values show that the sites are well-spread across the country.

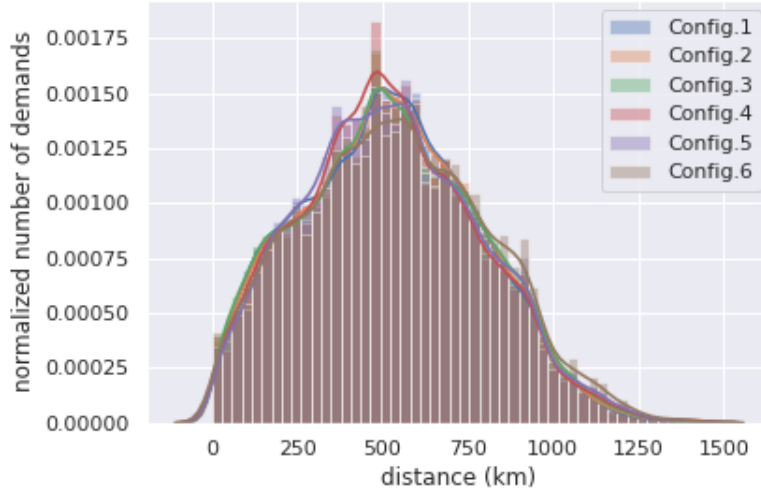


Figure 3.5: Distribution of the distance (in km) for the demands

Now, we can see in Figure 3.6, the distribution of distances corresponding to parcels (not demands) for the six configurations. We see that many parcels are being sent on a short distance. The mean of the distances averaged over all parcels (not demands) is 430.17 km (mean of the means of the six configurations). The mean of the medians of the six configurations is 420.10 km. The mean of the standard deviations of the six configurations is 509.35 km. These values also show that the sites are well-spread across the country.

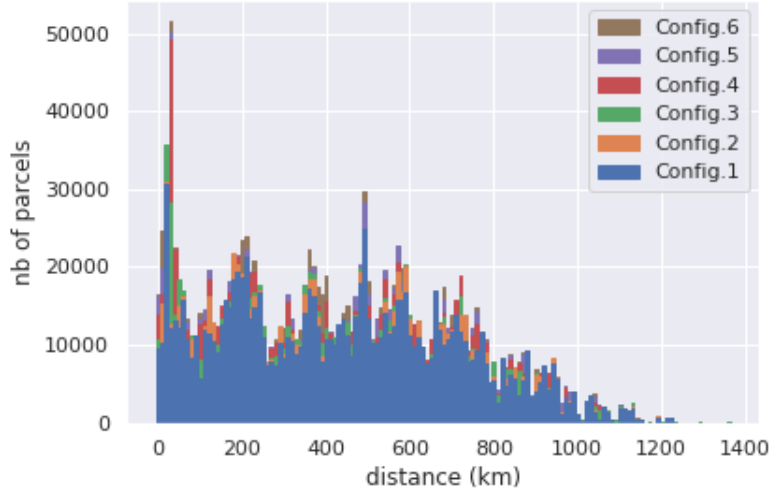


Figure 3.6: Number of parcels as a function of the distance

On average, 28% of the demands (not parcels) are sent inside the catchment areas (see Table 3.3). For the parcels, this varies from 14.9% to 8.5% (see Table 3.3). These demands go directly from the sorting center of the catchment area to an associated delivery depot. For these demands, the optimization decisions are the vehicles used, the consolidation with demands from other sites if they arrive soon enough and the combinations of delivery depots in the case of a double delivery. The percentage of demands which stays inside a single catchment area is quite stable as it is the number of demand pairs which corresponds to a criterion which depends on the type of site. It means that the catchment areas have been designed to spread consistently the delivery depots in the catchment areas. Indeed, when designing these catchment areas, there is a willingness to distribute the volumes to sort in order to respect the sorting capacities of the sorting centers. However, the number of parcels inside a single catchment area clearly decreases, while the total amount of parcels sent increases (see Figure 3.4). But the number of parcels increases more for demands which change catchment area. Indeed, new large volumes (see Table 3.2) appear on some sites where there are big customers (plants or warehouses) which send their parcels thanks to our network. This leads to a deformation of the parcel flows (not demand flows) which are not well balanced.

Table 3.3: Number of demands delivered inside a catchment area

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of dem. in the c. area	Demands in the c. area	Parcels in the c. area
Config. 1	292	3627	1084	29.9%	14.9%
Config. 2	263	3720	1078	29.0%	13.6%
Config. 3	256	4063	1064	26.2%	11.7%
Config. 4	212	3120	932	29.9%	8.8%
Config. 5	173	2496	758	30.4%	8.9%
Config. 6	154	2312	581	25.1%	8.5%

We said that all sorting centers send parcels to all delivery depots, but these parcel flows are not balanced. Indeed, in some areas there are industrial customers who send their parcels via our network to the richer and more populated regions of the country. Indeed, the spectral clustering (defined in Section 1 of Chapter 4) applied on our demand volumes give us Figure 3.7. The input graph for this clustering algorithm is the graph whose vertices are all sites and in which edges are weighted by the parcel flow between each pair of sites. Note the parcel flow is directed from sorting center to delivery depot, but we give the same value to the delivery depot to sorting center pair as we needed a symmetric matrix to apply the clustering algorithm. The objective of the spectral clustering is to build two clusters with little exchanges between them. Therefore the Figure 3.7 shows that there are more parcels exchanged inside the blue area and inside the red area than between the blue and red areas.

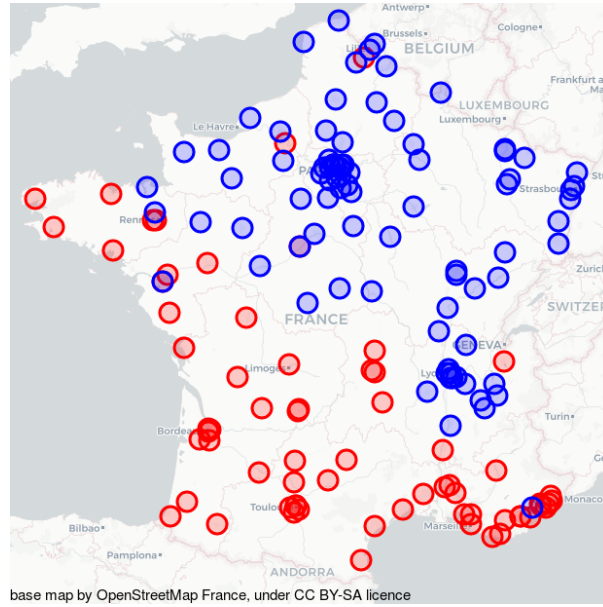


Figure 3.7: Spectral clustering on demand volumes

In this section, we introduced the datasets on which we will test our algorithms and some of their specific properties. Indeed, the same algorithms applied on the same problem (the LHPTP) but with an input with other defining traits (for instance larger demand volumes) might behave differently.

2 Optimization Strategies for Using the Network

As we said in Section 2.3 of Chapter 2, in this application framework, there are two soft constraints: The last sorting has to respect the catchment area and the inner-hubs have to be the ones provided by the transportation managers. By **optimization strategy**, we mean the set of operational paths offered to the MILP solver. This set depends on the soft business constraints that are imposed. The number of operational paths affects

the size of the input to the MILP solver (or to our algorithms) and the quality of the output (i.e., restricting the paths too much could lead to a poor quality solution). Thus, our main challenge is to keep the input small (to have a manageable problem size) while choosing the set of operational paths in such a way as to still have a good quality solution.

In this Section, we first explain how the number of operational paths increases from a computational point of view (e.g., linear, quadratic, exponential, etc.) and then we introduce the current operational strategy. Next, we introduce new additional types of operational paths that we can use (and could be included in the output transportation plan if chosen by the MILP or the algorithms): Direct paths, repositioning the trucks without respecting the catchment areas, hub sorting paths. Then we discuss the problem size when we respect or ignore the two soft constraints. The problem size is critical when giving an MILP to an MILP solver as we know that if there are many variables the optimization (in reasonable time) is compromised. Note that, as presented in Section 3.3 of Chapter 2, the link variables are integers while the paths variables are floats. Finally, we introduce our new optimization strategies, which depend on which operational paths we select for the input.

2.1 Computational Aspects

The LHPTP-MILP is very large when considering a problem instance that does not respect either of the two business constraints. Using a complete data instance with all the sites in the country and all the possible operational paths, the optimal solution of the LHPTP-MILP would represent the optimal solution to the global problem. However, if we run the LHPTP-MILP on such an instance, it does not provide good solutions in reasonable time (see Section 3.1). One reason for this is that there are too many paths for a standard MILP-solver to handle the problem well. As shown in Figure 2.6 of Chapter 2 Section 1.6, there are many possible operational paths for each demand and these paths can use various intermediate sites. Furthermore, on these operational paths, the demands can be combined in trucks in various ways thus the problem size grows very quickly.

Number of Operational Links

If we denote N_{sc} the number of sorting centers and N_{dd} the number of delivery depots, the number of operational links without considering double delivery is

$$20N_{sc}^2 + 4N_{sc} \cdot N_{dd} \quad (3.1)$$

In this number, we have the number of links between sorting centers, plus the number of links between sorting center and delivery depot. The $20 = 2 \cdot 2 \cdot 5$ represents the options for the vehicle on each link (one or two containers), the direction of the link, and the timeslots (before or after each sorting and hub-sorting). The delivery depot to sorting center links are called repositioning links while the sorting center to delivery depot links are the one used for direct paths and might therefore be called direct link. The $4 = 2 \cdot 2$ works the same except that there is only one timeslot for which the link is actually created instead of five. Indeed, only the shortest direct link is created.

The double delivery is made via one link which starts in a sorting center and serves the two delivery depots successively. Note that the double delivery is allowed only in the same catchment area and that for two depots A and B , only one such double delivery is created (i.e., if the double delivery serves A and then B , then we do not allow a double delivery serving B and then A . We choose the shorter of the two possibilities in terms of travel time). The number of links of this type is

$$N_{sc} \cdot N_{dd} \cdot N_{dd*}$$

where N_{dd*} is the number of delivery depots which are in the same catchment area as the previous delivery depot. But we also need to recall that the double delivery can be done only with a truck with two containers, and there is one direction only as it is a delivery, there are two timeslots, but we cut symmetries so we need to divide by two hence there is a factor one. Moreover, this is an order of magnitude as if a link is too long (for the parcels to reach the second delivery depots before it closes) it is not created.

Therefore the order of magnitude of the number of operational links is:

$$20N_{sc}^2 + 4N_{sc} \cdot N_{dd} + N_{sc} \cdot N_{dd} \cdot N_{dd*} \quad (3.2)$$

Number of Operational Paths

If we denote by N_{sc} the number of sorting centers and by N_{dd} the number of delivery depots, the number of operational paths without considering double delivery is:

$$2 \cdot N_{sc} \cdot N_{dd} + 2^2 \cdot N_{sc} \cdot (N_{sc} - 1) \cdot N_{dd} + 2^3 \cdot N_{sc} \cdot (N_{sc} - 1) \cdot (N_{sc} - 2) \cdot N_{dd} \quad (3.3)$$

$$= (8N_{sc}^3 - 16N_{sc}^2 + 18N_{sc}) \cdot N_{dd}$$

This quantity contains the number of direct paths, paths with one sorting and paths with two sortings. The power of two represents the number of options for the vehicle on each link (one or two containers). Note that, we are counting paths for parcels, so each path has only one direction. The operational path formats which take into account the timeslots combinations for the opening hours of the sites are showed in Figure 2.6 in Chapter 2 Section 1.6. Notice that in (3.3), we actually include operational paths that are not possible for precedence reasons or too long. Thus, (3.3) is an overestimate by a constant factor on the number of possible operational paths. Nevertheless, the quantity in (3.3) is asymptotically correct.

If we consider the double delivery, we need to replace N_{dd} by $N_{dd} \cdot (1 + N_{dd*})$ where N_{dd*} is the number of delivery depots which are in the same catchment area as the first delivery depot. But we also need to recall that the double delivery can be done only with a truck with two containers on the last link, so the leading constant is smaller by a factor of two. Therefore the number of operational paths is:

$$(16N_{sc}^3 - 32N_{sc}^2 + 36N_{sc}) \cdot N_{dd} + (8N_{sc}^3 - 16N_{sc}^2 + 18N_{sc}) \cdot N_{dd} \cdot N_{dd*}, \quad (3.4)$$

with N_{dd*} the number of delivery depots which can be delivered in the double delivery. Here again, this quantity contains the number of operational paths that are not possible for precedence reasons or too long, so (3.4) is an overestimate, but it is the correct order of magnitude.

2.2 Current Operational Strategy

In the **current operational strategy**, the parcels are handled regionally, without hub sorting nor direct paths. All parcels follow a path with one sorting, in the sorting center of the catchment area of the destination. To be more precise, the parcels whose origins are in the same catchment area are gathered in the corresponding sorting center. The gathering of parcels from their shipping site to each sorting center (i.e., the "first-mile delivery problem") will not be detailed here as it has not been integrated in the problem expression. In a sorting center, the parcels are sorted by catchment area of destination, or sent directly to their destination if the destination is inside the same catchment area. After this first sorting – the post-collection sorting (see Section 1.5 of Chapter 2) – the parcels are sent to the sorting center of the catchment area of the delivery depot they are sent to. In each of these sorting centers, parcels are then sorted again to be gathered and sent to their destination delivery depot. In this final delivery depot, the parcels are sorted at a more precise level and sent to the post office they depend on but we do not deal with this last shipment (i.e., this is the "last-mile delivery problem"). The trucks in the delivery depots are returned to the sorting center to which the delivery depot is associated. The balance is then made between sorting centers. The Figure 3.8 represents the current operational strategy from a network point of view while the Figure 1 of the Introduction represents it from a parcel point of view.

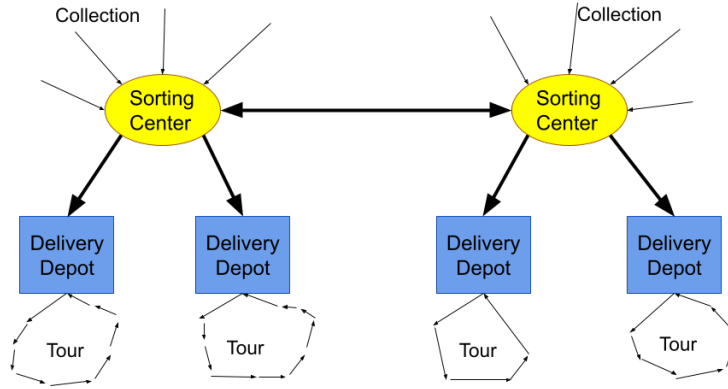


Figure 3.8: The current strategy to deliver parcels

In terms of variables, as this strategy involves neither direct paths nor repositionings outside the catchment area, the number of operational links is quite small (see Table 3.4). However, the number of operational paths does not seem particularly small as the combinations of links for trucks with one or two containers have to be considered.

Table 3.4: MILP variables of the current operational strategy

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Nb. of variables	Nb. of constraints
Config. 1	292	3627	5 K	226 K	231 K	12 K
Config. 2	263	3720	4 K	203 K	207 K	10 K
Config. 3	256	4063	4 K	215 K	219 K	11 K
Config. 4	212	3120	3 K	133 K	137 K	8 K
Config. 5	173	2496	2 K	84 K	86 K	6 K
Config. 6	154	2312	2 K	77 K	79 K	5 K

Note that in practice, the parcels used to be delivered via semi-trailers (basically, a truck with only one slightly larger container), which were replaced progressively by trucks with one or two trailer over the timespan which the six datasets cover. We do not take this into account in this thesis. We choose to use the same types of vehicles for the six configurations in order to obtain comparable results. Unlike trucks with two containers, the semi-trailers do not permit double deliveries. Moreover, the semi-trailers do not offer the opportunity to adapt the vehicle to the demand, as they can contain 3000 parcels while one current container contains 2000 parcels. Notice that our estimated cost of the current operational strategy is lower than the actual one, because we only consider the (simpler) strategy of using trucks with containers, while the actual current strategy uses a hybrid of semi-trailers and trucks with containers and is sometimes forced to use the less efficient semi-trailers.

2.3 Addition of New Paths to the Current Operational Strategy

We know that adding new paths to the current operational strategy will result in an improved optimal solution, but as it also increases the problem complexity, it is not clear that we can find such an improved solution. However, our goal is to add new paths to obtain a better solution (i.e., a solution of lower cost). Thus, compared to the current operational strategy, we allow the following new types of paths:

- the direct paths (i.e., the paths that send parcels from the sorting center to a delivery depot that is not in the catchment area of the origin);
- the repositioning of trucks outside the catchment area of the delivery depots in which it ended;
- the hub sorting operation and the paths with two sortings.

Table 3.5 shows that adding the possibility to reposition the vehicles outside their catchment areas add new operational links to the current operational strategy. It does not add operational paths (as they concern parcels) nor constraints. Indeed, there were already balance-design constraints and the empty repositionings do not have capacity constraint nor delivery constraint.

Table 3.5: MILP variables of the current operational strategy + optimization of empty repositioning

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Nb. of variables	Nb. of constraints
Config. 1	292	3627	11 K	226 K	237 K	12 K
Config. 2	263	3720	11 K	203 K	214 K	10 K
Config. 3	256	4063	11 K	215 K	226 K	11 K
Config. 4	212	3120	9 K	133 K	142 K	8 K
Config. 5	173	2496	7 K	84 K	91 K	6 K
Config. 6	154	2312	6 K	77 K	84 K	5 K

Table 3.6 shows that adding the possibility to do a hub sorting (in the given inner-hubs) to the current operational strategy to which we already have added the optimization of empty repositioning increases the number of operational paths (as they concern parcels). It does not add constraints as there were already delivery constraints for all the demands and the new possible paths are added in this already existing constraints. It does not add operational links as they were already modeled.

Table 3.6: MILP variables of the Current operational strategy + optimization of empty repositioning + hub sorting

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Nb. of variables	Nb. of constraints
Config. 1	292	3627	11 K	339 K	350 K	12 K
Config. 2	263	3720	11 K	392 K	403 K	10 K
Config. 3	256	4063	11 K	456 K	467 K	11 K
Config. 4	212	3120	9 K	264 K	273 K	8 K
Config. 5	173	2496	7 K	175 K	182 K	6 K
Config. 6	154	2312	6 K	172 K	178 K	5 K

Finally, adding the direct paths to the current strategy to which we already have added the optimization of empty repositionings and hub sorting increases the number of operational paths (direct paths are operational paths). It increases the number of operational links (before there was no connection from the sorting centers to the delivery depots outside their catchment areas). It increases the number of constraints as the direct links have capacity constraints. The current operational strategy with optimization of empty repositionings, hub sorting (in the given inner-hubs) and direct paths is Strategy 1 as in this strategy, the two soft business constraints are respected. Thus the number of variables for this strategy can be found in Table 3.7.

From now on, unless we state otherwise, we are allowing the direct paths, the repositioning of trucks outside the catchment area of the delivery depots in which it ended and the hub sorting.

2.4 Impact of the Soft Business Constraints on the Size of the MILP Formulation

In the previous Chapter (Chapter 2), we introduced two soft business constraints which impact the size of the MILP formulation. As a reminder, the first soft business constraint, respecting the catchment area, concerns doing the sorting in the sorting center associated to the delivery depot. The second constraint is related to using only the inner-hubs proposed by the transportation managers to do the hub sorting. In this section, we introduce four optimization strategies which respect or reject these two soft business constraints. We will test these strategies with the LHPTP-MILP (defined in Chapter 2 Section 3.3) and compare the results in Section 3.1.

Strategy 1: Respecting the Catchment Areas and the Provided Inner-Hubs

If we consider the current operational strategy to which we add empty repositionings optimization, hub-sorting (in the given inner-hubs) and direct paths, we have the Strategy 1 whose number of variables are presented in Table 3.7. In this optimization strategy, we respect both of the soft constraints. This restricts the operational paths in comparison with the following strategies (Strategies 2 to 4).

Table 3.7: MILP variables of the Strategy 1

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Nb. of variables	Nb. of constraints
Config. 1	292	3627	28 K	370 K	398 K	42 K
Config. 2	263	3720	27 K	420 K	447 K	38 K
Config. 3	256	4063	29 K	487 K	515 K	41 K
Config. 4	212	3120	21 K	284 K	305 K	28 K
Config. 5	173	2496	15 K	188 K	203 K	18 K
Config. 6	154	2312	13 K	183 K	196 K	16 K

Strategy 2: Respecting the Catchment Areas and Ignoring the Provided Inner-Hubs

Here, we respect the catchment areas for the last sorting and possibly use inner-hubs different from those provided by the transportation managers. This means that we consider all the sorting centers as candidates for the hub sorting (defined in Section 1.5 of Chapter 2). In this case, the possibilities for the operational paths are multiplied by more than 3 (see Table 3.8). Recall that in practice, some sorting centers cannot be used as inner-hubs for certain sites because they can be too far to make an operational path which deliver the parcels in less than two days. In such case, the variable is not created. If we consider the problem with all the sites and all the demands, each sorting center can be considered at least once (for at least one demand) as an inner-hub while respecting travel times and delivery times

Table 3.8: MILP variables of the Strategy 2

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Increase since S1 ⁵	Nb. of variables	Nb. of constraints
Config. 1	292	3627	28 K	1360 K	+ 268%	1388 K	42 K
Config. 2	263	3720	27 K	1418 K	+ 237%	1445 K	38 K
Config. 3	256	4063	29 K	1583 K	+ 225%	1612 K	41 K
Config. 4	212	3120	21 K	973 K	+ 242%	994 K	28 K
Config. 5	173	2496	15 K	623 K	+ 231%	637 K	18 K
Config. 6	154	2312	13 K	599 K	+ 227%	613 K	16 K

Strategy 3: Ignoring the Catchment Areas and Respecting the Provided Inner-Hubs

In this subsection, we consider the optimization strategy in which we respect the provided inner-hubs and ignore the catchment area constraint for the last sorting. This means that we consider all the sorting centers as candidates for the last sorting (defined in Section 1.5 of Chapter 2). In this case (see Table 3.9), the possibilities for the operational paths are multiplied by an even greater factor than in the Strategy 2. Indeed, the last sorting can be done in a new sorting center compared to the previous strategies. This impacts the number of operational paths with one sorting, but also the number of paths with two sortings as this last sorting can also be combined in a path with another sorting (done before it) performed at an inner-hub fixed by the transportation managers. Recall that in practice, some sorting centers cannot be used for the last sorting as they can be too far to make an operational path which deliver the parcels in less than two days. In this case, the variable is not created.

Table 3.9: MILP variables of the Strategy 3

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Increase since S1 ⁵	Nb. of variables	Nb. of constraints
Config. 1	292	3627	28 K	2204 K	+ 496%	2232 K	42 K
Config. 2	263	3720	27 K	2362 K	+ 462%	2389 K	38 K
Config. 3	256	4063	29 K	2734 K	+ 461%	2763 K	41 K
Config. 4	212	3120	21 K	1713 K	+ 503%	1734 K	28 K
Config. 5	173	2496	15 K	1084 K	+ 476%	1099 K	18 K
Config. 6	154	2312	13 K	1027 K	+ 461%	1040 K	16 K

Strategy 4: Ignoring the Catchment Areas and the Provided Inner-Hubs

Here, we ignore the catchment areas for the last sorting and ignore the provided inner-hubs. This means that we consider all the sorting centers as candidates for the hub

⁵increase of the number of operational paths compared to Strategy 1

sorting and for the last sorting (defined in Section 1.5 of Chapter 2). Table 3.10 shows the number of variables for the six configurations in this case.

Table 3.10: MILP variables of the Strategy 4

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of op. links	Nb. of op. paths	Increase since S1 ⁵	Nb. of variables	Nb. of constraints
Config. 1	292	3627	28 K	3194 K	+ 763%	3223 K	42 K
Config. 2	263	3720	27 K	3360 K	+ 700%	3387 K	38 K
Config. 3	256	4063	29 K	3831 K	+ 687%	3859 K	41 K
Config. 4	212	3120	21 K	2402 K	+ 746%	2423 K	28 K
Config. 5	173	2496	15 K	1519 K	+ 708%	1534 K	18 K
Config. 6	154	2312	13 K	1443 K	+ 689%	1457 K	16 K

2.5 Synthesis

For all the strategies, the number of operational links and constraints are the same. Indeed, in all four of our strategies, all the sorting centers are connected to all sorting centers (via links) and to all delivery depots, unlike in the current operational strategy (since the current strategy does not offer direct paths). The same argument stands for the number of constraints in the MILP formulation. Ignoring the soft business constraints does not add any delivery constraint, as all the parcels have to be delivered in all the optimization strategies. It does not add any capacity constraint as it does not add any new operational link which would not already have a capacity constraint. The sites and vehicle types allowed is the same in all four strategies so each strategy contains the same number of balance constraints.

From Strategy 1 to 2, the number of operational paths increases on average by 238% over the six datasets. From Strategy 1 to 3, the number of operational paths grows on average by 476%. And from Strategy 1 to 4, the number of operational paths increases on average by 716%. The reasons for these huge increases are explained in Section 2.1. The explosion of the number of possible operational paths leads to an increase in the corresponding variables, which are the continuous variables. Even if the integer variables do not increase in number, the capacity constraints (see eqref) relate these variables to the operational path variables whose number increases. We will see in Section 3 that this explosion has a negative impact when trying to find a solution with the MILP solver.

3 Exact Approach: Direct Solving with MILP

In this Section, we present the results obtained when testing the MILP formulation introduced in Section 3.3 of Chapter 2 on the instances presented in Section 1. Then we elaborate on these results and we explain why we cannot simply remove operational paths in order to lessen the impact of the resulting combinatorial explosion. Indeed, given the size of the problem and the MILP we know that giving the MILP to an MILP solver will

be able to find a feasible solution in most cases but it will not produce the optimal solution in reasonable time. This is due to the number of variables discussed previously and the fact that these variables are both floats and integers with interdependence between them.

3.1 Application of the MILP on the LHPTP

In this Section we describe the results of the LHPTP-MILP presented in Chapter 2 Section 3.3. We run this MILP for 6 hours on 32 threads. The test environment is composed of a Linux server with 32 CPU and 150 Gbytes of RAM. The solver used is CPLEX 12.8.

We compare here the results obtained for the four strategies described in Section 2. The main difference between these optimization strategies is the number of operational paths proposed to the solver. Note that, as presented in Chapter 2 Section 3.3, the link variables are integers while the paths variables are floats. As discussed in the previous section, the number of link variables does not vary with the optimization strategies while the number of path variables is significantly increased as we go from Strategy 1 to 4. The number of integer variables (number of operational links) and float variables (number of operational paths) has been presented in Tables 3.7, 3.8, 3.9 and 3.10, thus it is not repeated in the tables of this section.

Potentially the number of operational paths is huge and is not tractable by a commercial MILP solver. Thus we restrict (with the optimization strategies) the number of operational paths in the MILP whose lower bound represents poorly the reality on the ground. The aim is to have a tractable model which is still able to find relevant solutions to the industrial problem. Indeed, the objective is not to have the smallest gap but the cheapest transportation plan. In other words, we are trying to find a trade-off between the cost of the best solution found and the minimization of the gap. Restricting the number of path variables allows to have a smaller gap, but this reduces as well the solution space so the lower bound has a higher value and so does the optimal solution of the MILP that the solver minimizes. Even if this optimal solution is not reached, it limits the best solution found which is what we will compare in the upcoming tables. Note that as the transportation plan depends on the possible paths offered, there is no guarantee that the optimal solution of the real world problem that the model represents is even above the lower bound of the model.

Table 3.11: Results of the Current Operational Strategy

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of variables	Best Obj. in 6 hours	Gap ⁶ (%)
Configuration 1	292	3627	231 K	417 K	8.10
Configuration 2	263	3720	207 K	451 K	7.98
Configuration 3	256	4063	219 K	482 K	9.36
Configuration 4	212	3120	137 K	497 K	5.82
Configuration 5	173	2496	86 K	512 K	4.36
Configuration 6	154	2312	79 K	531 K	3.23

First, Table 3.4 gives the best solutions found while respecting the current operational strategy that we aim to improve. We have to compute these values (in Table 3.4) as the data we have are predictive and represent an average day for each configuration and the postal company did not provide the objective values (i.e., an actual implemented solution for these instances). Even if the problems are small, the MILP solver does not find the optimal solution, this is probably due to the symmetries induced by the trucks with one or two containers [CRWM07].

Table 3.12: Results with Strategy 1

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of variables	Best Obj. in 6 hours	Gap ⁶ (%)	Gain over current strategy
Configuration 1	292	3627	398 K	373 K	11.66	10.7%
Configuration 2	263	3720	447 K	398 K	14.62	11.8%
Configuration 3	256	4063	515 K	416 K	13.74	13.7%
Configuration 4	212	3120	305 K	409 K	9.29	18.8%
Configuration 5	173	2496	203 K	424 K	11.44	17.3%
Configuration 6	154	2312	196 K	430 K	8.80	19.0%

Strategy 1 is computationally the better optimization strategy as it has fewer variables. If one strategy over the four should reach its optimal solution it would be this one. Thus it possibly could be as good as less restricted optimization strategies for which the MILP solver is lost in the solution space at it is too large. In Table 3.12, we can observe the smallest gaps (i.e., smaller than in Tables 3.13, 3.14 and 3.15) but not the smallest objective values over all the strategies (Tables 3.12, 3.13, 3.14 and 3.15). Indeed, as this optimization strategy offer less possibilities for optimization, the optimal solution of this restricted problem has a higher cost than the optimal solution of the less restricted optimization strategies. We can conclude that Strategy 1 is too restricted with respect to the trade-off we are looking for. Moreover, note that all the operational paths in Strategy 1 are included in Strategies 2, 3 and 4, so the optimal solution of this Strategy is a feasible solution for the other strategies.

Table 3.13: Results with Strategy 2

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of variables	Best Obj. in 6 hours	Gap ⁶ (%)	Gain over current strategy
Configuration 1	292	3627	1388 K	377 K	13.73	9.6%
Configuration 2	263	3720	1445 K	402 K	17.26	10.8%
Configuration 3	256	4063	1612 K	418 K	16.87	13.3%
Configuration 4	212	3120	994 K	405 K	12.77	18.6%
Configuration 5	173	2496	637 K	406 K	10.20	20.8%
Configuration 6	154	2312	613 K	423 K	10.95	20.4%

⁶Gap to the best lower bound found by the solver

The last column of Tables 3.12 and Table 3.13 gives the gain over the current strategy. It is the money the transportation plan found by the solver allows to save compared to the current operational strategy.

The operational paths of Strategy 2 are included in the paths of Strategy 4, but not in the paths of Strategy 3. As we offer more possibilities for optimization than in Strategy 1, the optimal solution of this restricted problem is lower than the optimal solution of Strategy 1. Table 3.13, which represents the results for Strategy 2, has the smallest objective values but not the smallest gaps. The best solutions found by the solver are better than the ones found with Strategy 1, but since the gaps are higher than with Strategy 1, it shows that the MILP starts to have too many variables for the MILP solver.

Table 3.14: Results with Strategy 3

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of variables	Best Obj. in 6 hours	Gap ⁶ (%)	Best Obj. in 24 h	Gap ⁶ (%)
Config. 1	292	3627	2232 K	1213 K	75.12	504 K	39.51
Config. 2	263	3720	2389 K	1302 K	76.35	496 K	37.4
Config. 3	256	4063	2763 K	799 K	60.83	799 K	60.20
Config. 4	212	3120	1734 K	443 K	28.87	419 K	24.03
Config. 5	173	2496	1099 K	424 K	24.86	424 K	24.86
Config. 6	154	2312	1040 K	419 K	23.76	419 K	23.53

In Table 3.14, we have very bad solutions compared to the previous strategies (Strategies 1 and 2). The number of variables offered to the solver prevents it from finding even a good solution. The variables of this Strategy are including the variables of Strategy 1 and they are included in the variable set of Strategy 4. This means that the solutions found for Strategy 1 are feasible solutions for this optimization strategy. But we offer too many possibilities for optimization, therefore, we do not even see that the optimal solution is improved compared to Strategy 1 as we do not reach it (the gap is not 0%). We can conclude that Strategy 3 does not provide useful output when we use an MILP.

Table 3.15: Results with Strategy 4

Name of the dataset	Nb. of sites	Nb. of demands	Nb. of variables	Best Obj. in 6 h	Gap ⁶ (%)	Best Obj. in 24 h	Gap ⁶ (%)
Config. 1	292	3627	3223 K	430 K	30.13	430 K	29.19
Config. 2	263	3720	3387 K	1404 K	78.16	445 K	30.19
Config. 3	256	4063	3859K	753 K	58.46	440 K	27.63
Config. 4	212	3120	2423 K	580 K	45.64	570 K	44.18
Config. 5	173	2496	1534 K	419 K	23.99	418 K	23.69
Config. 6	154	2312	1457 K	410 K	22.07	410 K	21.73

Strategy 4, depicted in Table 3.15, involves the most variables. As we offer more

possibilities for optimization, the optimal solution of this problem is the lowest (or the same) of all the optimization strategies. Indeed, all the operational paths of the previous strategies are included in the set of paths offered in this strategy. But here again, we have very bad solutions compared to the first two optimization strategies. However, we can note that the values of the objective function are better with this strategy than with Strategy 3.

From Tables 3.12, 3.13, 3.14 and 3.15, we can conclude that the best optimization strategy for finding an optimal solution with an MILP (with respect to the respective MILP formulation) is Strategy 2. Strategy 2 presents a good trade-off between the size of the problem (number of variables) and the quality of the solution obtained. However, if we could have a heuristic which allows us to handle more variables, it would be interesting to try Strategy 4, as it is the least restrictive in terms of the possible paths (i.e., it does not restrict the possibilities for the paths at all). Thus theoretically, it should lead to lower optimal solutions. Strategy 3 is definitively not efficient for an MILP, and Strategy 1 is too restricted (i.e., it has too few variables to yield a low cost solution).

In the rest of this manuscript, we will place ourselves in the case of Strategy 2 as it will allow us to compare the results obtained with our algorithms to the ones obtained with the MILP.

3.2 Limits of LHPTP-MILP on our Case Study

One way to obtain a good solution with the LHPTP-MILP is to formulate a smaller instance without compromising the solution quality. In the previous subsection, we tested various optimization strategies to decide which soft constraints should be respected or rejected. Respecting these constraints would lead to smaller instances of the problem as it reduces the number of operational paths offered to the MILP. We showed that Strategy 2 is a good way to select paths to build smaller instances which respect the catchment areas. This optimization strategy is relevant for the management of sites at an operational level and reduces the number of variables in the MILP formulation. It allows us to find a feasible solution that has a gap of around 15% for this smaller MILP formulation. But since we cannot prove the best solution that we have found is optimal, our goal is to improve this solution further, since we believe it is not in fact optimal.

In order to obtain improved solutions, we do not want to simply remove more paths. Indeed, we could remove some more path types, but they all seem to be useful. For instance in the best solution obtained for Strategy 2 (Configuration 5), 37.9% of the parcels are delivered on a direct path (outside its catchment area), 8.9% of the parcels are delivered inside the same catchment area, 48.8% of the parcels are delivered on a path with one sorting and 4.4% of the parcels are delivered on a path with two sortings. This shows that the direct paths are used in good and maybe optimal solutions and should not be removed from the model. The path with one sorting is the most used path type. The paths with two sortings represent a small percentage of parcel paths. But we need to remember that the demands which use this type of path are small demands as they are sorted twice and the sorting has a cost per parcel (so large demands are unlikely to

be sorted multiple times in a low-cost solution). So this type of paths should not be removed either as it allows consolidation of parcels, which leads to the use of less trucks in the transportation plan. Therefore, we need another way to reduce our problem size in order to obtain a better solution from the MILP solver.

4 Conclusion

In this chapter, we introduce the datasets used for the tests which are given by a postal company. We discuss the size of the MILP formulations on this input. In particular, we explore whether or not we should respect the two soft business constraints (respecting the catchment areas for the last sorting and the inner-hubs provided by the transportation managers for the hub sorting). We apply the MILP on all the optimization strategies we propose that respect or ignore these two constraints. The conclusion of these tests is that the best option is to use Strategy 2: Respect the catchment areas and ignore the provided inner-hubs. It is the best option in terms of MILP formulation size: It allows us to have sufficiently many options to obtain a lower bound small enough to have good results, while at the same time, it does not have too many variables which would prevent us from finding a good solution. Therefore we can say we validate the business constraint of the catchment areas. It is a good news from an operational point of view as the operational people prefer to change less their habits. Moreover, respecting the catchment area has lots of benefits for the implementation of optimized solutions in practice: It facilitates the organization by regions and allows to deal with the sizing and definition of catchments in a separate problem, taking into account in particular the sorting capacity constraints on sorting centers. Indeed, defining the catchment areas is another problem known as the Hub and Spoke Network Design problem (see Chapter 1 Section 3.2) that we do not treat here.

As for the other soft constraint (about the inner-hubs), in the following we either use the inner-hubs provided or use all the sorting centers as potential inner-hubs, and name inner-hub the sorting centers used for hub-sorting and "simple" sorting centers the others. In this last case, we will discuss the proposed inner-hubs defined by transportation managers and try to find better options for the hub sorting locations without developing a hub location model. Note that using all the sorting centers as inner-hubs is not an operational hurdle as the hub sorting takes place in the same time slots as the "usual" sorting. Thus there is no extra costs to turn a "simple" sorting center into an inner-hub. The only issue is the sorting capacity, that we do not consider in this case study.

In the subsequent chapters, we will consider a model in which we allow the direct paths, the repositioning of trucks outside the catchment area of the delivery depots in which it stops and the hub sorting. We will successively test what happens when ignoring the soft constraint on hub sorting (Strategy 2) and respecting this constraint (Strategy 1), while respecting the catchment areas for all the tests.

Since we actually need to include more paths than we can afford in the smaller MILP instances that are solvable, we need a way to reduce our problem size in order to obtain a

better solution from the MILP solver. To achieve this, we will propose two algorithms with divide-and-conquer approaches, in which we partition the problem into smaller instances. In these two heuristics, we will take advantage of the grouping by catchment areas as it allows to cluster sites more easily or to aggregate demands (see respectively Chapter 4 and 5).

Chapter 4

Cluster-based Approach for Solving the LHPTP

Contents

1	Background and Previous Work	79
1.1	Clustering Algorithms	79
1.2	Cluster-Based Approaches in Transportation Optimization	82
2	<i>k</i>-Clusters Algorithm	83
2.1	Approach to Solving the LHPTP	83
2.2	Detailed Steps of the <i>k</i> -Clusters Algorithm	85
3	Applying the Clustering Methods on our Data	87
3.1	Input Data Provided to the Clustering Algorithms	88
3.2	Input Data for Each Clustering Algorithm	90
3.3	Output Clusterings	91
3.4	Confirmation of the Catchment Areas Utility	95
3.5	Structural Properties of the Data Revealed via Clustering	96
3.6	Synthesis	98
4	Simulations and Results	98
4.1	Simulation Parameters	98
4.2	Evaluation of Results	99
	Results with Strategy 2	100
	Results with Strategy 1	102
4.3	Synthesis	102
5	Conclusion	103

Our goal is to set up a general and efficient method to tackle the large-size data instances of LHPTP provided by a postal company (see Chapter 3 Section 1 for details). This approach should be able to exploit an existing MILP or other methods for computing a solution. As we have seen in the previous chapter (Chapter 3), the size of the network we need to optimize is fixed and is quite large. Our first approach to solve this problem, whose results are presented in the previous chapter, was to formulate it as an Mixed-Integer Linear Program (MILP). Since the number of potential operational paths results in our formulation being too large to be solved by an MILP solver, one of our contributions is to propose a heuristic approach based on divide-and-conquer paradigm and clustering techniques. In order to divide the problem, we cluster the sites so that parcels can be routed either inside a cluster (**intracluster problems**) or between clusters (**intercluster problem**). This results in subproblems of small size which can be solved with the LHPTP-MILP. We compare standard clustering methods such as spectral clustering [NJW02], hierarchical clustering [Joh67] and k -means [Llo82] to determine which one works best on our case study. As it is typically the case when clustering algorithms are used in a black-box manner, the difficult aspect is to map our problem instance to the input of these algorithms. Specifically, the problem of determining an accurate distance function and/or similarity function to obtain meaningful and useful clusters is non-trivial.

Clustering-based approaches have been used on related problems such as Vehicle Routing Problems (VRP) which have a similar hurdle with too many variables in their MILP formulations. These previous works tailor their clustering approaches specifically to their objective or problem data. Our approach is different since we apply well-known, off-the-shelf, clustering algorithms to obtain smaller size subproblems, which can then be solved relatively quickly and/or in parallel.

In Section 1, we give some background on common clustering methods and then we discuss their use in transportation optimization. In Section 2, we present an algorithm that combines clustering techniques with the LHPTP-MILP to obtain a feasible solution to the LHPTP that is better than the solution obtained using the LHPTP-MILP alone. Then in Section 3, we discuss the necessary technical details of the clustering methods and how they are applied in our algorithm. For example, we define the similarity functions that we use in our algorithm. We then evaluate the structural information that is revealed when we apply the clustering methods on our datasets. In Section 4, we present results obtained with our algorithm on large-size real data instances and compare them to those obtained with a direct use of an LHPTP-MILP alone.

This algorithm has been presented at the ROADEF 2020 conference [GCH⁺20].

1 Background and Previous Work

In this section, we define clustering and describe the clustering algorithms that we will use as subroutines for the cluster-based algorithm we present in Section 2. We then give some background on cluster-based algorithms used in transportation optimization.

1.1 Clustering Algorithms

Clustering is a Machine Learning technique which allows grouping of unlabeled data points into meaningful groups. Clustering is defined by Jain et al. [JMF99] as the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The goal of clustering is to separate a finite unlabeled data set into a finite and discrete set of “natural”, hidden data structures [XW05]. The idea is to transform unlabelled data into groups which are labelled as having common features. So one could consider, like [JMF99], that in a sense, labels are associated with clusters also, but these category labels are data driven (e.g., they are obtained solely from the data).

The steps of a clustering process are detailed in [JD88]. They are:

1. pattern representation (optionally including feature extraction and/or selection),
2. definition of a pattern proximity measure appropriate to the data domain,
3. clustering or grouping,
4. data abstraction (if needed), and
5. assessment of output (if needed).

For our case study, we will go through these steps in Section 3.

Clustering algorithms partition data into **clusters** (groups, subsets, or categories). If the clusters are usually defined as the output of a clustering, there is no universal characterization [XW05]. Note that all clustering algorithms produce clusters when presented with data, regardless of whether the data contain clusters or not. If the data do contain clusters, some clustering algorithms may obtain “better” clusters than others. A survey on cluster analysis is provided by Duran and Odell [DO13]. Most researchers describe a cluster by considering the internal homogeneity and the external separation: Patterns in the same cluster should be similar to each other, while patterns in different clusters should be different from each other.

There are two types of clustering on a sample S of N points which assigns all the points to one and only one cluster [HJ97, XW05]:

1. A K -partition $P = \{C_1, C_2, \dots, C_K\}$ of S into K clusters with $K \leq N$:
 - (a) $C_i \neq \emptyset$, $i \in \{1, 2, \dots, K\}$;
 - (b) $C_i \cap C_j \neq \emptyset$, $i, j \in \{1, 2, \dots, K\}$ and $i \neq j$;
 - (c) $\bigcup_{i=1}^K C_i = S$.

2. A hierarchy which is a tree-like partition $H = \{P_1, P_2, \dots, P_q\}$ of $q \leq N$ partitions of S such that $C_i \in P_k, C_j \in P_l$ and $k > l$ imply $C_i \subset C_j$ or $C_i \cap C_j = \emptyset$
 $\forall i, j, k, l \in \{1, \dots, q\}$ with $i \neq j$.

We will use the most well-known clustering method of the first type – the k -means algorithm – and two well-known clustering methods of the second type – the hierarchical agglomerative clustering and the spectral clustering – on our case study. We present these algorithms hereafter.

Hierarchical Clustering

Hierarchical clustering is a family of clustering methods that builds nested clusters by merging or splitting them successively [Das02]. This family of clustering methods is divided into two categories: the top-down and the bottom-up. We use the bottom-up version of the hierarchical clustering which is called hierarchical agglomerative clustering. The clustering output by this algorithm is a hierarchy of clusters which forms a tree whose root is the unique cluster that gathers all the points to clusters. The leaves are the clusters made of only one point in each cluster. Between the leaves and the root a bottom-up approach is used to recursively merge the couple of clusters that minimally increases the given linkage distance. The algorithm is as follows:

1. Each point is its own cluster.
2. On each iteration, the two closest clusters under construction are grouped into one cluster.
3. Step 2 is repeated until all there is only one cluster, or until the targeted number of cluster is reached.
4. If the number of clusters was not predefined, the tree (or dendrogram) allows the user to choose which number of clusters seems the best.

The hierarchical agglomerative clustering has a complexity of $O(N^3)$, but it captures well the natural hierarchy of the data provided.

Spectral Clustering

The spectral clustering algorithm, detailed in [NJW02], is drawn from linear algebra. The output clustering aims to minimize the exchanges between the clusters with respect to an input similarity matrix. It extracts this information of the first eigenvalues of the distance matrix. It first does a low-dimensional spectral embedding of the input matrix. Then it uses a clustering method to cluster this low-dimensional point set. The clustering of a sample $S = \{s_1, \dots, s_N\}$ works as follows:

1. Compute the affinity matrix (also called similarity matrix) $A \in \mathbb{R}^{N \times N}$ defined by $A_{ij} = \exp(-||s_i - s_j||/(2\sigma^2))$ if $i \neq j$, and $A_{ii} = 0$.

2. Compute D the diagonal matrix in which D_{ii} is the sum of A 's i^{th} row, and construct the Laplacian-derived ¹ matrix $L = D^{-1/2}AD^{-1/2}$.
3. Find X_1, X_2, \dots, X_k , the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalues), and form the matrix $X = [X_1 X_2 \dots X_k] \in \mathbb{R}^{N \times k}$ by stacking the eigenvectors in columns.
4. Form the matrix Y from X by renormalizing each of X 's rows to have unit length (i.e. $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$).
5. Treating each row of Y as a point in \mathbb{R}^k , cluster them into k clusters via k -means or another clustering algorithm (in our case it is k -means).
6. Assign the original point S_i to cluster j if and only if row i of the matrix Y was assigned to cluster j .

It is especially computationally efficient if the affinity matrix is sparse. Otherwise its complexity is $O(N^3)$. It requires a positive semi-definite matrix as an input.

k -Means Clustering

The k -means clustering, also called Lloyd's algorithm [Llo82], is a local clustering, which uses a distance matrix as input, unlike the previous two clusterings which take as input similarity matrices. It requires the number k of clusters to be specified as it is a k -partition clustering. This clustering algorithm isolates sites in k disjoint clusters of roughly equal variance by minimizing the within-cluster sum-of-squares: $\sum_{i=0}^n \min_{\mu_j \in C} (||x_i - \mu_j||^2)$ where μ_j is the mean of the cluster j . It works as follows:

1. Choose the number k of clusters to create.
2. Initialize a k -partition randomly or based on some prior knowledge by picking center points for the group.
3. Each data point is classified by computing the distance between that point and each group center point, then the data point is assigned to the group whose center is closest to it.
4. Based on these newly classified points, recompute the group center points by taking the mean of all the positions of the points in the group.
5. Repeat Steps 3 and 4 until the group center points stop changing between iterations.

The standard k -means algorithm finds clusters that are convex when the input distance function used obeys the triangle inequality. The k -means algorithm has the advantage that it is fast in practice [Pak14].

¹the actual Laplacian is $I-L$

1.2 Cluster-Based Approaches in Transportation Optimization

Transportation optimization problems, by nature, lend themselves well to cluster-based optimization. Indeed, dividing a transportation problem into clusters on which it is possible to solve the same transportation problem of smaller size before putting the solutions back together seems natural. But clustering techniques can be used in other ways to create heuristics. We describe here some cluster-based heuristics used in transportation optimization focusing on how the clustering is used. We mainly discuss VRP (vehicle routing problems) instead of SND or long-haul transportation problems. Indeed, the VRP lends itself well to division into physical subproblems as the addition of the solutions of these subproblems forms a solution to the complete problem. This might not be the case in SND and especially in parcel transportation as parcels are sent from all the cities to all the cities. There is no pair of cities which do not exchange parcels. Thus there is an intercluster subproblem to solve and the merging phase of the divide-and-conquer approach is non-trivial, which is likely the reason why approaches to SND involving clustering seem to be rarely found in the literature.

There are three main groups of cluster-based approaches to solve the VRP and its variants: In the first one, the authors [CX06, HWH15, HGLLP0⁺18, JH12] use the k -means algorithm to cluster the customers as a first step. Chunyu and Xiaobo [CX06] and Herrera-Granda et al. [HGLLP0⁺18] combine it to a hierarchical clustering to group the clusters to serve them on a route. He et al. [HWH15] and Ji and Wang [JH12] respectively use a particle swarm optimization and the Artificial Fish-Swarm Algorithm for the vehicle routing optimization. The second approach consists of using a distance-based clustering method [CMPT08, MJL19, PCDCLS16]. Crainic et al. [CMPT08] combines the clustering to improvements heuristics to improve the solutions obtained in the first step on the two-echelon VRP. Min et al. [MJL19] summarize their algorithm as clustering first and routing later. Patiño Chirva et al. [PCDCLS16] use a two steps method on the collection services of recyclable waste problem: first a clustering step (with a centroid-based heuristic algorithm) and then a routing step. The last approach is when dedicated heuristic are used to cluster [DC07, TS01]. Thangiah and Salhi [TS01] create a genetic clustering for the multidepot VRP. The best set of clusters obtained during the search of the genetic clustering is used to route the vehicles. Dondo and Cerdá [DC07] build a heuristic-based dedicated clustering method used in a preprocessing stage which clusters nodes together. This heuristic aims to yield a more compact cluster-based MILP problem formulation for the multi-depot heterogeneous fleet vehicle routing problem with time windows.

Note that there is a problem called Clustered Vehicle Routing Problem (Clu-VRP), in which the clusters are part of the input and not of the optimization approach [BEV14, SS⁺08]. To conclude on the VRP optimization via clustering, we can say that this clustering approach fits well to this transportation problem. Indeed, once the clusters are created, the subproblems can be solved independently and that makes a solution for the original problem (before clustering) while in the LHPTP, there are demands between clusters whose transport also needs to be optimized.

Cluster-based approaches also have been used for transportation problems which are not variations of VRP. Jiang et al. [JBW⁺21] recently applied clustering techniques to demand prediction problems for the stochastic Service Network Design. The clustering algorithm used is k -means. This method has not been used for deterministic Service Network Design to the best of our knowledge. Wang et al. [WAL⁺18] use clustering techniques in the two-echelon location-routing optimization with time windows for customer clustering. They cluster customers based on what they purchase; this is a knowledge-based approach.

The takeaway message of this subsection is that in transportation optimization, clustering techniques can be used for two purposes:

- To physically group customers or sites to serve as it is done in most of the cluster-based approaches for VRP. It allows to take advantage of the distribution of the customers as minimizing the distance is a key aspect in transportation optimization.
- To group on demand (i.e., demand forecast or expectation) as it is done for the non VRP problems and some VRP. It allows to learn from the input of the problem in order to propose an optimization adapted to this knowledge.

For the LHPTP, these two points of view will be tested separately and even combined in the next sections. Indeed, as we do not know which of these two pieces of information will works better with the cluster-based algorithm we propose, we will test various clustering algorithms with different input data (see Section 3 for details).

2 k -Clusters Algorithm

As discussed previously in Section 3.2 of Chapter 3, the LHPTP-MILP is too large for the complete dataset to be solved optimally or even reach a near optimal solution. Thus, we propose an algorithm which follows a divide-and-conquer approach based on the clustering methods detailed in Section 1 applied on the input detailed in Section 3. Indeed, the resolution process using MILP is much more efficient when we reduce the number of sites considered. This algorithm, denoted **k -Clusters algorithm**, is detailed in Algorithm 1 and in this Section.

2.1 Approach to Solving the LHPTP

The k -Clusters algorithm consists of solving several times the LHPTP-MILP with various inputs. Each MILP is associated with fewer variables than the original one. All the MILPs mentioned here are represented by the set of Equations (2.1) in Section 3.3 of Chapter 3 even if sometimes some variables are fixed. We denote $MILP_{fixed x}$ the MILP with all the x_p^d (the rate of demand d using path p) fixed to a precomputed feasible solution. In this model, the y_l^{veh} (the number of each vehicles of type veh on link l) are unconstrained. Note that the main idea of our algorithm is to divide the problem into

smaller subproblems and to solve them optimally via the LHPTP-MILP, but a solution to these subproblems could also be found by other means.

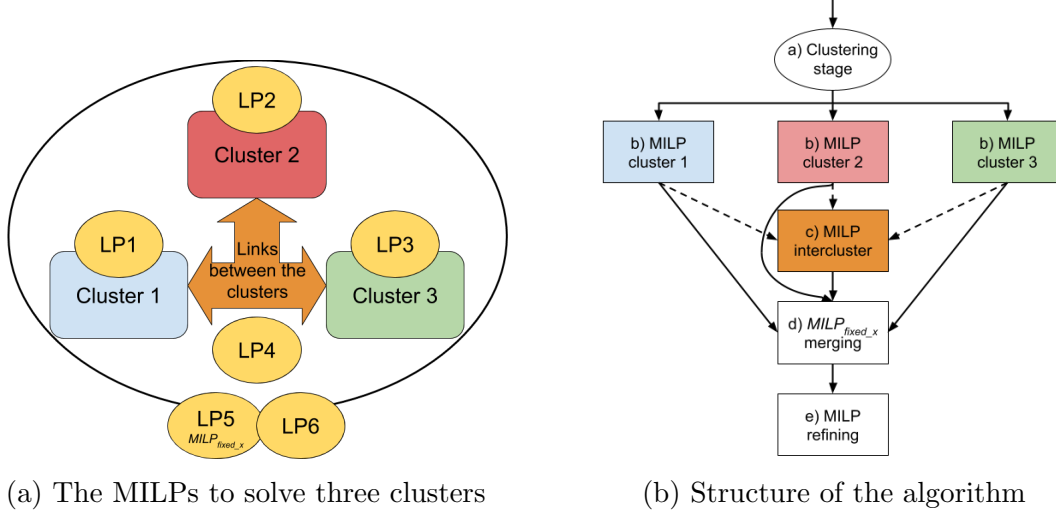


Figure 4.1: k -Clusters algorithm

To clarify the course of events in the k -Clusters algorithm, Figure 4.1a represents the clusters for $k = 3$, the LP to solve the subproblems they form and the order in which they are solved. Figure 4.1b gives the order and interactions between steps of the algorithm for $k = 3$. These five steps are:

- (a) Clustering stage: clustering the sorting centers,
- (b) Intracusters problems: solving the routing problem within each cluster using the MILP (see LP1, 2 and 3 in Figure 4.1a),
- (c) Intercluster problem: solving the problem between clusters using the MILP (see LP4 in Figure 4.1a),
- (d) Merging stage: solving a global MILP (see LP5 in Figure 4.1a) to merge the solutions from steps (b) and (c) in a cost-effective way with x_p^d fixed ($MILP_{fixed x}$),
- (e) Refining stage: verifying the constraints via solving the LHPTP-MILP (see LP6 in Figure 4.1a) on the complete instance.

As we divide the global problem into subproblems, we solve independent intracuster problems which cannot communicate (i.e. they are solved independently). The challenge of the k -Clusters algorithm (Algorithm 1) is to find a tradeoff between the exploitation of global and local information. Indeed, it is useful from a computational perspective to partition the problem into subproblems, but it is also necessary to step back to have a global view of the problem in order to take advantage of consolidation of demands and address certain requirements. For instance, we must coordinate the repositioning of empty trucks to the sorting centers. We might also consolidate demands that are routed in subproblems which are solved independently. Indeed, demands between different intracuster subproblems cannot be consolidated whereas demands in the intercluster subproblem can be consolidated with demands in an intracuster subproblem which has already been

solved. This is due to the fact that the intracluster subproblems are solved independently and their results have to be added to construct the global solution. Note that for our problem, the sum of optimal solutions for the subproblems does not necessarily make a globally optimal solution.

Algorithm 1: k -Clusters Algorithm

```

1  a) Clustering Stage
2  divide the sites into  $k$  clusters
3  for each operational link  $l$  do
4     $\lfloor$  initialize the variable representing the operational link:  $y_l^{veh} = 0$ 
5  b) Intracluster problems
6  for each one of the  $k$  clusters do
7     $\lfloor$  solve the MILP to route the internal demands of the cluster  $k$ 
8     $\lfloor$  increment the  $y_l^{veh}$ 
9  c) Intercluster problem
10 according to the MILP solutions build the MILP to route the remaining
    demands according to the following rule:
11 for each path  $p$  do
12    $\lfloor$  if the path  $p$  is made of already chosen operational links plus one new
    operational link then
13    $\lfloor$  activate this path
14 solve this  $k+1$ st MILP and increment the  $y_l^{veh}$ 
15 d) Merging Stage
16 solve  $MILP_{fixedx}$  with all the demands, with only the chosen operational paths
    (the  $x_p^d$ ) to optimize the flow and check the feasibility (until a 1% gap is reached)
17 e) Refining Stage
18 solve a last MILP with all the demands, all the operational paths and the
    solution of the previous MILP injected as a first solution (until a time limit is
    reached)

```

2.2 Detailed Steps of the k -Clusters Algorithm

In the following, we detail each of the five steps of our proposed resolution approach, expressed in Algorithm 1.

a) Clustering Stage

The structure of the demand graph (defined in Chapter 2 Section 3.2) is such that there is no natural clustering in a graph theoretical sense (e.g., there are no sparse cuts). Thus, we tried various clustering methods, detailed in Section 3, to find which one results in the best solutions for our problem. We also vary the number of clusters between two and four, because we want to determine which number of clusters provides the best balance

between the reduction of the size of the problems inside the clusters and the resulting loss of optimality. Indeed, if we have too many clusters, we will not be able to take full advantage of the benefits yielded by the consolidation of parcels. This is discussed in Section 3.5.

b) Solving the Intracluster Problems

In this step, we solve the smaller versions of our problem on the clusters using the LHPTP-MILP (see MILP (2.1) in Section 3.3). For k clusters, k models are created for our problem. They represent the internal demands of each cluster. They are solved in k independent executions of MILPs. After each solving, the chosen operational links are memorized in an incremental way: we have a table which contains the variables and each solving adds the number of vehicles it needs to this table.

c) Solving the Intercluster Problem

We run a $k+1$ st MILP to find one or more operational paths for the intercluster demands. After this computation, the chosen operational paths are recorded. But the intercluster demands are not sparse (see Table 4.2). We observe that, for two clusters, when we have satisfied the demands inside the clusters, we have treated roughly half of the demands. And the more clusters there are, the larger the intercluster problem is (see Section 3.5). To solve this intercluster problem, we first tried to simply solve a $k+1$ st MILP (see set of Equations (2.1)) with the demands between the clusters and the operational paths to satisfy these demands (see Figure 4.1a for $k=3$). However, the problem of routing between clusters still has too many variables and we need to investigate methods to decrease the size of the instance without increasing the cost of the solution. To do so, we can sparsify the path graph by pre-selecting some potential good operational paths.

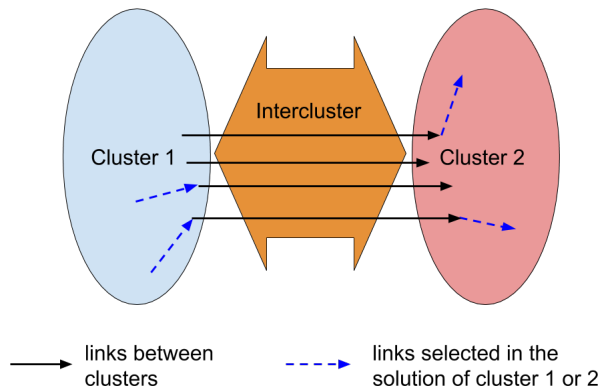


Figure 4.2: The paths activated between two clusters

Therefore we choose to allow only the operational paths made of an operational link which changes cluster (intercluster operational link in black in Figure 4.2) plus possibly

other operational links already activated in the first k solutions (in dotted blue in Figure 4.2). It restricts the possibilities quite brutally but makes the problem manageable by the solver (see Section 3.5) and allows to limit the number of operational links and of vehicles used, which aids the MILP solver.

We decide to solve only one intercluster problem and not a series of intercluster subproblems pairing the clusters in order to enhance consolidation. Indeed, all the subproblems are solved independently. And when it is not the case, when some operational links are re-used, the interaction between the subproblems is not comparable as when the demands are considered in a same optimization problem.

d) Merging Stage

In this step, we solve a global MILP (for Equation (2.1)) to merge the locally optimized solutions: $MILP_{fixed x}$, in which we fix the chosen operational paths variables, determined in the previous steps for all the intracluster problems and the intercluster problem. As these variables are the only variables activated, this $MILP_{fixed x}$ is solved nearly instantaneously. It is made to recompute the y variables (number of vehicles on the operational links) actually needed since the $k + 1$ MILP computations of the previous steps are independent therefore some vehicles might have been counted twice or more. Indeed, the intercluster computation is performed after the intracluster computations. Thus the only type of consolidation between intercluster and intracluster subproblems is that we might reuse some operational links in both solutions. This can be viewed as “light weight” consolidation as compared to the more involved consolidation performed in the intracluster subproblems. This computation also allows to optimize the empty repositionings of vehicles with a global point of view. To summarize, this solution is the one that we compute using the operational paths for parcels chosen in our successive solutions but it optimizes the vehicle flow. We call this solution obtained at the end of this merging stage the “**computed solution**” to the problem in Tables 4.4, 4.5 and 4.6.

e) Refining Stage

In this last step, the global solution obtained at the end of the merging stage is injected as a first feasible solution in a run of LHPTP-MILP without any fixed variables (once again it is the MILP for the set of Equations (2.1)). It allows us to verify the feasibility of our complete solution and to evaluate it by a distance to a lower bound. Indeed, when the operational paths variables are fixed (in $MILP_{fixed x}$ in the previous step) the lower bound is not relevant. The solution obtained at the end of this step is called “**Refined solution**” in Tables 4.4, 4.5 and 4.6.

3 Applying the Clustering Methods on our Data

Now that we have presented in Section 2 the k -Clusters Algorithm, our objective is to provide various clustering as inputs for the k -Clusters Algorithm to see which one works

better. We want to cluster the sites in such a way that merging solutions obtained for the subproblems results in a good global solution. One way to do this would be to find a clustering that has few exchanges between clusters (fewer intercluster variables, i.e., a sparse cut). This type of clustering could be obtained via an algorithm that finds sparse cuts, but in general our data instances do not seem to have such cuts (see Section 3.5).

We test clustering algorithms with different properties in order to find which one works the best on our data. Thus, we apply on our data the three clustering algorithms introduced in Section 1: the hierarchical clustering, the spectral clustering and the k -means algorithm. We also test a random clustering to see if the “quality” of the clustering has an impact on the quality of the solutions we obtain with our cluster-based algorithm (Algorithm 1).

The objective of this section is to present how we apply the clustering methods (e.g., hierarchical clustering, spectral clustering, k -means and random clusters) on our data and the results obtained when using them. These clustering algorithms take as input either a similarity or a distance matrix based on the data points to be clustered, which in our case are the sorting centers (see Section 3.4 for justification). To construct these matrices, we use two types of routing data: demands and physical distances between sites. A formal discussion of this construction can be found in Section 3.1.

We use the clustering algorithms to construct a small number of clusters (i.e., 2, 3 or 4 clusters). We then have to solve the problems inside each cluster which are called intracuster problems. We also have to solve an intercluster problem to take into account the demands with the origin in one cluster and the destination in another one. Finally we merge the solutions of all these subproblems (intracuster and intercluster problems) (see the k -Clusters Algorithm, i.e. Algorithm 1). While creating the clusters, a major concern is not to have the number of intercluster demands explode (see Section 2.2). Moreover, we use only a small number of clusters because if there are too many clusters, there are too few possibilities for beneficial consolidation of demands. For instance, in our algorithm, an intracuster demand in cluster 1 cannot be consolidated with an intracuster demand in cluster 2 on an intercluster path as neither of these demands can use this type of path. In contrast, in an optimal solution one could use this type of path (on Figure 4.2, the intracuster demand in cluster 1 stays in the blue zone, intracuster demand in cluster 2 stays in the red zone, thus they cannot be consolidated on a path using a black plain link). Therefore, there is a tradeoff between the number of clusters and the opportunity to consolidate demands. The more clusters, the more decrease in the intracuster subproblems size, but many clusters makes the size of the intercluster grow and prevents consolidation as each intracuster subproblem is solved independently.

3.1 Input Data Provided to the Clustering Algorithms

The input data for these clustering algorithms are similarity or distance matrices based on the demands and on the physical distances (see Section 3.2 of Chapter 2). As a reminder, a demand is an (origin, destination, volume) triple and the demand matrix is the matrix that contains the volume from site i to site j . The physical distance between two sites is

the distance a vehicle has to travel between these sites.

We apply these algorithms to cluster only the sorting centers. Then each delivery depot belongs to the cluster of its associated sorting center. To do this, we respect the catchment areas (mentioned in Section 1.3 of Chapter 2), so each delivery depot associated to the same sorting center should belong to the same cluster. For the demands, if their origin and destination are in the same cluster then the demand is in this cluster, otherwise the demand is in the intercluster subproblem.

Each of the classical algorithms considered has the objective of maximizing pairwise similarity or minimizing pairwise distance of points within a cluster. In our case, we aim to minimize the physical distances between sites inside the clusters, hence for the algorithms which maximize similarity inside a cluster we want this similarity measure to be inversely proportional to the distance which represents the physical distance. On the other hand, we want to maximize the intracluster demand volumes and minimize the intercluster demand volumes. We aim to minimize the volume of demands between the clusters because this allows to limit the size of the intercluster routing problem and to facilitate its resolution. Therefore the demand matrix is a similarity matrix which needs in some cases a transformation to allow us to minimize the feature that we want.

Moreover, some of these algorithms require the similarity matrix to be positive semi-definite matrix so we need to symmetrize our input matrix. Hence as a preprocessing step we prepare a symmetric matrix of the physical distances between sites and a symmetric matrix representing the demands. Since we are only clustering sorting centers, the demand of a sorting center is the **aggregate demand** of all of its associated delivery depots. Since the input demands are directed, the aggregate demands are not symmetric: They are directed and vary in volume. When the input require a PSD matrix, we symmetrize the matrix by adding together the demand from i to j and the demand from j to i (which exists because this is an aggregate demand).

In order to facilitate the intercluster solving, it can also be favorable to balance the cluster weights in terms of demands, otherwise there would be a large instance which would not be solvable optimally. Thus we want each cluster to contain at most one big sorting center (i.e., a “big” sorting center is one with a large amount of incoming and outgoing demands) and sites that are physically close. Intuitively we want the smaller sorting centers to be gathered around one bigger sorting center. So we use a weight matrix which gives a weight to each sorting center. For example, if the weight is 2, the site is represented twice in the matrix to cluster, but the two instances can’t be separated. Thus we combine both information of demands and physical distances as input to apply the clustering algorithms allowing to weight input data.

Formally, each clustering algorithm takes as input one or two of the following matrices (with $i, j \in S_{s,c} \times S_{s,c}$):

- D^{dist} the physical distance matrix with distance for vehicles to go from site i to site j ,
- S^{dist} the physical similarity matrix: $S_{i,j}^{dist} = \max_{k,l \in S_{s,c} \times S_{s,c}} (D_{k,l}^{dist}) - D_{i,j}^{dist}$,

- $D^{dist\ sym}$ the symmetrized physical distance matrix: $D_{i,j}^{dist\ sym} = (D_{i,j}^{dist} + D_{j,i}^{dist})/2$,
- $S^{dist\ sym}$ the symmetrized physical similarity matrix: $S_{i,j}^{dist\ sym} = (S_{i,j}^{dist} + S_{j,i}^{dist})/2$.
- S^{dem} the aggregate demand matrix of demand between site i and site j ,
- D^{dem} the demand distance matrix: $D_{i,j}^{dem} = \max_{k,l \in S_{s.c} \times S_{s.c}} (S_{k,l}^{dem}) - S_{i,j}^{dem}$,
- $S^{dem\ sym}$ the symmetrized demand matrix: $S_{i,j}^{dem\ sym} = S_{i,j}^{dem} + S_{j,i}^{dem}$,
- W^{dem} the vector to weight the sites with the demands: $W_i^{dem} = \sum_{k \in S_{s.c}} S_{i,k}^{dem} + \sum_{k \in S_{s.c}} S_{k,i}^{dem}$.

We use implementations from the Scikit-Learn library for the aforementioned clustering algorithms (except the algorithm to construct random clusters).

3.2 Input Data for Each Clustering Algorithm

Hierarchical Clustering

The first clustering algorithm we apply is hierarchical clustering. We use the function *AgglomerativeClustering* of Scikit-learn (sklearn.cluster), which provides an implementation of hierarchical clustering. It allows to choose the number of clusters expected and the algorithm stops when it reaches this number. This implementation does not require a symmetric input matrix hence we use the matrices S^{dem} and S^{dist} as input.

Spectral Clustering

We also use the spectral clustering algorithm which approximately minimizes the exchanges between the clusters output with respect to an input similarity matrix, which is required to be positive semi-definite matrix. Therefore we need to symmetrize our input matrices. We use the function *SpectralClustering* of Scikit-learn (sklearn.cluster) on the matrices $S^{dem\ sym}$ and $S^{dist\ sym}$.

k -Means Clustering

The previous two clusterings take as input similarity matrices. The k -means clustering is a local clustering, which uses a distance matrix as input. We use the implementation *k-means* of Scikit-learn (sklearn.cluster) on the matrices D^{dist} and D^{dem} . We also use the feature of Scikit-Learn *kmeans* to combine both information of demands and physical distances and use D^{dist} and W^{dem} as input to apply the algorithm.

Semi-Random Clustering

In order to see whether or not the clustering algorithm and the similarity used have an impact, we also do a **semi-random clustering** in which the clusters of the sorting centers are decided randomly. For this “clustering algorithm” we uniformly assign a number between 1 and k to each sorting center. This number represents the cluster in which the site is. Then we put the delivery depots in the same cluster as their associated sorting center. We also tried to use a totally random clustering on all the sites: sorting centers and delivery depots. But with this algorithm, the clusters are very much unbalanced as the sites are not all sources and destinations thus the demands are badly divided as there are clusters with few sorting centers and clusters with many sorting centers.

Synthesis: Data Types and Clustering Algorithm

Table 4.1 is a summary of all the clustering methods we tested as input for the k -Clusters Algorithm (see Algorithm 1) in order to find out which clustering method provides the best results for the tackled optimization problem. These different clustering algorithms take various data types as input and this is captured in Table 4.1.

Table 4.1: The clustering algorithms tested and their input

Algorithm \ Similarity	Physical distance	Demand	Both distance and demand	No input
Hierarchical Clustering	S^{dist}	S^{dem}		
Spectral Clustering	$S^{dist\ sym}$	$S^{dem\ sym}$		
k-means	D^{dist}	D^{dem}	D^{dist} and W^{dem}	
Semi-random				x
Totally random				x

3.3 Output Clusterings

In this section, we show the clusterings obtained for all the algorithms tested in Section 4 for Configuration 6. Some outputs are similar for this configuration, but they are not similar for all the configurations.

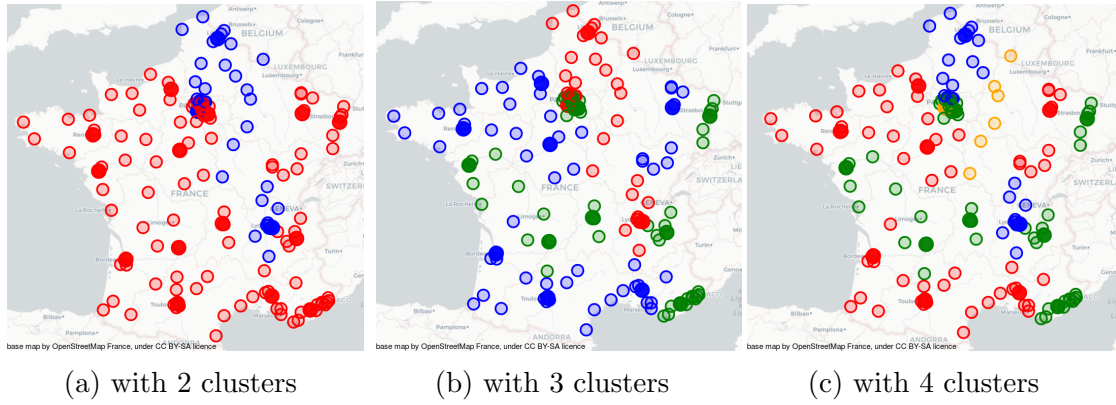


Figure 4.3: The hierarchical clustering on demands

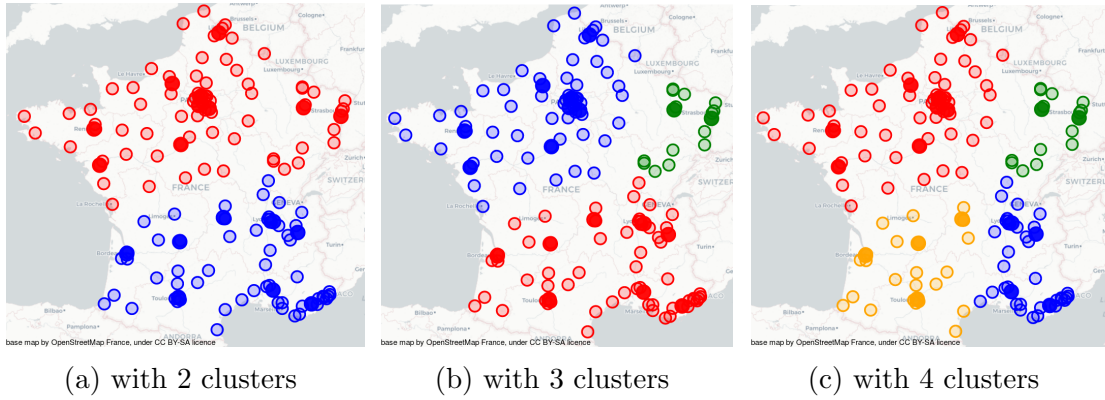


Figure 4.4: The hierarchical clustering on distances

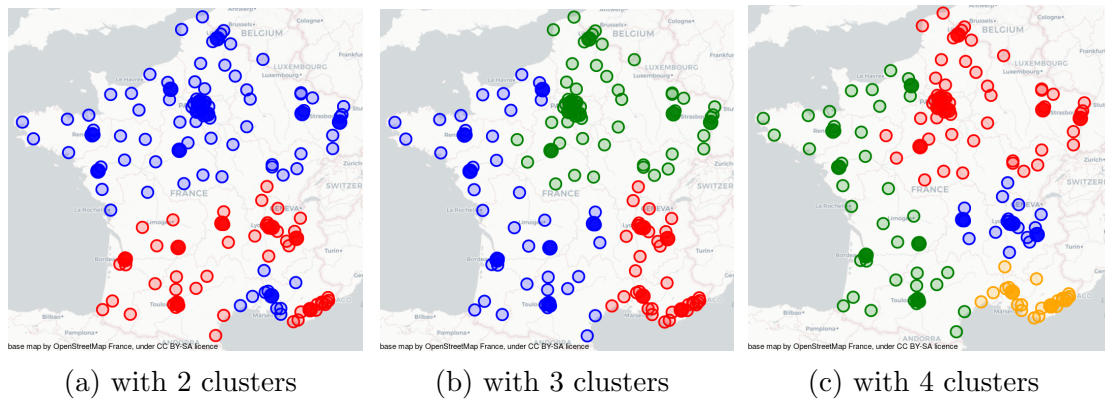


Figure 4.5: The spectral clustering on demands

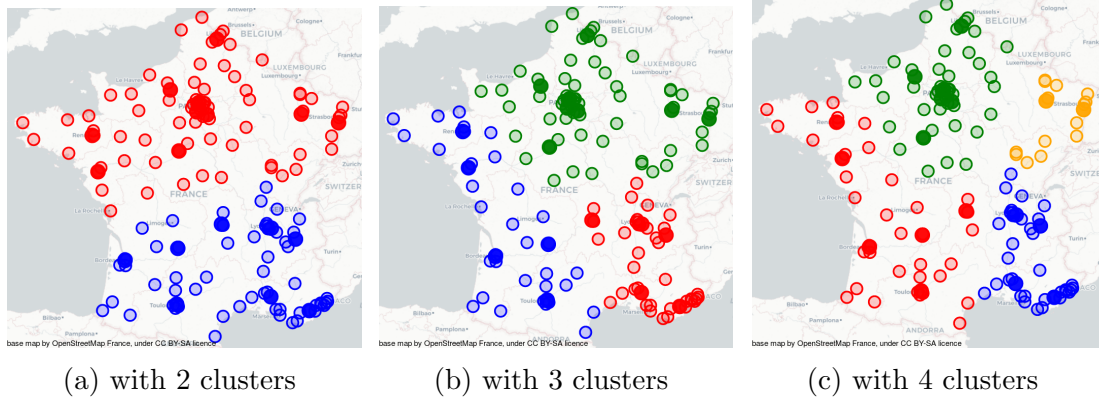


Figure 4.6: The spectral clustering on distances

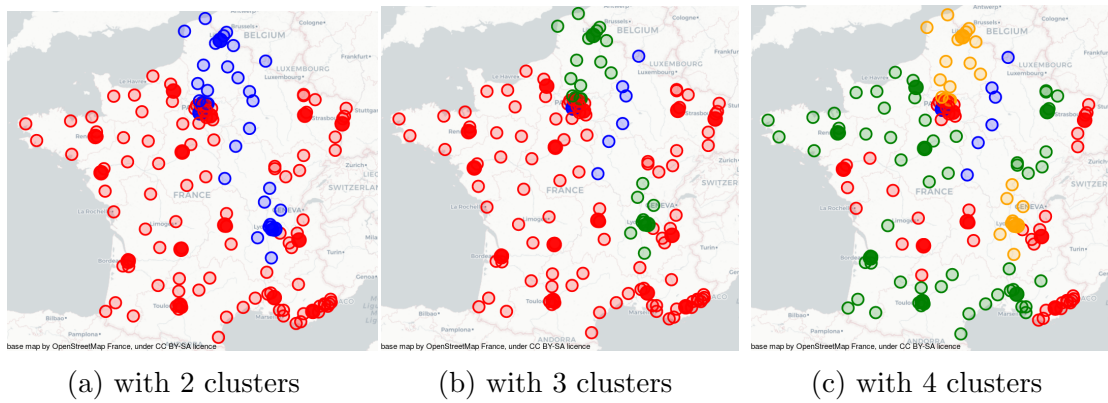


Figure 4.7: The k -means clustering on demands

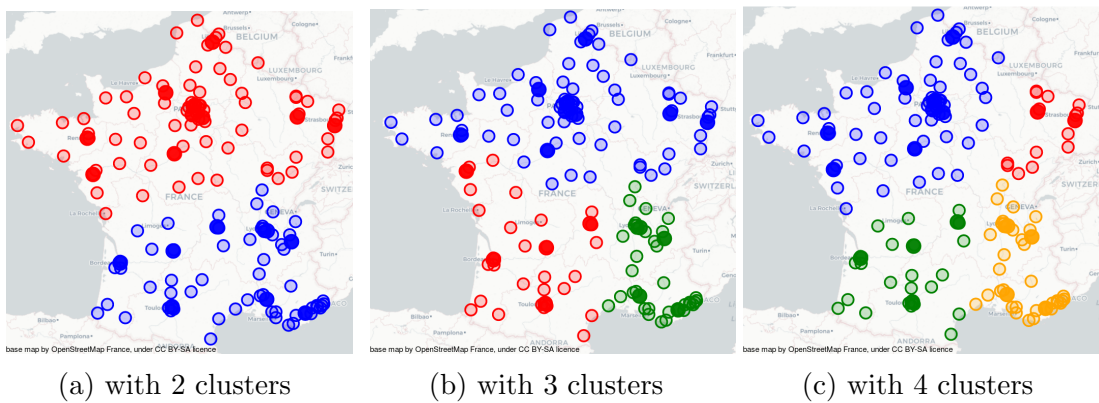


Figure 4.8: The k -means clustering on distances

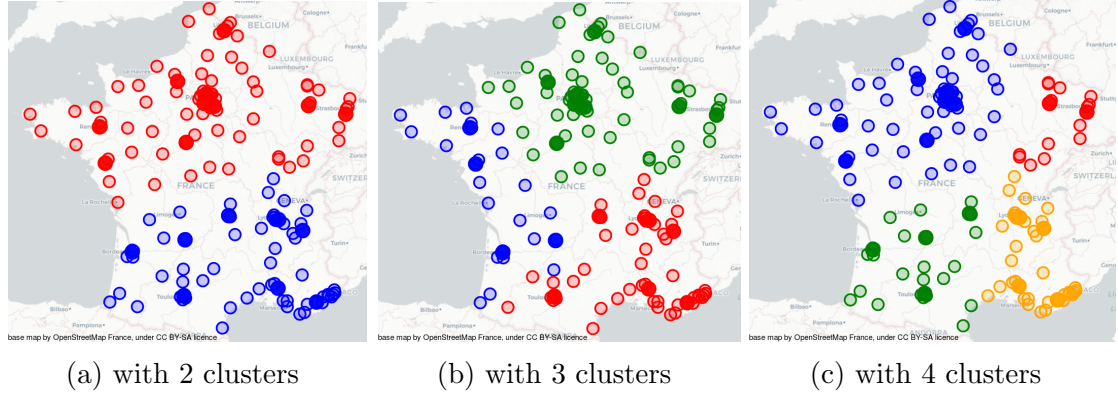


Figure 4.9: The k -means clustering on demand and distances

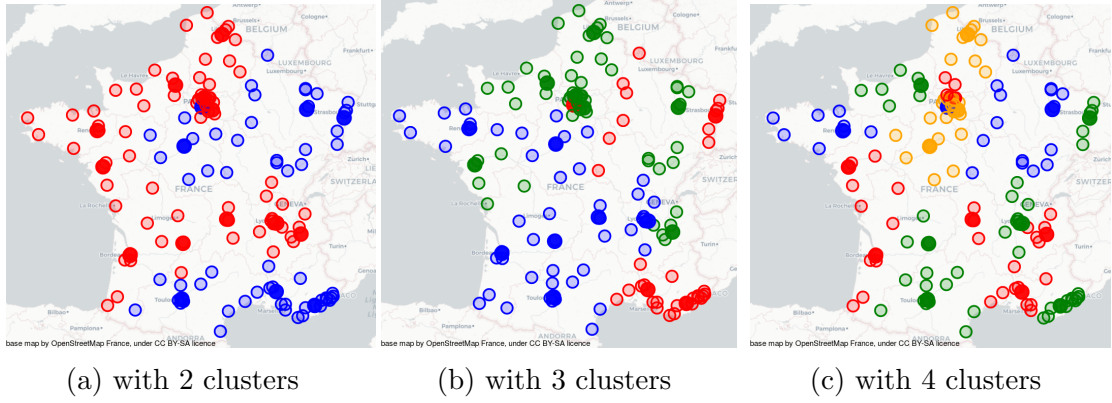


Figure 4.10: The semi random clustering

Comparison of the Methods

The hierarchical clustering algorithm, in our case the hierarchical agglomerative clustering algorithm, provides clusters forming a hierarchy. We notice in Figures 4.3 and 4.4 that the increase in the number of clusters is more or less made by dividing a cluster into two other ones. The spectral clustering algorithm (Figures 4.5 and Figures 4.6) provides clusterings which are not nested: It is not hierarchical. The k -means algorithm applied on demands (Figures 4.7) seems to provided nested clusters. However, it is not the case when applied on distances, or when applied on demands but on other configurations. The fact that it is the case for the dataset presented here is a coincidence. The semi-random clustering algorithm has gathered the catchment areas randomly. The clusters (Figures 4.10) are not nested, and they are not interpretable.

Comparison of the Similarities/Input

It is no surprise that the clusterings based on the physical distance (see Figures 4.4, 4.6, 4.8 and 4.9) are more visually pleasant. The clusters are connected: They are not divided

into pieces. The clusterings based on demands (see Figures 4.3, 4.5 and 4.7) are not connected and are less interpretable. But we can notice that the k -means algorithm applied on demands and the hierarchical algorithm applied on demands for two clusters provide the same result (Figures 4.3a and 4.7a). The clustering based on demand and distances (Figures 4.9) is connected as it is based on distance. The semi-random clustering (Figures 4.10) is the less connected and the more spread across the country. The distance-based clustering methods always give the same output for two clusters (see Figures 4.4a, 4.6a, 4.8a and 4.9a). For three clusters, we have twice the same output (spectral clustering (Figure 4.6b) and k -means on demands and distances (Figure 4.9b)), the two others being different. We can notice that for four clusters, with distance based clustering, the east side of the country is always clustered the same while the west changes once (over the four clustering methods).

Some clusterings seem more balanced and relevant when we look at them. We will see in Section 4 if the good-looking clustering provide good results when used as the first step of the k -Clusters Algorithm.

3.4 Confirmation of the Catchment Areas Utility

In order to confirm the interest of respecting the catchment area while using a divide and conquer method, we try to cluster on all the sites. It gives us interesting information.

We test a totally random clustering (which independently assigns each site to one of the k clusters uniformly at random) and compare it to classical clustering methods (see Table 4.1). This totally random clustering has a high probability of separating delivery depots from their associated sorting centers. When this happens, it leads to cases for which not all the parcels can be delivered with our cluster-based algorithm. For instance, if the origin sorting center and delivery depots are far apart (no direct path possible) and the associated sorting center is not in the same cluster of either of these two sites, then we cannot guarantee that the package would be delivered as there might not be any possible operational paths for this demand in its subproblem in our model. Our computational results show that many parcels could not be delivered with a totally random clustering. We can conclude that it is important that the clustering respects the catchment areas of the sorting centers. Hence we will not discuss the totally random clustering in the rest of this section. Results obtained with the semi-random clustering are however presented in the rest of the section as it allows to verify if the respect of the catchment area is the only feature of a clustering needed to have good results or if the use of demand-based or physical-distance-based similarity function improves the results.

We just justified that it is better that the clusterings respect the catchment areas as it ensures not to separate any sorting center from its associated delivery depot. Here is another practical reason why we cluster only on sorting centers. We tried to cluster all the sites with the classical clustering methods. When applied with a demand-based similarity function, the hierarchical clustering and the k -means algorithms provide the following result: The sites are divided between the sorting centers on one hand and the delivery depots on the other hand. Therefore, all the demands are from a site in

one cluster (the origin sorting center) to a site in another cluster (the delivery depot of destination): All the demands are intercluster. Thus the intracluster subproblems are empty and the intercluster problem has the same size as the original problem. There is no divide-and-conquer as there is no division of demands. Thus, using all the sites as the data points for the demand-based clustering algorithms does not result in a problem that is easier to handle for the hierarchical clustering and k -means algorithms.

The only clustering algorithm which provides exploitable results when applied on the demands is the spectral clustering. These results are showed in Chapter 3 Section 1.4. Moreover, our goal is to compare clustering algorithms with demand-based and physical-distance-based similarity functions. That is why in both cases we cluster only the sorting centers and put the delivery depots in the cluster of their associated sorting center.

3.5 Structural Properties of the Data Revealed via Clustering

In this Section, we introduce key aspects of the data relative to the k -Clusters Algorithm revealed when we applied the clustering methods combined with the inputs presented in this Section. As a reminder, in this algorithm, we want to divide carefully the original problem into subproblems so the solving of the subproblems and the merging of these solutions makes a good solution for the original problem. In Section 2, we said that the intercluster problem is a key aspect of the algorithm as our objective is to divide the problem in a way which minimizes the intercluster problem size and balances the sizes of the intracluster problems.

Balance of Clusters: When we apply the clustering techniques presented in Table 4.1, we do not always obtain well-balanced clusters in terms of number of variables when we cut the graph into only two, three or four pieces. We would like to have well-balanced clusters because it would minimize the overall computation time and reduce the size of the problem more effectively. Indeed, if we have a cluster with 10 sites and a cluster with 140 sites, the one with 140 sites is still too large to be manageable in a reasonable amount of time. We test these unbalanced clusterings anyway in order to confirm that the best clustering does indeed have well-balanced clusters.

Table 4.2 shows what percentages of the demands are in each cluster and shows which methods provide unbalanced clusters. In this table, the numbers of demands is not the volume of demands (number of parcels, which we used to create the clusters), but the number of arcs in the demand graph. It appears here as it is a good indicator of the size of the problem to solve with a solver.

Table 4.2: Balance of demands in clusters

Number of clusters	Demands in the intracusters (%)			Demands in the intercluster(%)		
	min	avg	max	min	avg	max
2 clusters	0.6	28.6	85.0	14.4	42.8	50.0
3 clusters	0.3	13.3	59.0	37.5	60.2	65.9
4 clusters	0.2	8.2	30.7	59.5	67.0	73.3

Table 4.2 shows that the more clusters there are, the more intercluster demands there are. To have the best-balanced clusters and therefore have all the benefits of our divide-and-conquer approach, we need to have fewer clusters. Thus, we do not use more than four clusters. However, we test 2, 3 and 4 clusters to be sure that in practice the results confirm this intuition that fewer clusters are better.

No Sparse Cut: The first approach we study is to cluster using a similarity function based on the demands. Indeed, we want to divide our network into parts which have little exchanges between them in order to prevent the intercluster problem size from exploding. We notice it is hard to divide into such parts: The intercluster problem is always quite large when the clusters are of balanced sizes. Indeed, one particularity of the long-haul parcel transportation problem is that parcels are sent from all the sorting centers to all the delivery depots. In practice, with our datasets, around 44% of the demands cross between the clusters on average. Thus, the demand graph does not appear to have a “sparse” cut. This means that there are no sets of sites that are sending most of their parcels among themselves: They are all sending a large number of packages to other sites outside the set. Intuitively, it means that there are many operational paths from any site on one side of a cut to any site on the other side of the cut, thus the intercluster problem has lots of variables. Because of all these path variables, the performance of the MILP solver is very poor on the intercluster problem. Therefore it is necessary to use a heuristic for this intercluster problem. It is a natural approach to restrict the number of such paths. We have decided to choose operational paths which exhibit consolidation.

Note that when the clusters are unbalanced, it makes the intercluster problem very small. However, this is not a sparse cut: It is useless in practice due to the fact that nearly all the sites are in one cluster. Therefore we only consider balanced clusters when we discuss the notion of a sparse cut.

Reduction of the Problem Size via Clustering: Table 4.3 shows that the total number of variables for each problem is quite large (see the row “Merging model” in Table 4.3). As a large part of them are integer variables, optimal solutions cannot be computed in reasonable time with the MILP on the complete instance. But the number of variables in the intracuster models is small enough to allow an exact solving. The intercluster model has more variables but we use a heuristic to solve it efficiently thus many of these variables are fixed to zero. The merging model has as many variables

as the model for a computation without cluster, but in this model the operational path variables are fixed to the values we recorded while solving the successive models which represent the solution we built (only the operational link variables are recomputed in this phase). Thanks to this reduction of the model size, it is solvable in less time.

Table 4.3: Reduction with 2 clusters

Name of the model	Number of demands			Number of sites			Number of variables			Number of constraints		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
Cluster 1	58	1115	3084	31	127	269	10K	240K	1043K	2K	12K	36K
Cluster 2	22	720	3150	23	98	225	7K	113K	1017K	1K	8K	31K
Intercluster	521	1388	2047	154	225	292	232K	529K	896K	8K	15K	26K
Merging model	2312	3223	4063	154	225	292	613K	1115K	1612K	16K	31K	42K

3.6 Synthesis

In this Section, we detail the application of the clustering methods described in Section 1 on the data of our case study. To find the best partition of sites for the k -Clusters Algorithm presented in Section 2, we compare various clustering algorithms (random, spectral clustering, hierarchical clustering, k -means) and apply them on the demand graph and the physical distance graph. When designing these clusterings, we respect the catchment areas as it is the best option. The similarities and distances needed by each algorithm are described in Section 3.2. Applying the clustering methods on the data of our case study showed that there are no sparse cut in the data we work with. Thus the intercluster subproblem can never be sparse. However, some clustering methods allow to have better balanced clusterings than others which permits to reduce better the size of the subproblems. In the next section, we test the algorithm on the clusterings displayed in Section 3.3. The objective is to determine which clustering method, similarity and number of clusters results in the better solution for the k -Clusters Algorithm.

4 Simulations and Results

In this section we present computational experiments on various realistic input scenarios provided by a postal company. These simulations and their results are presented next.

4.1 Simulation Parameters

The datasets used for our tests are real data provided by a company. There are six different configurations of the network which makes six datasets. They are described in Table 3.1 in Section 1 of Chapter 3. As a reminder, a dataset is made of a list of sites, a list of demands (origin, destination, number of parcels) between these sites and the costs of all the logistics and transportation operations (sorting, entering sites, leaving sites, etc.). Table 4.3 of Section 3.5, shows the number of variables in each subproblem for all

the methods and the six datasets in the case of two clusters. The test environment is composed of a Linux server with 32 CPU and 150 Gbytes of RAM. The solver used is CPLEX 12.8. The Scikit-Learn library is used to build the clusters (except the algorithm to construct random clusters).

We test each clustering algorithm presented in Section 3 on each of the 6 configurations. Moreover each of the clustering algorithms is set to output two, three or four clusters for each distance function presented in Table 4.1. For each combination (clustering method, distance function, number of clusters, network configuration) we run the k -Clusters Algorithm (Algorithm 1) once. Therefore we launch 162 simulations with the k -Clusters Algorithm plus the simulations without clustering, solved with the global MILP applied on the whole problem instance.

We allow the MILP solver to run for up to one hour to solve each subproblem (intracluster, intercluster, merging, complete). This is a choice we made which constitutes a trade-off between having a good solution and having quickly a solution. Indeed, in ten minutes we can also have solutions but they are far worse. We are able to fix nearly all the variables before entering the merging stage of the k -Clusters Algorithm, which requires then only a few seconds to reach the optimum. Thus the total computing time is less than what could be expected. For example, the tests with two clusters could take up to four hours (first intracluster, second intracluster, intercluster, merging stage) but in practice each takes only three hours (one hour for each intracluster problem, one hour for the intercluster problem). It is the same with the tests with four clusters which could take up to six hours, but take only five hours. In the following tables, the time to build the models for each subproblem and to save the results are not taken into account as they are negligible compared to the solving stages. Note that generating the MILP for the solver (construction of the paths and links) takes around five seconds, while saving in a database the whole transportation plan takes around twenty minutes ².

At the end of the merging stage (step d)), we obtain a **computed solution** to the problem (see columns 3 and 4 in following tables), which is injected as an initial solution in a one hour execution of the LHPTP-MILP for refinement (step e)). The value of this **refined solution** is given in column 5 in following tables.

Note that the time values shown in the following are the total amount of time needed to obtain the solutions (i.e., on a single machine, without parallelization). As the intracluster subproblems are independent, they could be launched in parallel.

4.2 Evaluation of Results

In the network on which we optimize the parcel transportation, we have from two to four hubs fixed by the transportation managers according to the configurations (see Chapter 3 Section 1.1). In this section, we decided to release this constraint and to consider that all the sorting centers can be inner-hubs in order to enhance the possibilities of consolidation and to confirm if the four hubs chosen by the transportation managers the right ones.

²This is currently done with sqlite3 and could be optimized.

Results with Strategy 2

In this section, we use the Strategy 2 (defined in Chapter 3 Section 2) which consists in considering all the sorting centers as inner-hubs and to respect the catchment areas provided. This means the MILP has more variables than if we used only the hubs chosen by the transportation managers but this allows to propose more operational paths to the solver and therefore leads to better solutions.

Number of Clusters: The first question we consider is: What is the number of clusters which produces the best quality solution? We present in Table 4.4 a comparison of the solutions and gaps found with the different numbers of clusters we tested in the k -Clusters Algorithm (Algorithm 1). In this table, we provide the minimum, the average and the maximum values in each column, the average being computed over all the configurations and all the clustering methods. Table 4.4 shows that the best results are obtained when we use 2 clusters. As a matter of fact the more clusters there are, the more the macro point of view is lost. Indeed, because we solve the intracluster subproblems independently, some possibilities of consolidation are lost. Moreover, the more clusters there are, the larger the intercluster problem is. This intercluster problem is solved with a heuristic in which some possible operational paths are removed to make the problem manageable (see Figure 4.2) but it also means the solution for this problem is sub-optimal. Therefore we want to divide to obtain smaller subproblems but we want to divide as little as possible because each division costs us a lot in terms of consolidation and intercluster solution optimality.

Table 4.4: Comparison of the number of clusters

					Computed solution via clustering					Refined solution						
Number of clusters	Total time (h)				Objective (K)			Gap (%)		Objective (K)			Gap (%)			
	min	avg	max		min	avg	max	min	avg	max	min	avg	max			
MILP (no cluster)	4				—			—		377	408	426	10.3	14.3	20.5	
MILP (no cluster)	5				—			—		377	408	426	10.3	14.1	19.5	
2 clusters	3.0	3.8	4.1		379	409	430	8.3	14.3	19.2	379	407	430	7.8	13.9	19.2
3 clusters	3.0	3.9	5.0		379	410	428	9.5	14.5	18.8	376	408	427	8.0	14.1	18.7
4 clusters	3.0	3.7	5.0		382	411	438	9.5	14.9	20.7	382	410	438	7.1	14.6	20.7

The first two lines of Table 4.4 give the results of the LHPTP-MILP run for respectively four hours and five hours. This solution is obtained without clustering and therefore the solution is presented in the column refined solution as it is the solution obtained at the end of the total computation time. Note that the results of the MILP are slightly different as the one given in Table 3.13 (in Chapter 3 Section 3.1) as in Table 3.13 the MILP is run for six hours. The other lines of Table 4.4 give the solution obtained at the end of step d) in the column “Computed solution via clustering” and at the end of step e) in column “Refined solution”. The column “total time” gives the time to obtain the refined solution. The time to have the solution computed via clustering is the total time minus one hour.

Table 4.4 shows that the k -Clusters Algorithm provides in three hours to five hours solutions which are slightly better than a solver without cluster in the same computation time. The difference is not huge but the interest is that the solution computed via clustering can be computed on computer with less memory. Indeed, CPLEX requires lots of memory to solve the complete problem but for the subproblems and the problem with fixed variables, it is less intensive. Another advantage of this approach is that it deals with smaller MILP which could be handled by open source solvers. Indeed, CPLEX is needed to solve large MILP but for small instances it is not necessary anymore.

As the best number of clusters is two clusters, in the following, we present in-depth results for our algorithm with 2 clusters for all the configurations.

Best Clustering: The second question to consider is: What is the best clustering method and with which distance function ? Table 4.5 allows to compare the methods tested in the case of two clusters.

Table 4.5: All configurations with 2 clusters

Name	Total time (h)			Computed solution via clustering			Refined solution		
				Objective (K)			Objective (K)		
	min	avg	max	min	avg	max	min	avg	max
MILP (no cluster)	4			—			377	408	426
Hier Dem	3.0	3.5	4.0	386	410	428	9.4	14.7	18.9
Hier Dist	4.0	4.0	4.0	382	406	425	8.3	13.8	18.3
Spec Dem	3.0	3.5	4.0	392	412	430	8.9	15.0	19.2
Spec Dist	4.0	4.0	4.0	379	407	424	9.1	13.9	18.1
k -means Dem	3.0	3.5	4.1	381	408	422	9.4	14.3	17.5
k -means Dist	4.0	4.0	4.0	385	409	426	8.6	14.4	18.5
k -means both	4.0	4.0	4.0	383	407	425	8.4	14.1	18.4
Semi-random	4.0	4.0	4.0	387	408	421	8.7	14.2	17.6
				387	406	420	8.0	13.8	16.9

The values in the table are the minimum, the average and the maximum for the 6 configurations. We can notice that all the clustering methods provide results which are slightly better than what the solver provides within four hours. As the results in Table 4.5 are very close we cannot point out one winner but we can conclude that the k -Clusters Algorithm provides a better solution than the solver in less time when the clustering respects the catchment areas of the sorting centers. Table 4.5 gives the value of the objective of the solution built thanks to the k -Clusters Algorithm and then the value of the objective of the solution obtained by injecting this solution as a first solution in one hour of solver computation (refined solution on the complete problem). We can see that the solutions built are better when the clustering is done according to the physical distance. It is probably because the transportation costs, which are the main part of the cost function, are closely related to the physical distance between sites.

The spectral clustering based on distances and the k -means based on both distances and demands seems to be the best when we look at the value of the objective after the extra one hour of solver computation. But it is not the case when we look at the computed

solution via clustering, obtained before the refinement step. In this case, the Hierarchical Clustering on the distances provides the best results.

With more heterogeneous datasets, one could highlight the qualities and drawbacks of each clustering method. However, as we work with realistic specifications and a well-defined use case, all our datasets are on the same network and have roughly the same properties (see Chapter 3 Section 1.3).

Results with Strategy 1

In this section, we use the Strategy 1 (defined in Chapter 3 Section 2) which consists in using only the hubs chosen by the transportation managers as inner-hubs. This allows to have less variables (which helps the solver to go faster) and to compare with the results obtained with Strategy 2.

Table 4.6: All configurations with 2 clusters and Strategy 1

Name	Total time (h)	Computed solution via clustering			Refined solution		
		Objective (K)	Gap (%)		Objective (K)	Gap (%)	
	min avg max	min avg max	min avg max		min avg max	min avg max	
MILP (no cluster)	4	—	—		377 413 439	9.7 12.9 17.0	
Spec Dist	4.0 4.0 4.1	388 415 431	8.2 13.3 17.2	388 410 422	6.8 12.4 15.0		
<i>k</i> -means both	4.0 4.0 4.0	388 415 432	8.4 13.3 16.9	380 411 422	8.0 12.6 16.9		

Table 4.6 shows that the *k*-Clusters Algorithm provides slightly better solutions than the global MILP for the Strategy 1 (respecting the catchment areas and the inner-hubs provided by the transportation managers). With this strategy, the gaps (to the best lower bound found for Strategy 1) are lower as the models have less variables than with Strategy 2. But the objective values are higher as these variables allow to decrease the cost of the objective value.

4.3 Synthesis

This heuristic algorithm allows us to have results comparable to the ones obtained with an exact method in terms of quality. It allows however for less memory usage and it allows us to use a freely-available MILP solver, which can handle problems of smaller sizes than expensive commercial solvers. Moreover, we have proven that the less clusters the better the results. Thus there is no need to test the algorithm with more clusters as the best results are obtained with only two clusters and then the increase in the number of clusters degrades the solution quality. Finally, there is no “better clustering method” for these datasets and this algorithm. Indeed, all the clustering methods give comparable results which proves that the key aspect is to divide into balanced clusters. The semi-random clustering gives satisfying results as well as the clusterings obtained with well-known algorithms applied on two key features of the data. Thus we decided not to create a tailored clustering method for our case study as it did not seemed promising.

5 Conclusion

In this chapter, we apply a new divide-and-conquer approach based on clustering techniques to solve the Long-Haul Parcel Transportation Problem: the k -Clusters Algorithm. Indeed, the real data instances at stake are too large to be solved efficiently with a single global MILP. Thus we use several clustering techniques (spectral clustering, hierarchical clustering, k -means and random) to divide the problem into smaller subproblems that can be addressed with the MILP. This constitutes a tailored heuristic which permits us to solve the long-haul parcel transportation problem more accurately and more efficiently.

The k -Clusters Algorithm allows us to have results comparable to the ones obtained with an exact method in terms of quality. Its advantages are that it permits to use less memory and/or eventually a free solver. As we expected in Chapter 3, the Strategy 2 allows to have better results than the Strategy 1 with the k -Clusters Algorithm as it was already the case with only the MILP. There is no "better clustering method" for our case study, as all the clustering methods give comparable results which proves that the key aspect is to divide into balanced clusters and to respect the catchment areas. The semi-random clustering gives satisfying results as well as the clusterings obtained with well-known algorithms applied on two key features of the data: the physical distance and the demands. Thus creating a tailored clustering method for our case study does not seem promising. Finally, we have proven that the less clusters the better the results. Thus there is no need to test the algorithm with more clusters as the best results are obtained with only two clusters and then the increase in the number of clusters degrades the solution quality.

The division in subproblems allows to use the MILP which works well and seems promising. One weakness of the k -Clusters Algorithm is that the intracluster subproblems are solved before the intercluster subproblem and it impacts its resolution through the heuristic. It could be interesting to see what happens if we solve between the clusters first and then inside the clusters. This is not possible in the k -Clusters Algorithm (the intercluster problem is large and will not be solved efficiently with an MILP), but this can be done in another algorithm. It is what we do in the next chapter, with the Hierarchical Algorithm with Aggregate Demands.

Chapter 5

Hierarchical Approach: Algorithm with Aggregation of Demands

Contents

1	Background and Previous Works	107
2	A Hierarchical Algorithm with Aggregation of Demands	107
2.1	Approach to Solving the LHPTP	108
2.2	Detailed Steps of the Algorithm	110
3	Simulations and Results	114
3.1	Simulation Parameters	114
3.2	Evaluation of Results	115
	Results with Strategy 2	115
	Results with Strategy 1	117
3.3	Synthesis	117
3.4	Comparison of the k -Clusters Algorithm and the Hierarchical Algorithm	118
4	Conclusion	120

In this chapter, we propose an approach that takes advantage of the hierarchical structure of our network: we divide the whole problem into tractable subproblems in the most natural way possible, which follows the two-level structure of the network. We were inspired by the idea of Baumung and Gunduz [BG15] of sending directly large demands and consolidating residual demands. It seems natural to maximize the truck filling rate and minimize the number of parcels sorted as sorting has a cost per parcel. However, we do not want to impose a mandatory sorting (as in [LM14]) on all the demands, as it would incur extra costs. To apply this idea to our approach for solving the LHPTP, we send directly (bypassing all sorting operations) trucks filled more than a truck filling rate threshold, rather than only those that are fully filled. Roughly speaking, we define the demands above this threshold as **large demands** and the ones below this threshold as **residual demands**.

We will test various thresholds of truck filling rate in order to measure the impact of this filling rate on the quality of the solutions. The truck filling rate threshold will help to find a good trade-off between the benefits of routing demands directly and the disadvantages of sending suboptimally filled trucks.

In this hierarchical algorithm, the routing of both large and residual demands are first optimized separately, and then we see if combining this solutions can improve the global solution. Indeed, for the large demands, there is more than one delivery option (due to the options for vehicles and due to the double deliveries ¹). Moreover, as we want to optimize the global problem of routing all the demands we do not want to completely separate these subproblems of routing demands above and under the threshold, especially while taking into account the optimization of balancing with empty vehicles. Note that all these subproblems are small enough to be solved optimally with the LHPTP-MILP (see Chapter 2 Section 3.3).

The LHPTP has two levels which can both be bypassed by direct paths, and a heuristic is used to optimize this decision only for bypassing the two sortings. However, this heuristic is not used within the inner level. Indeed, the optimization on the inner level can be solved exactly at the optimum, thus there is no need for a heuristic to decide if an inner-level demand need to be consolidated in an inner-hub or not.

In Section 1, we present background and previous work on two-level networks to position the Hierarchical Algorithm with Aggregation of Demands in this literature. This new algorithm is described in Section 2. Finally, in Section 3 the results of this algorithm on our case study are presented as well as a comparison of the results of the two divide-and-conquer algorithms.

This algorithm has been presented at the ROADEF 2021 conference [GCH⁺21] and will be presented at APMS 2021 conference [GCHN21].

¹Double deliveries are defined in Chapter 2 Section 1.4. This is when a truck with two containers delivers the two containers in two different depots.

1 Background and Previous Works

Hierarchical networks [Cur88, CRC86] are composed of sites which are of different types of varying centrality. The sites of a same type (inner-hubs, sorting centers, depots) form a **layer** or stage. Each pair of layers constitutes one **level** of the hierarchical network and is sometimes referred to as an **echelon**. Note that our hierarchical network is a non-hierarchical hub-and-spoke network, as stated in Chapter 1 Section 3.1, because a hierarchical hub-and-spoke network, according to Lin and Chen [LC04], assigns each spoke site to one hub site. Here, we are given a two-level hierarchical network in the sense that we have three types of sites (inner-hubs, sorting centers and delivery depots). In this chapter we explore how to exploit this hierarchical network structure to design good transportation plans ².

Two-echelon networks are a special case of multi-echelon networks in which the network is made of two levels. That is why this type of problem is also called two-level network optimization. Note that two-level optimization must not be confused with bi-level programming (which is a larger and different subject). Surveys on two-echelon problems have been written by Gonzalez-Feliu [GF11] and Cuda et al. [CGS15]. In this last paper, the two-echelon routing problems are divided into three categories but our problem does not fit in any of these problems which are mainly related to vehicle routing on one or two levels of the network (Two-Echelon Location Routing Problem, Two-Echelon Vehicle Routing Problem (2E-VRP), Truck and Trailer Routing Problem). It is not a Two-Echelon Location Routing Problem as the sites in the network are fixed. Nor is it a Two-Echelon Vehicle Routing Problem (2E-VRP) as the parcels are in bulk in containers so there are no delivery tour to consider as a container is completely emptied at destination. Moreover, in [CGS15] they define the 2E-VRP with no cost associated with the use of any depot and any satellite, which is not our case as we have sorting costs. Finally, it is not a Truck and Trailer Routing Problem (TTRP) which is another real-life problem in which some sites can receive only truck with one container while others can receive trucks with one or two containers. This last type of problem is the closest to our problem but it routes a single good with a fleet to manage, which is different from our framework.

The LHPTP is not a VRP but it does have three types of sites which are linked by a two-level network. To the best of our knowledge, there is no two-echelon optimization for service network design as most of the applications of two-echelon optimization deal with VRP and not with more general SND problems.

2 A Hierarchical Algorithm with Aggregation of Demands

As we saw in Chapter 2 Section 3.3, we can formulate our problem as an MILP and an optimal solution for this MILP yields an optimal solution for LHPTP. However, we

²A transportation plan is defined in Chapter 2 Section 2.4.

cannot solve this MILP in a reasonable amount of time when considering realistic sized datasets. Therefore, the main idea of our algorithm is to divide the problem into smaller subproblems, each of which we can solve optimally via an MILP or other means, and then to add the solutions to these subproblems together to obtain a final solution of good quality, although we note that it can be suboptimal.

2.1 Approach to Solving the LHPTP

Our idea is first find an optimal transportation plan on the inner level of the network (see Chapter ref/chap2), then we optimize the extension of this transportation plan on the outer level of the network, and finally, in the last steps of the algorithm, we combine and refine these solutions to obtain a transportation plan for the whole network.

Recall that the original demands of our problem are from sorting centers to delivery depots. For the inner problem, we create aggregate demands which are demands from sorting centers to sorting centers (in order to separate the two levels). In our algorithm, we require that the last sorting will be done in the sorting center to which the delivery depot of the destination is associated. Thus, we can aggregate the demands which were from sorting centers to delivery depots to make them from sorting center to sorting center to separate the initial problem into subproblems. An **aggregate demand** is the sum of demands from a sorting center to all the corresponding delivery depots of another (destination) sorting center.

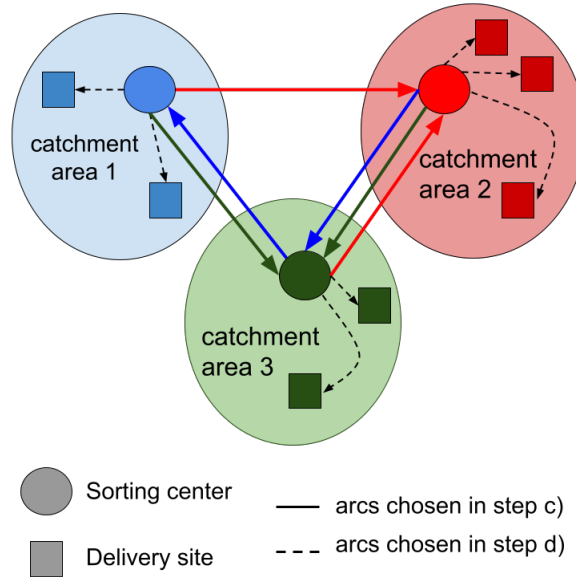


Figure 5.1: The two levels of the resolution

The **Hierarchical Algorithm with Aggregation of Demands** (also called Hierarchical Algorithm for short) chooses the links between sorting centers first (plain arcs in Figure 5.1), then it chooses the links from sorting centers to delivery depots in each

zone around each sorting center (dotted arcs on Figure 5.1) and finally it assembles the solutions. But if we aggregate all the demands, we lose the possibility of using direct paths from a sorting center to a delivery depot, which have been proven to be useful [O’K98, ZW02]. We therefore consider the following approach (represented in Figure 5.2): If a demand is large enough to nearly fill a truck, then we consider sending a truck with this demand directly from the origin to the destination. Thus, our algorithm first splits the demands into large demands, whose operational paths are actually determined in the last step (deferred demands), and residual demands whose operational paths are specified by the transportation plan we construct on the inner and outer levels of the network. These residual demands are either routed through an inner-hub, in which case they are subject to an additional sorting, or they are routed directly from their initial sorting center to their final sorting center. This latter determination is made during the single call to the appropriate MILP on a set of aggregate residual demands.

The proposed algorithm follows a kind of two-echelon approach but it is not a true two-echelon approach for two main reasons. First, in a first step we are only optimizing the transportation plan between sorting centers. The demand of a sorting center is the **aggregate** demand of all of its corresponding delivery depots. It is a made-up demand. Secondly, if we simply aggregate all the demands into demands from sorting center to sorting center, it will make a pure hierarchical two-echelon approach. But it forces direct paths to be unused. However, we know this type of operational path is used in an optimal solution. So we choose to use some direct paths that we think are useful.

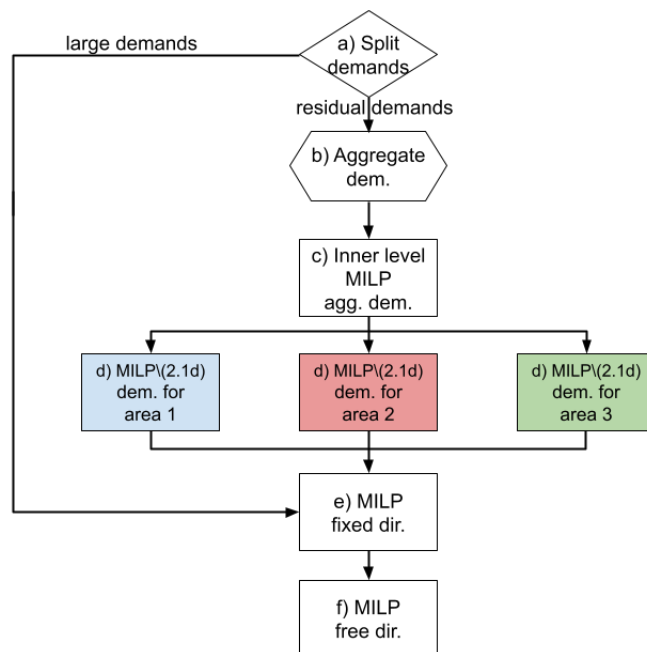


Figure 5.2: Structure of the algorithm

Note that our problem has two levels which can both be bypassed by direct paths, and we use a heuristic to optimize this decision only for bypassing the two sorting operations.

However, we do not use this heuristic within the inner level. Indeed, the optimization on the inner level reaches a 1% gap with the MILP, thus there is no need for a heuristic to decide if an aggregate demand (from sorting to sorting center) need to be consolidated in an inner-hub or not.

2.2 Detailed Steps of the Algorithm

Now we detail each of the seven steps of our proposed resolution approach, expressed in Algorithm 2.

Algorithm 2: Hierarchical Algorithm with Aggregation of Demands

- 1 a) Split demands
 - 2 for each demand d do
 - 3 if $v_d < \sigma \cdot C_{veh}$ then
 - 4 Add d to the set of residual demands
 - 5 else
 - 6 $k = \lceil v_d / C_{veh} \rceil$
 - 7 if $k \cdot \sigma \cdot C_{veh} \leq v_d \leq k \cdot C_{veh}$ then
 - 8 Add d to the set of large demands
 - 9 else
 - 10 Split d into a large demand of volume $(k - 1)C_{veh}$ and a residual demand (twin demands)
 - 11 b) Aggregate residual demands
 - 12 for each of the residual demand d do
 - 13 Aggregate it with the demands with the same associated sorting center as destination
 - 14 c) Solve the aggregate subproblem (inner level)
 - 15 Solve the MILP which is made only of operational links between sorting centers
 - 16 d) Extend for each catchment area (outer level)
 - 17 for each one of the n catchment areas do
 - 18 Solve the MILP without constraints (1d) to route the demands towards their catchment area
 - 19 e) Add the solutions of all residual demands, large demands (on direct paths) and integrate the empty repositioning (global level)
 - 20 Solve the MILP with all the residual demands and with only the chosen operational paths to optimize the vehicle flow. The large demands are enforced to use a direct paths. (for up to 1h)
 - 21 f) Optimize the routing of the large demands
 - 22 Solve the MILP with all the demands, the solution of the previous MILP fixed and the option for the large demands to follow a direct path or an already chosen operational path. (for up to 1h)
-

a) Splitting Demand Volumes into Two Sets

The first step of the algorithm is to split the demand volumes into two sets. We call the first set of demands the **large** demands (i.e., essentially, those whose volume is above a given threshold $\sigma \cdot C$, where $\sigma \in (0, 1]$ and C is the capacity of a container) and the remaining set the **residual** demands, which is standard terminology [BG15]. Recall that a demand is a triple (s_d, t_d, v_d) , where s_d is the origin, t_d is the destination and v_d is the volume (i.e., number of parcels to be routed from s_d to t_d). We define $k := \lceil v_d/C \rceil$. For each demand d , if $v_d \leq \sigma \cdot C$, then demand d is a residual demand. Otherwise, if $v_d \in [k \cdot \sigma \cdot C, k \cdot C]$, then we can use k containers, so d is a large demand. In the last case, we have $v_d \in [(k-1) \cdot C, k \cdot \sigma \cdot C]$, the demand d is split into twin demands: A large demand of volume $(k-1) \cdot C$ and a residual demand of volume $v_d - (k-1) \cdot C$. Note that sometimes $k \cdot \sigma \cdot C < (k-1) \cdot C$ and in this case, v_d is a large demand.

For example, suppose the given threshold is 60% of the volume of a container, which is 1000. And suppose we have a demand whose volume of 1100 is a bit larger than the capacity of a container, then this demand will be split into one large demand with volume 1000 and one residual demand with volume 100. In the set of residual demands, we have demands from sorting centers to delivery depots, all of which are smaller than the given threshold. Large demands are unique in their set but they can have a twin residual demand.

In the algorithm, the two sets of demands will be handled differently. The large demands will be set aside and possibly routed directly at the end of the algorithm (see step f). The residual demands will be routed according to the solution of MILP (see Equation (2.1) and steps b to d). Note that only after determining the routing of the residual demands, will we decide how to route the large demands: Either they will be routed directly, or they will be combined and sent with the residual demands if there is leftover capacity.

b) Aggregating the Residual Demands

In order to separate the optimization of the routing on the two levels of the network, we first need to have demands on the inner level of the network (from sorting centers to sorting centers). To achieve that, we require that the last sorting for a parcel is done in the sorting center associated with the delivery depot of its destination. Due to this requirement, we can aggregate residual demands which are sent to a common outer-hub.

The **aggregate demands** represent the addition of the residual demands from a single origin to all the corresponding delivery depots of each sorting center. Let's denote $S_{d,d}^j$ the set of delivery depots in the catchment area of the sorting center j . The volume of the aggregate demand from the sorting center i to the sorting center j is $v_{d_{i,j}} = \sum_{k \in S_{d,d}^j} v_{d_{i,k}}$.

For example, let's consider two sorting centers A and B (see Figure 5.3). We aggregate all residual demands whose origin is A and destination/delivery depot is in the catchment area of B , resulting in an aggregate demand from A to B whose volume is the sum of the volumes of these residual demands. Note that every aggregate demand is a truncated

demand which ends up in a sorting center (whereas original demands end in a delivery depot).

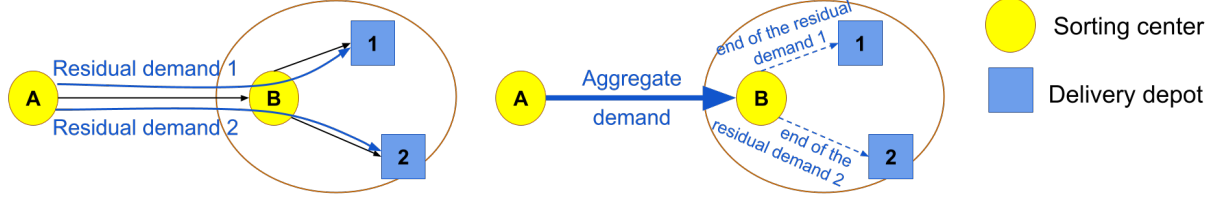


Figure 5.3: Aggregation of demands by catchment area

c) Finding the Routing for Aggregate (Residual) Demands Between Sorting Centers

Now that we have aggregate (residual) demands going from sorting center to sorting center, the problem size is reduced. We have indeed fewer possible destinations: Initially, we had around 200 sites and now we have the same problem of routing parcels but with around 20 sites. So we can use the MILP (see Equation (2.1)) to optimize this smaller problem as it is the same problem as before but on a smaller instance. Note that the MILP includes the design-balance constraint (see Equation (2.1d)). The MILP applied on the network composed of the sorting centers and the aggregate demands provides a transportation plan for these truncated demands.

d) Extending the Routing for Each Catchment Area

At this point, we have one or more operational paths chosen for each aggregate demand. In this step, we disaggregate the aggregate demands and turn them back into residual demands ending in delivery depots. We have one or more operational paths chosen for each group of residual demands from its origin to the sorting center associated with their delivery depot and we need to complete these paths so that each parcel reaches its final destination. This is not trivial for two reasons: (i) paths between sorting centers can be shared by several aggregate demands; (ii) paths between sorting centers reserved for each aggregate demand can be shared by several residual demands.

(i) If in step c, two aggregate demands were sharing a vehicle on a link, and one of these aggregate demands is split over more than one operational paths (for instance, if we are routing 20% of an aggregate demand A on an operational path which shares a link with the operational path chosen for an aggregate demand B), then we do not want the parcels for one of these two demands (demand A in our example) to use completely the vehicle capacity allowed on this link. It could happen as the catchment areas are solved separately and that will be a problem in the next steps. So we need to enforce the parcels to use the links chosen for each aggregate demand by the MILP of step c with respect of the capacities fixed in this MILP (to avoid mixing the catchment areas).

(ii) We also need to reassign the global capacity of the paths chosen for the aggregate demand and make sure that it is well shared between residual demands. Indeed, the capacity reserved on each path may be used by several residual demands. For instance, let us consider an aggregate demand A which is disaggregated in residual demands A_1 and A_2 . If in step c the paths p_a and p_b has been chosen for A , the residual demand A_1 might use both paths p_a and p_b while A_2 only uses the path p_b . In this case, the capacity reserved for the path p_b has to be shared.

To take into account these two points, we add the following new constraint to the MILP in which we denote $P_{p_{agg}}^{desc}$ the set of operational paths descending from the aggregate path p_{agg} (which is associated to the aggregated demand d_{agg}). Note that $x_{p_{agg}}^{d_{agg}}$ is defined similarly as x_p^d in Chapter 3 Section 3.3. It represents the rate of the parcel flow of the demand d_{agg} on the path p_{agg} .

$$\forall x_{p_{agg}}^{d_{agg}}, \sum_{p \in P_{p_{agg}}^{desc}} v_{x_p^d} x_p^d \leq v_{agg} x_{p_{agg}}^{d_{agg}} \quad (5.1)$$

We solve this problem separately for each catchment area. For each set of residual demands, whose final destination belongs to the same catchment area, we use the MILP\((2.1d)\) to extend the operational path (from Step c) from final sorting center to final destination. For each subproblem, we consider the network of all sorting centers (network considered in the previous step) and the delivery depots of the relevant catchment area. The idea is we restrict the variables so that the operational paths output by the MILP\((2.1d)\) solution will follow the operational paths already chosen in the previous step between sorting centers. In this model, we do not put the design-balance constraint (Equation (2.1d)) as we want the repositioning of the vehicles to be optimized globally and not by catchment area. Note that the design-balance constraint means that the number of outgoing and incoming trucks must be equal for each site in the course of a day (e.g., it ensures that all vehicles are returned to the sorting centers).

At the end of this step, we have operational paths for all the residual demands. But we are not finished yet, since these operational paths are in several transportation plans, one for each catchment area, and our goal is to find a single global transportation plan.

e) Merging the Residual Demands Solutions with the Large Demands on Direct Paths and Integrating the Empty Repositioning

In this step we merge the solutions (found in step d) to deliver the residual demands and optimize the large demands paths and the empty repositioning. We fix in the MILP the operational paths variables for parcels (the x_p^d variables in the MILP) and the variables representing the number of vehicles on each operational link (the y_l^{veh} variables in the MILP) accumulated in the previous steps. Indeed, in the previous steps we have optimized the inner level and these links are not recomputed, neither are the links which allows to end the parcel path in the delivery depot. As in each subproblem we optimize one catchment area (or the inner level) these links cannot be used in two separate

subproblems, therefore there is no need to recompute it as it was the case in Chapter 4 Section 2. We add the solutions for each catchment area obtained in the previous steps to have a global vision of the whole network. In this resolution the empty repositioning from delivery depots to sorting centers are optimized as the design-balance constraint of the MILP is activated. We do so for two reasons: first to optimize simultaneously the empty repositioning of trucks and the direct paths and secondly to have a transportation plan which answer the original problem in order to compare it with the one obtained at the end of the next step. Note that, there is more than one direct path option for each demand (because of the two types of vehicles and of the double deliveries). Thus there is an optimization done by the MILP solver for the large demands even if they are allow to use only a direct path. This gives us the global solution built from the solutions to the subproblems solved in the previous steps, with the large demands sent on direct paths. At this stage, we have what we call **Sol fixed dir.** (solution with fixed direct paths) in Table 5.1 for the results analysis.

f) Optimizing Operational Paths for the Large Demands

In this step we fix the variables to the solution obtained at the previous step for the residual demands and we allow the large demands to follow either a direct path or the operational path used by their residual twin demand if there is enough “leftover” space. Note that if there is no residual twin demand, the large demand will be necessarily routed on a direct path. This gives us the final global solution built with the Hierarchical Algorithm with Aggregation of Demands, called **Built solution** in Table 5.1.

3 Simulations and Results

In this section we present simulations on realistic input scenarios provided by a postal company. The results of these simulations are presented next.

3.1 Simulation Parameters

The datasets used for our tests are the same as those used in Chapter 4 Section 4. There are six different configurations of the network which makes six datasets provided by a postal company. They are described in Table 3.1 in Chapter 3 Section 1. As a reminder, a dataset is made of a list of sites, a list of demands (origin, destination, number of parcels) between these sites and the costs of all the logistics and transportation operations (sorting, entering sites, leaving sites, etc.).

The test environment is composed of a Linux server with 32 CPU and 150 Gbytes of RAM. The solver used is CPLEX 12.8.

We test different values for the truck filling rate threshold (σ in the Algorithm 2). As it is a percentage, we test all values from 100% to 10% with a step of 10 on each of the six configurations. Therefore we launch 60 simulations with the Hierarchical Algorithm

with Aggregation of Demands plus the simulations without clustering, solved with the global MILP applied on the whole problem instance.

We allow the MILP solver to run for up to one hour to solve each subproblem (inner level, extension to each catchment area, merging). This is a choice we made which constitutes a trade-off between having a good solution and quickly finding a solution. Indeed, in ten minutes we can also have solutions but they are far worse. The solver can stop before the one hour of computation if it reaches a gap of 0.01%. This is nearly always the case for the subproblem addressing the extension to the catchment areas (step d), which usually takes less than one minute to reach this gap. As we are able to fix nearly all the variables before entering the merging stage of the algorithm, it requires only a few seconds to reach this 0.01% gap. Thus the total computing time is much less than what could be expected. Indeed, the only subproblem which systematically takes one hour of computation is the inner level problem. In the following tables, the time to build the models for each subproblem and to save the results are not taken into account as they are negligible compared to the solving stages. Note that generating the MILP for the solver (construction of the paths and links) takes around 5 seconds, while saving in a database the whole transportation plan takes around 20 minutes.

3.2 Evaluation of Results

In the network on which we optimize the parcel transportation, we have from two to four hubs fixed by the transportation managers, depending on the configuration. In this section, we decided to first place ourselves in the framework of the Strategy 2 (defined in Section 2 of Chapter 3). It consists in ignoring this constraint and to consider that all the sorting centers can be inner-hubs in order to enhance the possibilities of consolidation and to confirm if the 4 hubs chosen by the transportation managers are the right ones. Then, we check what happens if we respect this constraint (Strategy 1). Finally, we compare the Hierarchical Algorithm and the k -Clusters Algorithm.

Results with Strategy 2

We first compare results obtained with several thresholds (σ in Algorithm 2) for splitting demand volumes into large demands and residual ones. In this algorithm we solve each MILP with a 1h time limit and all sorting centers are considered as inner-hubs. Note that the gap presented in the table represents the distance to the best lower bound computed by the solver for an exact resolution (see first line of Table 5.1).

The results in Table 5.1 show that compared to the LHPTP-MILP when run without any heuristic for 6 hours (lines with threshold none), the Hierarchical Algorithm with Aggregation of Demands can provide better solution values in five to six times less computational time for the appropriate thresholds. Note that the line “threshold: none and time 6 hours” presents the average, minimum and maximum values of the values given in Table 3.13 (in Chapter 3 Section 3.1) to give a reminder and to allow the comparison.

We see that the threshold which provides the best results is $60\% \pm 10\%$. We can also observe that there are not many difference between the solution obtained in step

e) and step f), even if the optimal solution is reached most of the time (with respect to the variables fixed). As a reminder, in step e), the large demands are enforced to use a direct paths, while in step f) they are offered the opportunity to use the path of their twin residual demand. But the lower the threshold is, the less there are demands which are split into twin large and residual demands. If a large demand does not have any twin demand, we do not offer any possibility to use a path which is not direct. For these demands, there is no difference between the solutions obtained at the end of steps e and f. Note that steps e) and f) are run for much less than one dedicated hour.

Table 5.1: Comparison of the thresholds

Thres. (%)	Time (h)			Sol fixed dir. e)			Built sol f)			Gap (%)			Filling rate (no empty vehicles)			Global fill. rate (%)		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max
none	1			—			377	477	807	11.6	21.5	58.7	47.5	66.5	72.2	31.6	41.0	43.7
none	6			—			377	406	423	10.2	13.7	17.4	70.0	71.3	73.2	42.7	43.4	44.4
100	1.0	1.1	1.3	376	415	444	376	414	442	13.4	15.5	16.7	82.6	84.0	86.0	52.4	54.1	55.7
90	1.0	1.1	1.2	375	411	437	375	411	436	13.2	14.7	15.9	82.1	84.2	86.4	52.0	54.0	55.5
80	1.0	1.2	1.2	374	406	429	374	406	429	12.2	13.8	14.9	81.6	83.9	85.9	51.4	53.4	54.8
70	1.0	1.1	1.2	369	405	437	369	405	435	11.8	13.5	15.1	80.2	82.4	84.9	50.8	52.1	53.6
60	1.0	1.1	1.2	367	399	419	367	399	419	10.1	12.2	13.9	79.5	81.8	83.6	49.3	50.9	52.0
50	1.1	1.2	1.3	374	401	419	374	401	419	10.0	12.7	14.3	76.7	78.7	80.3	47.4	48.3	48.7
40	1.1	1.2	1.5	374	406	426	374	406	426	11.5	13.8	15.7	75.4	76.0	77.1	45.5	45.7	46.4
30	1.0	1.2	1.5	393	421	435	393	421	435	13.3	16.9	19.1	70.1	70.9	72.7	41.1	41.6	42.1
20	1.0	1.3	1.5	436	463	480	436	463	480	19.9	24.5	27.7	60.1	61.5	63.3	34.1	34.7	35.4
10	1.1	1.3	1.5	559	583	627	559	583	627	33.0	39.9	44.6	42.7	45.6	49.4	23.0	24.6	26.6

When the threshold is 60%, the vehicle filling rates with and without considering repositioning of empty vehicles are respectively 50,9% and 81,8%. Slightly higher vehicle filling rates (like 84,2%) can be obtained with higher filling rate thresholds but at the expense of sorting cost for more consolidation. Both of these pieces of information are important. Indeed, one could think that maximizing the filling rate of trucks will lead to cost minimization, because when trucks are more filled, we need less trucks. That is why we study the filling rate without the empty vehicles. But it is profitable to send a truck filled at 60% on a direct path as it allows to save both empty repositioning of vehicles and sorting costs. That is why it is necessary to look at the global filling rate. These simulations showed that the higher filling rate does not correspond to the lowest cost of objective value. Indeed, a higher filling rate means more sorting and sorting has a cost.

In the transportation plans associated to the solutions presented in Table 5.1, we note that nearly all the sorting centers are used as inner-hubs. It is probably because the demands are small compared to the vehicle capacity and because the sorting costs are not very high compared to the kilometric cost. Consolidation of demands is then useful as it allows to reduce the number of vehicles used, even if this leads to more sorting operations. This might however not be the case with much higher sorting costs. We tested this hypothesis by comparing results obtained with several sorting costs considering a threshold of 60% for the activation of direct paths.

We can see in Table 5.2 that the number of inner-hubs used decreases when the sorting cost increases. We can also see in Table 5.2 that when the sorting costs increase, the solving is much faster as sorting is a less affordable option to consider in the mathematical

model and the MILP can perform more efficient cuts in the solution space. But the interesting output of this test is that with a higher sorting cost, five hubs are used in average and by very small demands. Moreover, these inner-hubs do not include all the inner-hubs selected by transportation managers.

Table 5.2: Comparison of the sorting costs with $\sigma = 60\%$

Sorting cost	Time (h)			Sol fixed dir. e)			Built sol f)			Gap ³ (%)			Used inner-hubs		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max	min	avg	max
normal	1.0	1.2	1.3	359	405	436	358	405	436	10.1	12.2	13.9	13	14.7	17
10 \times normal	0.0	0.1	0.2	1635	1733	1806	1634	1731	1803	78.6	79.8	80.7	8	8.5	10
20 \times normal	0.0	0.1	0.2	3030	3188	3330	3026	3182	3324	88.3	89.0	89.6	4	5.2	7

All the inner-hubs are selected by the Hierarchical Algorithm with the “normal” costs. Thanks to that, the Hierarchical Algorithm provides better solutions and can be an interesting decision support tool for the managers.

Results with Strategy 1

In this section, we use only the inner-hubs chosen by the transportation managers as inner-hubs, it corresponds to the Strategy 1 (defined in Chapter 3 Section 2). This allows to have less variables (which helps the solver to go faster) and to compare with the results obtained with Strategy 2 to see if with this strategy it is possible to have best results.

Table 5.3: Comparison of the chosen inner-hubs with the Hierarchical Algorithm

Inner-hubs	Time (h)			Sol fixed dir. e)			Built sol f)			Gap (%)		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
all s. centers (S2)	1.0	1.2	1.3	359	405	436	358	405	436	9.3	11.9	13.9
only the fixed ones (S1)	0.5	1.0	1.3	377	412	435	377	411	435	9.0	13.3	16.1

The gap in Table 5.3 is the gap to the best lower bound computed with the Strategy 2. This table confirms that Strategy 2 provides better results than Strategy 1.

3.3 Synthesis

All of these simulations show that it is profitable to divide this type of naturally hierarchical problem into smaller problems which can be solved optimally with exact methods. Indeed, it allows to solve larger instances of the problem. Moreover, the simulations show that in parcel transportation, due to sorting costs and empty balancing of vehicles, the optimization of the truck filling does not obviously result in cost optimization. Indeed, the cheaper solutions are not the ones in which the trucks have the highest filling rate. The best threshold to decide if a demand should be offered the opportunity to use a direct

³Gap to the lower bound of the MILP without heuristic with the same variables.

path is when it fills 60% of a container, not 100%. This result can seem counter-intuitive for someone on the ground with only a local point of view, as they do not have a global perspective, which leads to a different conclusion. But it allows to optimize the network globally and to minimize the total cost.

We also show that the advantages of consolidation in inner-hubs depends strongly on the sorting costs and kilometric costs. Thus the optimal threshold found empirically depends on the datasets and would need to be recomputed if the algorithm were to be applied on another dataset. Finally, while the simulations were on very large datasets, without access to even larger datasets, it is nevertheless hard to see how to handle issues related to scaling the size of the input data.

With the data we used, provided by a postal company, greater truck filling rates do not correspond to the cheapest solution. It means, in general, that in long-haul parcel transportation networks, it is not always better to send fully-filled trucks.

3.4 Comparison of the k -Clusters Algorithm and the Hierarchical Algorithm

In this section, we compare the best results obtained with each one of the heuristics we developed. For the Hierarchical Algorithm with Aggregation of Demands, this corresponds to a truck-filling rate threshold σ of 60% (see Section 3.2) and for the k -Clusters Algorithm this corresponds to the k -means algorithm with two clusters and with demands and distances as input (see Section 4.2 of Chapter 4). All tests (including using MILP alone) are run with the Strategy 2 which respects the catchment areas and uses all the sorting centers as inner-hubs.

Table 5.4: Comparison of the algorithms on Strategy 2 (in one hour and four hours)

Algorithm	Time (h)	Objective value			Gap ⁴ (%)			Filling rate (%)		
	min avg max	min	avg	max	min	avg	max	min	avg	max
LHPTP-MILP	1	377	477	807	11.6	21.5	58.7	31.6	41.0	43.7
LHPTP-MILP	4	377	408	426	10.3	14.3	20.5	42.3	43.3	43.9
k -Clusters Algorithm	4.0 4.0 4.0	383	405	425	8.3	13.6	18.3	40.0	41.5	42.9
Hierarchical Algorithm	1.0 1.1 1.2	367	399	419	10.1	12.2	13.9	49.3	50.9	52.0

Table 5.4 shows that on average, both algorithms beat the MILP alone. We can see however that the k -Clusters Algorithm (run for four hours) allows to find smaller gaps in some cases than the MILP alone or the Hierarchical Algorithm with Aggregation of Demands (run for one hour). This appears more clearly in Figure 5.4. The Hierarchical Algorithm has a smaller deviation from the lower bound on the 6 configurations than the k -Clusters Algorithm or the MILP alone which can find solutions with much smaller gaps but also solutions with much higher gaps. In Table 5.4 the filling rate (with the

⁴This gap is computed with the best lower bound we have computed in six hours and called “lower bound 6h”.

empty repositionings) of the various algorithms can be compared. It is noteworthy that the solution of the k -Clusters Algorithm with a 8.3% gap has at most a filling rate of 42.9%. This confirms that increasing the filling rate does not provide the better solutions (due to the sorting costs). This is probably the reason why greedy algorithms (which maximize the truck filling) perform poorly on the LHPTP.

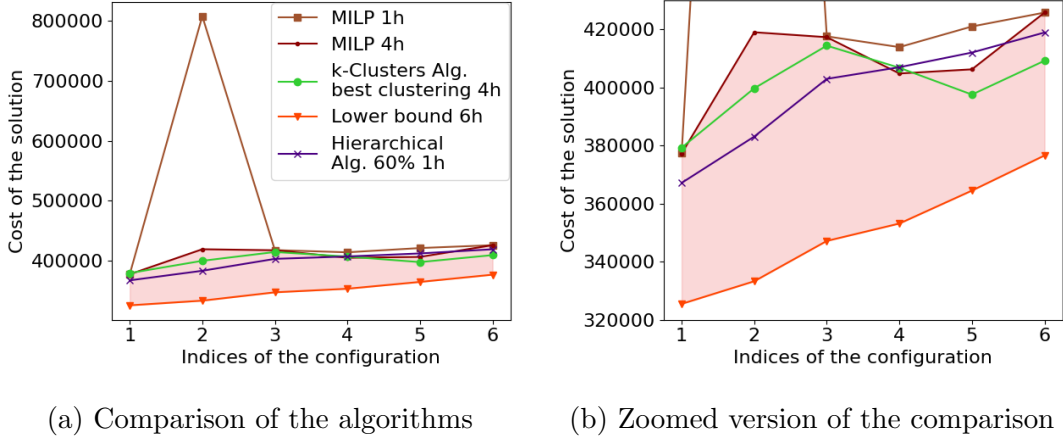


Figure 5.4: Comparison of the algorithms for the 6 configurations

Figure 5.4 also shows that the Hierarchical Algorithm, run for one hour, does not always beat the MILP alone if it is given enough time (6 hours). Indeed, the goal of our work is to beat the MILP alone (e. g., to have all our solutions in the colored zone of Figure 5.4). Thus we add a refining step (on the model of the refining step introduced for the k -Clusters Algorithm in Section 2 of Chapter 4). This step consists in adding an extra one hour run of the MILP which starts with the solution found by the Hierarchical Algorithm as a first solution. This gives a bias to the solver but it helps it not to struggle to start its search.

Table 5.5: Comparison of the algorithms on Strategy 2 (in two hours and six hours)

Algorithm	Time (h)	Objective value			Gap ⁵ (%)			Filling rate (%)		
	min avg max	min	avg	max	min	avg	max	min	avg	max
LHPTP-MILP	2	377	409	426	10.2	14.4	19.6	42.3	43.0	43.9
LHPTP-MILP	6	377	406	423	10.2	13.7	17.4	42.7	43.4	44.4
Hierarchical Algorithm	2.0 2.1 2.2	366	394	418	8.1	11.1	13.6	45.8	49.0	51.5

The results of the Hierarchical Algorithm with an extra refining step are compared to the MILP alone on a similar time (two hours) and on a six hours run in Table 5.5. It shows that the Hierarchical Algorithm with an extra refining step gives clearly better results than the MILP. The MILP is run for six hours since the results for a longer

⁵This gap is computed with the best lower bound we have computed in six hours and called “lower bound 6h”.

run are roughly the same: the solver becomes stuck. However, injecting the solution found with the Hierarchical Algorithm in an extra refining step allow to beat the MILP on the six configurations (see Figure 5.5). Note that the optimal solution of each of these configurations is somewhere between the best solution we can compute with all our algorithms and the lower bound that we have computed with the MILP solver. The refining step decreases the filling rate and the total cost of the solution (see Table 5.4 and 5.5). The trucks are less filled but better filled.

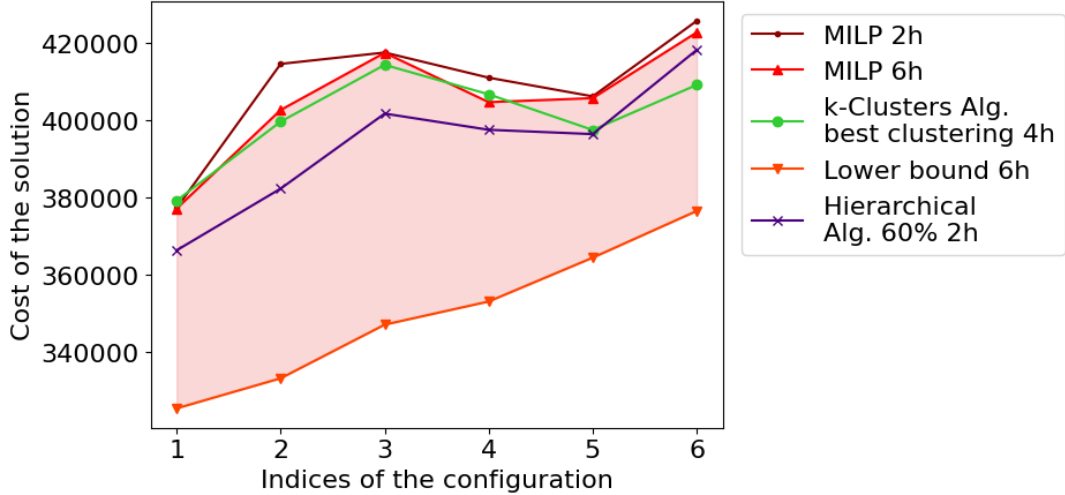


Figure 5.5: Comparison of the algorithms for the six configurations

The Hierarchical Algorithm is clearly better than the MILP alone (in two hours). Moreover, it provides better results than the k -Clusters Algorithm most of the time and it is faster. This might be because in the Hierarchical Algorithm the transportation plan found is very well optimized on the inner-level (gap of less than 0.5% in one hour) and then extended. This shows that in the k -Clusters Algorithm, the initial subproblem solved that is then extended is not as good a choice as in the Hierarchical Algorithm.

4 Conclusion

In this chapter, we present a heuristic to design a transportation network on a two-level hybrid hub-and-spoke network. We apply a new hierarchical approach to solve the Long-Haul Parcel Transportation Problem. As the two-levels hub-and-spoke network is also including the possibility to use direct paths, we choose to offer this option in our algorithm for large demands in order not to waste space in the vehicles and to save sorting costs. The proposed Hierarchical Algorithm with Aggregation of Demands is a hierarchical algorithm which divides the problem into subproblems solved through a multi-step process. In the algorithm, the large demands are separated from the residual demands and they are treated differently. The residual demands are aggregated to form aggregate demands which stays on the inner-level of the network while large demands

can bypass the levels by using direct paths. The inner-level subproblem can be solved with an MILP as it is smaller. The extension of its solution is also made thanks to an MILP. Finally, the addition of the large demands to the solution hierarchically built is made with an MILP.

The Hierarchical Algorithm with Aggregation of Demands is tested on data instances at the scale of a country. These data instances are too large to be solved efficiently with a single global MILP, but the use of the hierarchical nature of our network to divide it into smaller subproblems allows to have subproblems that can be solved with MILPs. We test various thresholds of truck filling rate for the large demands to find which ones suits our data the best. This constitutes a tailored heuristic which permits us to solve the LHPTP more accurately and more efficiently.

Conclusion and Perspectives

Contents

1	General Synthesis	123
2	Perspectives	125

1 General Synthesis

In this work, the Long-Haul Parcel Transportation Problem (LHPTP) is defined. It is an industrial problem containing specific strong industrial constraints and has not been addressed so far in the literature to the best of our knowledge. The delivery constraint states that all parcels must be delivered from their origin site (a sorting center) to their destination site (a delivery depot). The capacity constraint requires that the vehicle capacities must be respected. However, the number of vehicles the transportation plan can use is not bounded, neither is the number of vehicles between each pair of sites. The last constraint is the design-balance constraint. As we design a daily planning which will be applied the following day, the vehicles have to go back at their starting points before the end of the day in order to be used again the next day. All these constraints are not gathered in one standard optimization problem. The LHPTP is a point-to-point delivery problem, which consists in minimizing the cost of the long-haul transportation of parcels on a two-level hybrid hub-and-spoke network with three types of sites. In our case study, presented in Chapter 2, there are two types of vehicle to manage according to the number of containers and the daily repositioning of empty trucks.

The LHPTP is a Service Network Design problem with Asset Management. The network on which we design a transportation plan is precisely a two-level non-hierarchical hybrid hub-and-spoke network⁶ which is hybrid on the inner-level only. However, this is a two-level hierarchical network in the sense that we have three types of sites (inner-hubs, sorting centers and delivery depots). The LHPTP has distinct properties such as the fact that demands have both fixed origin and destination, and the sorting operation has a cost per parcel. It is a point-to-point delivery problem, with a sorting operation, which is one of the characteristics of the parcel transportation domain. The LHPTP has linear

⁶According to Lin and Chen [LC04] for the definition of non-hierarchical hub-and-spoke network given in Chapter 1 Section 3.1.

costs thus we develop an MILP formulation for it: the LHPTP-MILP. But this MILP formulation was not providing an optimal solution on the datasets provided by the postal company (presented in Chapter 3). Actually the gap between upper and lower bounds obtained from the MILP is quite consequential in terms of daily costs. Thus, there seems to be room for improvement, which is why we design and implement two algorithms with divide-and-conquer approaches, in which we partition the problem into smaller instances. In these two heuristics, we take advantage of the grouping by catchment areas as it allows to cluster sites more easily or to aggregate demands (see respectively Chapter 4 and 5).

The first algorithm we propose is the k -clusters Algorithm (see Chapter 4). In this algorithm, we apply a new divide-and-conquer approach based on clustering techniques to solve the LHPTP. Several clustering techniques (spectral clustering, hierarchical clustering, k -means and random) are used to divide the problem into smaller subproblems that can be addressed with the MILP. We test clustering on demands and/or physical distances, and try various numbers of clusters in order to find out which one fits the best to our case study.

Since all the clustering methods give comparable results, it does not seem to be worthwhile to find a better clustering algorithm in order to improve the algorithm. We observe that the key aspect is to divide into balanced clusters and to respect the catchment areas. The semi-random clustering gives satisfying results as well as the clusterings obtained with well-known algorithms applied on two key features of the data. Thus creating a tailored clustering method for our case study does not seem promising. Finally, we have shown that the fewer clusters the better the results, although we need at least two clusters to improve the MILP. Thus there is no need to test the algorithm with more than two clusters as the best results are obtained with only two clusters and then the increase in the number of clusters degrades the solution quality. The k -clusters Algorithm allows us to have results comparable to the ones obtained with an exact method in terms of quality. One weakness of the k -clusters Algorithm is that the intracluster subproblems are solved before the intercluster subproblem and it impacts its resolution through the heuristic. Thus we try the other way around in our second algorithm. We solve between the clusters first and then inside the clusters.

This is the main idea of the Hierarchical Algorithm with Aggregation of Demands. We solve first the inner level (between sorting centers) of our two-level network and then extend and complete this partial solution. This is made possible by the use of aggregate demands that we create to have demands on the inner level. To aggregate demands, we force a sorting of parcels which has a cost and has to be avoided for large demands. Thus we define a truck filling rate threshold to discriminate large demands and residual demands. We optimize the routing of the two types of demands, first separately and then together in a multi-step process in which the subproblems are solved via MILPs. The threshold impacts the performance of the Hierarchical Algorithm with Aggregate Demands, that is why we test various truck filling rate threshold values to find out which one is the best, in terms of solution quality obtained and computational time. The Hierarchical Algorithm gives cheaper solutions in less computational time than does directly solving the global MILP using a solver. Moreover, it gives better solutions than

the k -Clusters Algorithm most of the time while being faster.

The simulations show that it is efficient to divide instances from the case study into smaller problems which can be solved optimally with exact methods. Indeed, it allows to solve larger instances of the problem, so the divide-and-conquer approach used with an MILP is efficient for the LHPTP. Note that one can use another methods to solve the subproblems. We show that it is more efficient to aggregate the demands and to solve first the inner level of the problem and then extend the solutions than to cluster the sites and to solve first the intracluster subproblems and then the intercluster subproblem.

The next step is to give these solutions to the company that provided the datasets so they can see how these transportation plans behave in a real-world environment. This would allow us to have feedback from people from the ground on the transportation plans we propose. In particular, the new operational links which have been proven to be useful (direct links) need to be validated and the applicability of using new sorting centers as inner-hubs need to be confirmed. Moreover, it is not rare that when presented with transportation plans, the transportation managers realise that there were some constraints they forget to specify or some data which were inaccurate. Thus the feedback process will help us to get closer to the reality of long-haul parcel transportation.

2 Perspectives

Our two algorithms are designed to solve an industrial problem. In this section, we first offer research perspectives on the problem formulation. They are directly linked to the practical application of the proposed transportation plans. We then move on to potential generalization of our model to include more diverse industrial problems and point out which part of our methods could still be applied and the difficulties to overcome.

In all our tests, the last sorting is done in the associated sorting center of the destination (Strategy 2 presented in Chapter 3) except in the case of direct paths. This can remove some possible operational paths that could be chosen in a global optimal solution. An interesting option could be to relax this soft business constraint which assigns to each delivery depot an associated sorting center. However, both approaches presented here require well-chosen catchment areas which are necessary to aggregate demand or design relevant clusters. From an operational point of view, respecting the catchment area has lots of benefits for the implementation of optimized solutions in practice: It facilitates the organization by regions and allows to take into account the sorting capacity of the sorting centers. Catchment areas can be created with clustering algorithms or another mean. For instance, for the Vehicle Routing Problem, most of the authors use the k -means algorithm or other distance-based clustering approach to create kind of catchment areas. Some authors implement dedicated heuristics. If the objective is to take into account the sorting capacity of the sorting centers and the distance between sites to associate, this option might be the best.

In this manuscript, we only consider paths with sorting, in other works, we considered more logistics operations [GCH⁺19], but the problem was too large. To go even further,

we could introduce additional logistics operations such as transshipment (when parcels are added in a container without sorting in sorting centers) or swap (when containers change trucks). This last swap operation includes the four logistics operations with swap bodies (park, pickup, swap and exchange) defined by Huber and Geiger [HG14]. Indeed, we consider there is a swap operation as soon as at least one container of the truck is exchanged or removed. Adding new logistics operations would increase even more the size of the complete problem but would increase greatly the opportunities for consolidation and lead to cost savings as soon as we can find a relevant way to handle the trade-off between optimality and complexity, especially for the intercluster subproblem of the k -Clusters Algorithm. For the Hierarchical Algorithm with Aggregate Demands, new logistics operations could be introduced on the inner-level, but if an operation impacts both levels, there will be a need to find a heuristic way to optimize it, as it was the case for the sorting operation which was optimized thanks to the threshold and the aggregation. However, features like sorting capacity or link capacity for the road and not only for the vehicles (to avoid congestion for instance) could be added easily to both algorithms. The algorithms proposed can also be applied on one level networks or on networks in which the origin sites are the destination sites of the long-haul stage. Indeed, this last aspect impacts mainly the balancing of vehicles and not the approach for solving the parcel transportation problem.

In the k -Clusters Algorithm, the various clusterings tested provide comparable results and there is no obvious winner in the use cases studied. As future work, it could be interesting to design data instances with a sparse cut or an asymmetry in the demands to check whether or not one clustering method is better than the others under some specific data configurations. As for the Hierarchical Algorithm with Aggregate Demands, we show that the advantages of consolidation in inner-hubs depends strongly on the sorting costs and kilometric costs. Thus the optimal threshold found empirically depends on the datasets and would need to be recomputed if the algorithm were to be applied on another dataset. Moreover, for both algorithms, while the simulations were on very large datasets, without access to even larger datasets, it is hard to predict how scaling the size of the input data would impact the performances of the algorithms. One can study the impact of the problem size (smaller or larger datasets) on the algorithms proposed. Finally, to conclude about the datasets, one could also test the algorithms on datasets with a ratio sorting centers / total number of sites which vary to analyze how the algorithms behave.

In the Hierarchical Algorithm with Aggregate Demands, one perspective for future work could be to select the demands which should not be sorted on criteria others than their size or filling rate of vehicles. In this work we wanted to build a solution by deciding that large demands will not be sorted and smaller ones will be. But at the end of our algorithm we decided to question this idea in a very limited way. Indeed, we allow only the large demands which have a twin residual demand to use a direct path or a path with sorting, and in this case they can use only the path used by their twin demand. The other large demands have to use one of the direct path options. One perspective for future work could be not to limit the options for the large demands.

In the industrial problem we consider, the paths of the parcels cannot have a length

superior to three operational links. This allows to enumerate and classify the possible parcels paths. In order to take advantage of this strong constraint, we implement a path-based model. One could also propose an arc-based model for parcel transportation problems with a limit in term of path length. One could consider testing the arc-based model of Meisen [Mei15] after adapting it to the specific constraints of our industrial problem.

Glossary

Aggregate demand: Virtual demand designed to have demands from sorting centers to sorting centers. It is the sum of all the demands for the delivery depots whose default sorting center is the sorting center of destination.

Arc-based model: Model in which the flow is modeled with arc variables.

Associated sorting center: Sorting center assigned to a delivery depots which is usually the closest (with respect to physical distance).

Built solution: Final global solution built with the Hierarchical Algorithm with Aggregation of Demands, obtained at the end of step f).

Catchment area of a site: It is the zone around the sorting center in which all the delivery depots have this sorting center as their default sorting center.

Capacity constraint: (or Link capacity constraint) it states that the number of parcels on each link is smaller than the number that can be contained by the vehicles on this link (see Equation(2.1c) in Chapter 2).

Clustering: Clustering is a Machine Learning technique which allows grouping of unlabeled data points into meaningful sets: the clusters [JMF99].

Collection: Process in which the parcels are picked-up in post-offices and gathered in their sorting center to be sorted before being routed in the network [Seb12].

Computed solution: At the end of the merging stage (of both divide-and-conquer algorithms), we obtain what we call the computed solution to the problem.

Consolidation: When demands for a group of destinations (intermediate or final) are put in the same container (presented in details in Chapter 2 Section 1.5).

Current operational strategy: Strategy applied nowadays on the ground by the postal company (presented in details in the Introduction and Chapter 3).

D+1 delivery: When the parcel is delivered within 24 hours.

D+2 delivery: When the parcel is delivered within 48 hours.

Delivery: The transportation stage from the post office to the recipient's house [Seb12].

Delivery constraint: This constraint states that all the parcels of each demand have to be delivered (see Equation(2.1b) in Chapter 2).

Delivery depot: Destination site, from which the parcels are dispatched to their final destination.

Demand: Origin-destination-volume triple, in which the volume is the number of parcels sent from the origin to the destination.

Demand graph: Graph representing the demands between each pair of sites. It is a directed bipartite graph with all arcs directed from sorting centers to delivery depots.

Design-balance constraint: It states the vehicles have to go back at their starting points before the end of the day in order to be used again the next day (see Equation(2.1d) in Chapter 2).

Direct link: The only link which composes a direct path (from a sorting center to a delivery depot outside of the catchment area of the sorting center).

Direct path: Path composed of only one link from the origin sorting center to the destination delivery depot.

Distribution: Process in which the parcels are picked-up in their delivery depots and brought to the post-offices (or intermediate depots) associated to their final destination [Seb12].

Distribution path: The parcel are sent directly from the sorting center to a delivery depot which is in the catchment area of the origin.

Divide-and-conquer algorithm: A divide-and-conquer algorithm recursively divides a problem into two or more subproblems of the same or related type, until these become simple enough to be solved directly. The solutions to the subproblems are then combined (or merged) to give a solution to the original problem [CLRS09] (see Chapter 1 Section 4.3).

Double delivery: A truck with two containers delivers two sites in a row. The truck goes from the sorting center with parcels for depot 1 in the first container and parcel for depot 2 in the second container. It goes to depot 1 and distributes the parcels in the first container and then goes to depot 2 to distribute the parcels in the second container. It does not go through a sorting center between the two depots. The two delivery depots must be associated to the same sorting center.

Echelon: Each pair of layers constitutes one level of the hierarchical network and is sometimes referred to as an echelon [CGS15].

Empty repositioning: Movements of vehicles to ensure the balance over the course of a day. After delivering parcels in the delivery depots, vehicles are sent back empty to sorting centers in a way which ensures that at the end of the day there are as many vehicles in each sorting centers as it was the case at the beginning [JST⁺04] .

Gap: Gap between the best objective value found and the lower bound. When it is zero the value found is optimal (see Chapter 1 Section 4.1).

Hierarchical Algorithm with Aggregation of Demands: (also called Hierarchical Algorithm) Algorithm presented in Chapter 5.

Hierarchical hub-and-spoke network: It is a hub-and-spoke network in which each site is assigned to exactly one hub.

HLP: Hub Location Problem [GN11] (see Chapter 1 Section 3.3).

HSND: Hub-andSpoke Network Design [O’K98] (see Chapter 1 Section 3.2).

Hub-and-spoke network: Hub-and-spoke network are networks in which all links must either begin or end at a hub, the other extremities of the links being the spokes [BO99].

Hub sorting: Sorting which takes place in the inner-hubs. It is the first sorting over two when parcels from a demand are on a path with two sortings (the sorting post-collection does not count).

Hybrid hub-and-spoke network: A hub-and-spoke network in which there are possibilities to bypass the hubs with direct links.

ILP: Integer Linear Program (see Chapter 1 Section 4.1).

Inner-hub: Sorting center which is on a parcel path between two other sorting centers.

Intercluster problem: Problem of routing the demands whose origin and destinations are in different clusters.

Intraccluster problem: Problem of routing the demands whose origin and destinations are in the same cluster.

Large demand: Demand above the threshold in the Hierarchical Algorithm, which will have the possibility to use a direct path.

Layer: The sites of a same type (inner-hubs, sorting centers, delivery depots) form a layer or stage [CGS15].

Level: Each pair of layers constitutes one level of the hierarchical network and is sometimes referred to as an echelon [CGS15].

LHPTP: Long-Haul Parcel Transportation Problem.

Long-haul transportation: The transportation between the collection sorting center to the delivery depots [Cra03, Seb12].

LP: Linear Program (see Chapter 1 Section 4.1).

Lower bound: The cost of the best (minimum cost) transportation plan cannot be under this value, which increases with the solving of the MILP.

MILP: Mixed Integer Linear Program (see Chapter 1 Section 4.1).

MCND: Multicommodity Capacitated Network Design [GCF99] (see Chapter 1 Section 2.3).

Operational level: The operational level concerns the short-term optimization (schedules for services, crews, maintenance activities etc.) [Cra00, MW84].

Operational link: Arc between two sites associated with a vehicle type and a time slot.

Operational path: The combination of links between sites associated with a vehicle and operations performed on a parcel flow between its origin and its destination.

Optimization strategy: The set of operational paths offered to the MILP solver. It changes according to the soft business constraints respected.

Parcel: Physical object which must be delivered from a specified origin sorting center to a designated delivery depot. It has an average size.

Path-based model: Model in which the flow is modeled with path variables.

Physical distance graph: Each arc from point a to point b represents the real-life distance for a vehicle to travel from a to b .

Point-to-point delivery problems: Problems with point-to-point demands for delivery [LMSL92] (see Chapter 1 Section 2.1).

Point-to-point network: Network in which all pairs of sites are connected.

Post-collection sorting: First sorting which allows to sort the freshly collected packages. It is mandatory and takes place in the starting sorting center. As this sorting is mandatory, we will not talk about it in the optimization process.

Quality of service: It is one of the classical objectives in parcel transportation. It consists of delivering most of the parcels (a D+1-ratio is given) within 24 hours (D+1 delivery or next day delivery) [Mei15].

Refined solution: After the merging stage (of both divide-and-conquer algorithms), the computed solution is injected as a first solution in a one hour MILP run which results in the refined solution.

Residual Demand: Demand which is not large enough to be routed directly in the Hierarchical Algorithm with Aggregation of Demands.

SND: Service Network Design [Cra00] (see Chapter 1 Section 2.2).

SNDAM: Service Network Design with Asset Management [ACC09a] (see Chapter 1 Section 2.3).

Semi-random clustering: In this clustering, the clusters of the sorting centers are decided randomly. For this “clustering algorithm” a number between 1 and k is uniformly assigned to each sorting center. This number represents the cluster in which the site is. Then the delivery depots are put in the same cluster as their associated sorting center.

Soft constraint: Constraint which should be respected, otherwise there might be a penalty.

Sol fixed dir.: Solution with fixed direct paths in the Hierarchical Algorithm with Aggregation of Demands obtained at the end of step e).

Solution space: The feasible solution space of a linear program is a polyhedron and the set of optimal solutions are on these faces (see Chapter 1 Section 4.1).

Sorting: logistics operation which separates parcels according to their destination (intermediate or final).

Sorting capacity: Number of vehicles or parcels that a sorting center can receive.

Sorting center: (or s. center) origin site and intermediate site. The sorting operation takes place in these sites.

Strategic level: The strategic level of optimization deals with long-term optimization (acquisitions of vehicles, construction of roads or buildings etc.) [Cra00, MW84].

Strong constraint: Constraint which has to be respected, otherwise the solution is not feasible.

Tactical level: The tactical level of optimization treats medium-term optimization (effective use of the network) [Cra00, MW84].

Transportation plan: Output of the LHPTP. It contains the paths for parcels, the number of vehicles of each type on each operational link and the costs of the plan.

Vehicles: They are of two types: truck with one container, truck with two containers.

References

- [ACC09a] Jardar Andersen, Teodor G. Crainic, and Marielle Christiansen. Service network design with asset management: Formulations and comparative analyses. *Transportation Research Part C: Emerging Technologies*, 17(2):197–207, 2009. 22, 50, 133
- [ACC09b] Jardar Andersen, Teodor G. Crainic, and Marielle Christiansen. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377–389, 2009. 22
- [AK08] Sibel Alumur and Bahar Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1 – 21, 2008. 29
- [AMO88] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows*. Prentice-Hall, Inc., 1988. 19
- [Ayk95] Turgut Aykin. Networking policies for hub-and-spoke systems with application to the air transportation system. *Transportation science*, 29(3):201–221, 1995. 25
- [BA11] Omar Ben-Ayed. Re-engineering the inter-facility process of a parcel distribution company to improve the level of performance. *International Journal of Logistics Research and Applications*, 14(2):97–110, 2011. 13, 17, 18
- [Bat16] Ricardo F. Batista. *The South African Post Office supply chain design and route optimisation*. PhD thesis, University of Pretoria, 2016. 13, 15, 16, 17, 18
- [BDBP20] Mauro Boffa, Fernão De Borba, and Lukasz Piotrowski. Postal development report 2020. Technical report, Universal Postal Union, october 2020. 12
- [Ben62] Jacques F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik*, 4(1):238–252, 1962. 32

- [BEV14] Maria Battarra, Güneş Erdoğan, and Daniele Vigo. Exact algorithms for the clustered vehicle routing problem. *Operations Research*, 62(1):58–71, 2014. 82
- [BG15] Martin N. Baumung and Halil I. Gündüz. Consolidation of residual volumes in a parcel service provider’s long-haul transportation network. *Computational Logistics*, pages 437–450, 2015. 13, 15, 16, 17, 18, 106, 111
- [BGMS15] Martin N. Baumung, Halil I. Gündüz, Thomas Müller, and Hans-Jürgen Sebastian. Strategic planning of optimal networks for parcel and letter mail. In *Quantitative Approaches in Logistics and Supply Chain Management*, pages 81–103. Springer, 2015. 13, 15, 16, 17, 18
- [BJN⁺98] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W.P. Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998. 32
- [BKKW02] Cynthia Barnhart, Niranjan Krishnan, Daeki Kim, and Keith Ware. Network design for express shipment delivery. *Comput. Optim. Appl.*, 21(3):239–262, March 2002. 13, 14, 15, 16, 17, 18, 20, 22, 31, 32
- [BKS00] Arno Bruns, Andreas Klose, and Paul Stähly. Restructuring of swiss parcel delivery services. *OR-Spektrum*, 22(2):285–302, May 2000. 13, 15, 16, 18
- [BO99] Deborah L. Bryan and Morton E. O’kelly. Hub-and-spoke networks in air transportation: an analytical review. *Journal of regional science*, 39(2):275–295, 1999. 25, 131
- [Cam94] James F. Campbell. Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405, 1994. 28
- [CC15] Mervat Chouman and Teodor G. Crainic. Cutting-plane matheuristic for service network design with design-balanced requirements. *Transportation Science*, 49:99–113, 2015. 22
- [CDB⁺15] Uwe Clausen, Daniel Diekmann, Jens Baudach, Jan Kaffka, and Moritz Pöting. Improving parcel transshipment operations-impact of different objective functions in a combined simulation and optimization approach. In *2015 Winter Simulation Conference (WSC)*, pages 1924–1935. IEEE, 2015. 5

- [CGS15] Rosario Cuda, Gianfranco Guastaroba, and Maria Grazia Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015. 107, 130, 131
- [Che85] Larry Chestler. Overnight air express: Spatial pattern, competition and the future in small package delivery service. *Transportation Quarterly*, 39(1), 1985. 25
- [CHTV16] Teodor G. Crainic, Mike Hewitt, Michel Toulouse, and Duc M. Vu. Service network design with resource constraints. *Transportation Science*, 50(4):1380–1393, 2016. 22
- [Chv83] Vasek Chvatal. *Linear programming*. Macmillan, 1983. 30
- [CHVG10] Rutger Claes, Tom Holvoet, and Jelle Van Gompel. Coordination in hierarchical pickup and delivery problems using delegate multi-agent systems. In *Proceedings of the 4th Workshop on Artificial Transportation Systems and Simulation*, pages 1–7, 2010. 27
- [CK98] Michael A. Crew and Paul R. Kleindorfer. Efficient entry, monopoly, and the universal service obligation in postal service. *Journal of Regulatory Economics*, 14(2):103–126, 1998. 12
- [CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009. 18, 32, 130
- [CMPT08] Teodor G. Crainic, Simona Mancini, Guido Perboli, and Roberto Tadei. *Clustering-based heuristics for the two-echelon vehicle routing problem*, volume 46. CIRRELT Montréal, 2008. 82
- [Cos05] Alysson M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32(6):1429 – 1450, 2005. 20
- [Cpl09] IBM ILOG Cplex. V12. 1: User’s manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009. 31
- [CR86] Teodor G. Crainic and Jean-Marc Rousseau. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3):225–242, 1986. 19, 20
- [Cra98] Teodor G. Crainic. *A survey of optimization models for long-haul freight transportation*. Centre for Research on Transportation, 1998. 19, 20
- [Cra00] Teodor G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272 – 288, 2000. 3, 19, 20, 132, 133

- [Cra03] Teodor G. Crainic. Long-haul freight transportation. In *Handbook of transportation science*, pages 451–516. Springer, 2003. 3, 19, 36, 41, 132
- [CRC86] John R Current, Charles S ReVelle, and Jared L Cohon. The hierarchical network design problem. *European Journal of Operational Research*, 27(1):57–66, 1986. 107
- [CRWM07] Amy Cohn, Sarah Root, Alex Wang, and Douglas Mohr. Integration of the load-matching and routing problem with equipment balancing for small package carriers. *Transportation Science*, 41(2):238–252, 2007. 13, 15, 16, 18, 24, 71
- [Cur88] John R. Current. The design of a hierarchical transportation network with transshipment facilities. *Transportation Science*, 22(4):270–277, 1988. 107
- [CX06] Ren Chunyu and Wang Xiaobo. Research on VRP optimizing based on hierarchy clustering and IGA under common distribution. In *International Conference on Computational Intelligence and Security*, volume 1, pages 143–146. IEEE, 2006. 82
- [Dan83] George B. Dantzig. Reminiscences about the origins of linear programming. In *Mathematical Programming The State of the Art*, pages 78–86. Springer, 1983. 30
- [Dan90] George B. Dantzig. Origins of the simplex method. In *A history of scientific computing*, pages 141–151. Association for Computing Machinery, 1990. 31
- [Das02] Sanjoy Dasgupta. Performance guarantees for hierarchical clustering. In *International Conference on Computational Learning Theory*, pages 351–363. Springer, 2002. 80
- [DC87] Pierre J. Dejax and Teodor G. Crainic. Survey paper—a review of empty flows and fleet management models in freight transportation. *Transportation science*, 21(4):227–248, 1987. 22
- [DC07] Rodolfo Dondo and Jaime Cerdá. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European journal of operational research*, 176(3):1478–1507, 2007. 82
- [DLPR18] Émilie Dufour, Gilbert Laporte, Julie Paquette, and Marie-Ève Rancourt. Logistics service network design for humanitarian response in east africa. *Omega*, 74:1–14, 2018. 21
- [DO13] Benjamin S. Duran and Patrick L. Odell. *Cluster analysis: a survey*, volume 100. Springer Science & Business Media, 2013. 79

- [Dum07] Ekrem Duman. Decision making by simulation in a parcel transportation company. *Journal of the Franklin Institute*, 344(5):672–683, 2007. 5
- [DW60] George B. Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111, 1960. 32
- [EHJ⁺09] Andreas T. Ernst, Horst Hamacher, Houyuan Jiang, Mohan Krishnamoorthy, and Gerhard Woeginger. Uncapacitated single and multiple allocation p-hub center problems. *Computers & Operations Research*, 36(7):2230–2241, 2009. 29
- [ES87] Jonathan Eckstein and Yosef Sheffi. Optimization of group line-haul operations for motor carriers using twin trailers. *Transportation Research Record*, pages 12–23, 1987. 24
- [FHAN13] Reza Zanjirani Farahani, Masoud Hekmatfar, Alireza Boloori Arabani, and Ehsan Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096 – 1109, 2013. 29
- [FPL93] Judith M. Farvolden, Warren B. Powell, and Irvin J. Lustig. A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem. *Operations Research*, 41(4):669–693, 1993. 31
- [GCF99] Bernard Gendron, Teodor G. Crainic, and Antonio Frangioni. Multi-commodity capacitated network design. In *Telecommunications network planning*, pages 1–19. Springer, 1999. 22, 132
- [GCH⁺19] Camille Gras, Van-Dat Cung, Nathalie Herr, Alantha Newman, and Nicolas Teypez. Freight network design with logistics operation management. In *ROADEF2019*, 2019. 54, 125
- [GCH⁺20] Camille Gras, Van-Dat Cung, Nathalie Herr, Alantha Newman, and Nicolas Teypez. Point-to-point parcel delivery via clustering. In *ROADEF2020*, 2020. 78
- [GCH⁺21] Camille Gras, Van-Dat Cung, Nathalie Herr, Alantha Newman, and Nicolas Teypez. Long-haul parcel transportation on a hierarchical network. In *ROADEF2021*, 2021. 106
- [GCHN21] Camille Gras, Van-Dat Cung, Nathalie Herr, and Alantha Newman. A hierarchical network approach for long-haul parcel transportation. In *IFIP International Conference on Advances in Production Management Systems*, pages 168–178. Springer, 2021. 106
- [GF11] Jesus Gonzalez-Feliu. Two-echelon freight transport optimisation: unifying concepts via a systematic review. *Working Papers on Operations Management*, 2(1):18–30, 2011. 107

- [GH08] 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. 19
- [GN11] Shahin Gelareh and Stefan Nickel. Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1092 – 1111, 2011. 28, 29, 131
- [GS00] Tore Grünert and Hans-Jürgen Sebastian. Planning models for long-haul operations of postal and express shipment companies. *European Journal of Operational Research*, 122(2):289 – 309, 2000. 13, 14, 15, 16, 17, 18, 22, 31
- [Hak65] Louis S. Hakimi. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations research*, 13(3):462–475, 1965. 28
- [HG14] Sandra Huber and Martin Josef Geiger. Swap body vehicle routing problem: A heuristic solution approach. In Rosa G. González-Ramírez, Frederik Schulte, Stefan Voß, and Jose A. Ceroni Díaz, editors, *Computational Logistics*, pages 16–30, Cham, 2014. Springer International Publishing. 23, 126
- [HGLLP⁺18] Israel D. Herrera-Granda, Leandro L. Lorente-Leyva, Diego H. Peluffo-Ordóñez, Robert M. Valencia-Chapi, Yakcleem Montero-Santos, Jorge L. Chicaiza-Vaca, and Andrés E. Castro-Ospina. Optimization of the university transportation by contraction hierarchies method and clustering algorithms. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 95–107. Springer, 2018. 82
- [Hil15] Julia Hillebrandt. New approaches of realizing an optimized network. In *Quantitative Approaches in Logistics and Supply Chain Management*, pages 149–169. Springer, 2015. 29
- [HJ97] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1):191–215, 1997. 79
- [HWH15] Yong He, Jiechang Wen, and Meihua Huang. Study on emergency relief vrp based on clustering and pso. In *2015 11th International Conference on Computational Intelligence and Security (CIS)*, pages 43–47. IEEE, 2015. 82
- [JBW⁺21] Xiaoping Jiang, Ruibin Bai, Stein W. Wallace, Graham Kendall, and Dario Landa-Silva. Soft clustering-based scenario bundling for a progressive hedging heuristic in stochastic service network design. *Computers & Operations Research*, 128:105182, 2021. 83

- [JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988. 79
- [JH12] De-gang Ji and Dong-mei Huang. A research based on k-means clustering and artificial fish-swarm algorithm for the vehicle routing optimization. In *2012 8th International Conference on Natural Computation*, pages 1141–1145. IEEE, 2012. 82
- [JL99] Josef Jablonský and Josef Lauber. A time-cost optimization of the national postal distribution network. *Journal of Multi-Criteria Decision Analysis*, 8(1):51–56, 1999. 18, 29
- [JLK78] David S. Johnson, Jan Karel Lenstra, and Rinnooy A.H.G. Kan. The complexity of the network design problem. *Networks*, 8(4):279–285, 1978. 19
- [JMF99] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. 79, 129
- [Joh67] Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. 78
- [JST⁺04] Benjamin Jansen, Pieter C.J. Swinkels, Geert J.A. Teeuwen, Babette van Antwerpen de Fluiter, and Hein A. Fleuren. Operational planning of a large-scale multi-modal transportation system. *European Journal of Operational Research*, 156(1):41–53, 2004. 13, 15, 16, 18, 22, 31, 131
- [KBWR99] Daeki Kim, Cynthia Barnhart, Keith Ware, and Gregory Reinhardt. Multimodal express package delivery: A service network design application. *Transportation Science*, 33(4):391–407, 1999. 13, 14, 15, 16, 17, 18, 20
- [Kel60] James E Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the society for Industrial and Applied Mathematics*, 8(4):703–712, 1960. 32
- [Kel02] Jeffrey Dean Kelly. Chronological decomposition heuristic for scheduling: Divide and conquer method. *AIChE Journal*, 48(12):2995–2999, 2002. 32
- [KG93] Michael J. Kuby and Robert Gordon Gray. The hub network design problem with stopovers and feeders: The case of federal express. *Transportation Research Part A: Policy and Practice*, 27(1):1 – 12, 1993. 26
- [KLP14] Sang-Jin Kim, Hyunwoo Lim, and Minyoung Park. Analysing the cost efficiency of parcel distribution networks with changes in demand. *International Journal of Urban Sciences*, 18(3):416–429, 2014. 18, 20, 26
- [Knu99] Donald E. Knuth. The art of computer programming. *Sorting and searching*, 3:426–458, 1999. 5

- [Kri98] Niranjana Krishnan. *Design of large scale transportation service networks with consolidation: Models, algorithms and applications*. PhD thesis, Massachusetts Institute of Technology, 1998. 20
- [Lau18] Pierre Launay. *The technical and organisational redeployment of French groupage networks*. Theses, Université Paris-Est, March 2018. 2
- [LC04] Cheng-Chang Lin and Sheu-Hua Chen. The hierarchical network design problem for time-definite express common carriers. *Transportation research part B: methodological*, 38(3):271–283, 2004. 26, 107, 123
- [LC08] Cheng-Chang Lin and Sheu-Hua Chen. An integral constrained generalized hub-and-spoke network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 44(6):986–1003, 2008. 28
- [LDP⁺20] Xiangyong Li, Yi Ding, Kai Pan, Dapei Jiang, and YP Aneja. Single-path service network design problem with resource constraints. *Transportation Research Part E: Logistics and Transportation Review*, 140:101945, 2020. 22
- [LHX20] Zekang Lan, Shiwei He, and Yan Xu. Combining benders decomposition and column generation for scheduled service network design problem in rail freight transportation. *Transportmetrica A: Transport Science*, pages 1–30, 2020. 21
- [LJL⁺21] Jian Li, Liyan Jia, Tingting Li, Xiucui Guan, Haoxiang Wang, Dong Lu, and Hao Sun. Scheduled service network design of long-haul freight transportation based on the segment subcontract. *Computers & Industrial Engineering*, page 107253, 2021. 21
- [LLL03] Cheng-Chang Lin, Yu-Jen Lin, and Dung-Ying Lin. The economic effects of center-to-center directs on hub-and-spoke networks for air express common carriers. *Journal of Air Transport Management*, 9(4):255–265, 2003. 26
- [Llo82] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 78, 81
- [LLWZ18] Siqi Liu, Boliang Lin, Jianping Wu, and Yinan Zhao. Modeling the service network design problem in railway express shipment delivery. *Symmetry*, 10(9), 2018. 21
- [LM14] Jeong-Hun Lee and Ilkyeong Moon. A hybrid hub-and-spoke postal logistics network with realistic restrictions: A case study of Korea post. *Expert systems with applications*, 41(11):5509–5519, 2014. 13, 15, 16, 18, 25, 29, 106

- [MW84] Thomas L. Magnanti and Richard T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984. 3, 19, 132, 133
- [MW11] Qiang Meng and Xinchang Wang. Intermodal hub-and-spoke network design: incorporating multiple stakeholders and multi-type containers. *Transportation Research Part B: Methodological*, 45(4):724–742, 2011. 28
- [Nga07] Somnuk Ngamchai. *Air express network design with hub sorting*. PhD thesis, University of Maryland, College Park, 2007. 13, 14, 16, 17, 18
- [NJW02] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002. 78, 80
- [O’k87] Morton E. O’kelly. A quadratic integer program for the location of interacting hub facilities. *European journal of operational research*, 32(3):393–404, 1987. 28
- [O’K98] Morton E. O’Kelly. A geographer’s analysis of hub-and-spoke networks. *Journal of transport Geography*, 6(3):171–186, 1998. 25, 28, 109, 131
- [OM94] Morton E. O’Kelly and Harvey J. Miller. The hub network design problem: A review and synthesis. *Journal of Transport Geography*, 2(1):31 – 40, 1994. 27
- [Pak14] Malay K. Pakhira. A linear time-complexity k-means algorithm using cluster shifting. In *2014 international conference on computational intelligence and communication networks*, pages 1047–1051. IEEE, 2014. 81
- [PCDCLS16] Johana Andrea Patiño Chirva, Yesica Xiomara Daza Cruz, and Eduyn Ramiro Lopez-Santana. A hybrid mixed-integer optimization and clustering approach to selective collection services problem of domestic solidwaste. *Ingeniería*, 21(2):235–257, 2016. 33, 82
- [PCM09] Michael B. Pedersen, Teodor G. Crainic, and Oli BG Madsen. Models and tabu search metaheuristics for service network design with asset-balance requirements. *Transportation Science*, 43(2):158–177, 2009. 23
- [PLT18] José Miguel Quesada Pérez, Jean-Charles Lange, and Jean-Sébastien Tancrez. A multi-hub express shipment service network design model with flexible hub assignment. *Transportation Research Part E: Logistics and Transportation Review*, 120:116–131, 2018. 21
- [PR91] Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991. 32

- [RNDL15] Bart Van Riessen, Rudy R. Negenborn, Rommert Dekker, and Gabriel Lodewijks. Service network design for an intermodal container network with flexible transit times and the possibility of using subcontracted transport. *International Journal of Shipping and Transport Logistics*, 7(4):457–478, 2015. 21
- [Ruc18] Leah J. Ruckle. *Approaches to solving the express shipment service network design problem*. PhD thesis, Georgia Institute of Technology, 2018. 20
- [SA16] Elif Z. Serper and Sibel A. Alumur. The design of capacitated intermodal hub networks with different vehicle types. *Transportation Research Part B: Methodological*, 86:51 – 65, 2016. 28
- [Sch98] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998. 32
- [SDN⁺14] Maryam SteadieSeifi, Nico P. Dellaert, Wim Nuijten, Tom Van Woensel, and Rasa Raoufi. Multimodal freight transportation planning: A literature review. *European Journal of Operational Research*, 233(1):1 – 15, 2014. 19
- [Seb12] Hans-Jürgen Sebastian. Optimization approaches in the strategic and tactical planning of networks for letter, parcel and freight mail. In *Modeling for Decision Support in Network-Based Services*, pages 36–61. Springer, 2012. 12, 23, 129, 130, 132
- [SS03] Chang Sup Sung and Sang Hwa Song. Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, 54(12):1283–1295, 2003. 21, 31
- [SS⁺08] Marc Sevaux, Kenneth Sörensen, et al. Hamiltonian paths in large clustered routing problems. In *Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME*, volume 8, pages 411–417, 2008. 82
- [Sta21] Statista. E-commerce worldwide, July 2021. 1
- [SZH15] Li Sun, Lindu Zhao, and Jing Hou. Optimization of postal express line network under mixed driving pattern of trucks. *Transportation Research Part E: Logistics and Transportation Review*, 77:147–169, 2015. 23
- [Taa07] Menno Taanman. Service network design for express shipment delivery - an overview. Master’s thesis, Tilburg University, 2007. 21
- [TH85] Rex S. Toh and Richard G. Higgins. The impact of hub and spoke network centralization and route monopoly on domestic airline profitability. *Transportation journal*, pages 16–27, 1985. 25

- [Tom66] J. A. Tomlin. Minimum-cost multicommodity network flows. *Operations Research*, 14(1):45–51, 1966. 19
- [TS01] Sam R. Thangiah and Said Salhi. Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem. *Applied Artificial Intelligence*, 15(4):361–383, 2001. 82
- [TSC10] Nicolas Teypez, Susann Schrenk, and Van-Dat Cung. A decomposition scheme for large-scale service network design with asset management. *Transportation Research Part E: Logistics and Transportation Review*, 46(1):156 – 170, 2010. 22
- [vHMS19] Wouter van Heeswijk, Martijn Mes, and Marco Schutten. Transportation management. In *Operations, logistics and supply chain management*, pages 469–491. Springer, 2019. 3
- [WAL⁺18] Yong Wang, Kevin Assogba, Yong Liu, Xiaolei Ma, Maozeng Xu, and Yinhai Wang. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Systems with Applications*, 104:244–260, 2018. 83
- [Wan18] I-Lin Wang. Multicommodity network flows: A survey, part i: Applications and formulations. *International Journal of Operations Research*, 15(4):145–153, 2018. 31
- [Wie08] Nicole Wieberneit. Service network design for freight transportation: a review. *OR Spectrum*, 30(1):77–112, Jan 2008. 20
- [WN99] Laurence A. Wolsey and George L. Nemhauser. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, 1999. 32
- [WZ04] Michael Wasner and Günther Zäpfel. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *International Journal of Production Economics*, 90(3):403 – 419, 2004. Production Control and Scheduling. 28
- [XW05] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005. 79
- [Zhu11] Endong Zhu. *Scheduled service network design for integrated planning of rail freight transportation*. PhD thesis, Université de Montréal, 2011. 21
- [ZPM21] Wenxin Zhang, Alexia P Payan, and Dimitri N Mavris. Improving courier service network efficiency through consolidations. In *AIAA Scitech 2021 Forum*, page 1233, 2021. 13, 14, 15, 18

- [ZW02] Günther Zäpfel and Michael Wasner. Planning and optimization of hub-and-spoke transportation networks of cooperative third-party logistics providers. *International Journal of Production Economics*, 78(2):207–220, 2002. 13, 15, 16, 18, 24, 25, 26, 109
- [ZWL07] Jian Zhang, Yaohua Wu, and Pei Liu. Routing problem for hybrid hub-and-spoke transportation network: a case study of a ltl carrier. In *2007 IEEE International Conference on Automation and Logistics*, pages 170–175. IEEE, 2007. 25, 26

Résumé long

Dans cette thèse, nous concevons un réseau de transport de colis long-courrier pour une entreprise postale. Le transport long-courrier (de type inter-urbain) se fait des centres de tri aux dépôts de livraison. Ni la manière dont les colis sont acheminés de leur bureau de poste de départ à leur centre de tri de collecte, ni comment ils sont transportés vers les bureaux de poste puis aux particuliers ne sont pris en considération.

Le problème du transport long-courrier de colis (PTLCC), que nous définissons formellement, est un problème tactique d'optimisation qui consiste à définir un plan de transport annuel composé de liaisons fixes, basé sur des prévisions de volumes à moyen terme, dont on minimise le coût total. Ce coût est composé du coût logistique et du coût de transport. Dans notre cas d'étude, le transport de colis (à la fois pour des industriels et des particuliers) se fait avec deux types de véhicules (camions à un ou deux conteneurs) qui sont équilibrés chaque jour sur le réseau grâce à la gestion des conteneurs vides. Le transport est optimisé sur un réseau hybride hub-and-spoke biniveau à l'échelle d'un pays. En effet, ce problème industriel provient d'une entreprise postale et leurs ensembles de données sont de taille réaliste (environ 225 sites avec 2500 demandes). Les colis sont envoyés de tous les sites de départ vers tous les sites de destination. Ils sont transportés dans des camions dans lesquels les colis pour différentes destinations sont mélangés. Ainsi ils sont triés dans des centres de tri pour arriver à destination et ce tri a un coût. Mais cette opération de tri permet une meilleure mutualisation des colis dans les conteneurs. Notre objectif est de choisir combien de camions vont de chaque site à chaque site, avec combien de conteneurs et avec combien de colis et lesquels, tout en minimisant le coût total. De plus, une même demande (origine, destination, nombre de colis) peut être acheminée sur plusieurs chemins simultanément ce qui augmente la complexité du problème. Ainsi, le nombre de plans de transport possibles explose.

Dans le premier chapitre, le vocabulaire du transport de colis et les problèmes de transport proches du problème traité, le PTLCC, sont introduits. Les problèmes de routage et de conception de réseaux sont présentés et rattachés au problème étudié. Cela permet de positionner le PTLCC dans la littérature scientifique. Le PTLCC est un problème de livraison point à point qui consiste à minimiser le coût du transport long-courrier des colis sur un réseau hybride hub-and-spoke à deux niveaux avec trois types de sites. C'est aussi un problème de conception de réseau de services avec gestion d'actifs. Le réseau sur lequel nous concevons un plan de transport est précisément un réseau hub-and-spoke non-hiérarchique hybride à deux niveaux qui est hybride sur le niveau interne

uniquement. Cependant, c'est un réseau hiérarchisé à deux niveaux dans le sens où nous avons trois types de sites (hubs internes, centres de tri et dépôts de livraison).

Les principales caractéristiques du PTLCC sont aussi mises en avant en comparaison avec l'existant. Ainsi, on voit qu'une des spécificités du problème est que les origines et destinations des demandes sont fixées. L'opération de tri a un coût par colis. Cette opération de tri est l'une des caractéristiques du domaine du transport de colis. Comme le PTLCC a des coûts linéaires, nous développons un programme linéaire pour le modéliser. Les méthodes de résolution proposées dans la thèse sont basées sur la programmation linéaire en variables mixtes (PLM), les fondamentaux de la PLM sont rappelés.

Dans le deuxième chapitre, le problème du transport long-courrier de colis (PTLCC) est présenté formellement. Son cadre d'application ainsi que les notions et contraintes inhérentes au cas d'étude sont développés. Ces notions incluent les demandes, l'opération de tri, le repositionnement à vide des véhicules, la consolidation des demandes et les chemins opérationnels. Le problème d'optimisation est développé : ses entrées, son objectif, ses contraintes et sa sortie. Étant donné les sites (et leurs types), la distance et le temps de trajet entre eux, les demandes, la capacité des véhicules, les coûts de transport et de tri, l'objectif est de concevoir un plan de transport journalier qui permet de livrer toutes les demandes depuis leur centre de tri d'origine vers leurs dépôts de livraison de destination.

Le plan de transport doit respecter les contraintes du problème : la contrainte de livraison, la contrainte de capacité et la contrainte d'équilibrage. La contrainte de livraison stipule que tous les colis doivent être livrés de leur lieu d'origine (un centre de tri) à leur site de destination (dépôt de livraison). La contrainte de capacité force les capacités des véhicules à être respectées. Cependant, le nombre de véhicules que le plan de transport peut utiliser n'est pas limité, pas plus que le nombre de véhicules entre chaque paire de sites. La dernière contrainte est la contrainte d'équilibrage. Comme nous concevons un plan de transport quotidien qui doit être appliqué le lendemain, les véhicules doivent être repositionnés sur les sites d'expédition pour pouvoir être utilisés à nouveau le lendemain.

De plus, il existe deux contraintes métier souples : le tri hub doit avoir lieu dans les hubs internes proposés par les gestionnaires de transport, et le dernier tri doit être effectué à l'intérieur de la zone de chalandise de la destination. Enfin, nous donnons une définition formelle du PTLCC. Nous avons décrit le réseau hub-and-spoke à deux niveaux sur lequel se déroule le PTLCC. Nous présentons les graphes qui modélisent le problème et proposons un programme linéaire mixte (PLM) orienté chemin pour le PTLCC. Notez que ce PLM peut être adapté pour résoudre d'autres problèmes de transport de colis long-courrier (par exemple, avec différents types de véhicules, sans équilibrage des véhicules, ou avec des contraintes de capacité de tri, etc.).

Dans le chapitre 3, nous présentons les jeux de données utilisés pour les tests qui sont fournis par une entreprise de service postal. Six configurations du réseau qui s'étalent sur une période de plusieurs années sont considérées en France métropolitaine. Les configurations diffèrent sur le nombre total de colis envoyés, correspondant ainsi à une croissance du marché prévue. L'ensemble des données collectées pour les six configurations est

analysé, par exemple en termes de répartitions des sites sur le territoire, ou encore des demandes sur l'ensemble des sites. Nous discutons ensuite de la taille des formulations PLM sur ces entrées. Une borne supérieure du nombre de liens opérationnels et une borne supérieure du nombre de chemins opérationnels possibles du graphe associé au problème, qui constituent des entrées de notre modèle, sont déterminées afin de mesurer l'ampleur du problème. Puis nous explorons si nous devons ou non respecter les deux contraintes métier souples (respecter les zones de chalandise pour le dernier tri et les hubs internes choisis par les gestionnaires de transport pour le tri hub). Nous appliquons le PLM sur toutes les stratégies d'optimisation que nous proposons qui respectent ou ignorent ces deux contraintes. La conclusion de ces tests est que la meilleure option est d'utiliser la Stratégie 2 : respecter les zones de chalandise et ignorer les hubs internes. C'est la meilleure option en termes de taille de PLM : elle nous permet d'avoir suffisamment d'options pour obtenir une borne inférieure suffisamment petite pour avoir de bons résultats, alors qu'en même temps il n'y a pas trop de variables ce qui nous empêcherait de trouver une bonne solution. On peut donc dire que nous validons la contrainte métier du dernier tri en zone de chalandise. C'est une bonne nouvelle d'un point de vue opérationnel car les gens préfèrent changer le moins possible leurs habitudes. De plus, le respect des zone de chalandise a de nombreux avantages pour la mise en œuvre dans la pratique de solutions optimisées : il facilite l'organisation par régions et permet de traiter le dimensionnement et la définition des zones de chalandses dans un problème séparé, prenant notamment en compte des contraintes de capacité de tri sur les centres de tri.

Quant à l'autre contrainte souple (à propos des hubs internes), dans ce qui suit nous utiliserons soit les hubs internes fournis soit tous les centres de tri comme hubs internes potentiels, dans ce cas, nous appelons hubs internes les centres de tri utilisés pour le tri hub et les centres de tri " simples " les autres. Dans ce dernier cas, nous discuterons des hubs internes proposés par les gestionnaires de transport et essayerons de trouver de meilleures options pour les emplacements de tri du hub sans développer un modèle de localisation de hub. Notez que l'utilisation de tous les centres de tri comme hubs internes n'est pas une obstacle opérationnel car le tri hub a lieu dans les mêmes plages horaires que le tri " habituel ". Ainsi, il n'y a aucun coût supplémentaire pour transformer un centre de tri " simple " en un hub interne. Le seul problème est la capacité de tri, que nous ne considérons pas dans cette étude de cas.

Dans les chapitres suivants, nous considérerons un modèle dans lequel nous autorisons les chemins directs, le repositionnement des camions en dehors de la zone de chalandise des dépôts de livraison dans lesquels ils s'arrêtent et le tri hub. Nous testerons successivement ce qu'il se passe en ignorant la contrainte souple sur le tri hub (Stratégie 2) et le respect de cette contrainte (Stratégie 1), tout en respectant les zones de chalandise pour tous les tests.

Étant donné que nous devons en fait inclure plus de chemins que nous ne pouvons nous permettre dans les petites instances que le PLM peut résoudre, nous avons besoin d'un moyen de réduire la taille de notre problème afin d'obtenir une meilleure solution du solveur de PLM. Pour ce faire, nous proposons deux algorithmes diviser pour régner,

dans lesquelles nous partitionnons le problème en instances plus petites. Dans ces deux heuristiques, on profitera du regroupement par zones de chalandise, car il permet de regrouper les sites plus facilement ou d'agréger les demandes.

Dans le chapitre 4, un algorithme original, l'algorithme k -Clusters est présenté. Cet algorithme optimise le PTLCC après avoir regroupé les sites du réseau en clusters. Nous testons des techniques classiques de clustering (clustering spectral, clustering hiérarchique, k -means et aléatoire) en utilisant des fonctions de similarité appropriées (basées sur les demandes et sur les distances) pour étudier l'impact sur les résultats. Le problème d'origine est divisé en sous-problèmes intracluster et intercluster résolus avec le PLM. Les solutions des sous-problèmes sont ensuite fusionnées. Ceci constitue une heuristique sur mesure qui nous permet de résoudre le problème de transport de colis longue distance de manière plus précise et plus efficace qu'avec une utilisation directe du PLM sans clustering. Ce type d'approche a été utilisé avec succès dans les problèmes de transport de type planification de tournées. En effet ces problèmes autorisent des clusters générant des sous-problèmes résolus de manière indépendante. A contrario, pour les problèmes de conception de réseaux et de transport de colis ces méthodes ont très peu été utilisées, probablement à cause de la présence systématique de flux interclusters qu'il faut optimiser. L'algorithme k -Clusters nous permet d'avoir des résultats comparables à ceux obtenus avec une méthode exacte en termes de qualité. Ses avantages sont qu'il permet d'utiliser moins de mémoire et/ou éventuellement un solveur libre. Comme nous l'avions prévu au chapitre 3, la Stratégie 2 permet d'avoir de meilleurs résultats que la Stratégie 1 avec l'algorithme k -Clusters comme c'était déjà le cas avec seulement le PLM. Il n'y a pas de "meilleure méthode de clustering" pour notre étude de cas, car toutes les méthodes de clustering donnent des résultats comparables, ce qui prouve que l'essentiel est de diviser en clusters équilibrés et de respecter les zones de chalandises. Le clustering semi-aléatoire donne des résultats satisfaisants ainsi que les clusterings obtenus avec des algorithmes classiques appliqués sur deux caractéristiques clés des données : la distance physique et les demandes. Aussi, la création d'une méthode de clustering sur mesure pour notre étude de cas ne semble pas prometteur. Enfin, nous avons prouvé que moins il y a de clusters, meilleurs sont les résultats. Ainsi, il n'est pas nécessaire de tester l'algorithme avec plus de clusters car les meilleurs résultats sont obtenus avec seulement deux clusters ensuite l'augmentation du nombre de clusters dégrade la qualité de la solution.

La division en sous-problèmes permet d'utiliser le PLM qui fonctionne bien et semble prometteur. Une faiblesse de l'algorithme k -Clusters est que les sous-problèmes intracluster sont résolus avant le sous-problème intercluster et cela a un impact sur sa résolution via l'heuristique. Il pourrait être intéressant de voir ce qui se passe si nous résolvons entre les clusters d'abord, puis à l'intérieur des clusters. Ceci n'est pas possible dans l'algorithme k -Clusters (le problème intercluster est important et ne sera pas résolu efficacement avec un PLM), mais cela peut être fait dans un autre algorithme. C'est ce que nous faisons dans le chapitre suivant.

Dans le chapitre 5, pour aller plus loin dans le diviser-pour-régner et pour exploiter les zones de chalandises qui permettent d'avoir de bons résultats dans le cas du chapitre

précédent, nous concevons un deuxième algorithme : l'algorithme hiérarchique avec agrégation de demandes qui exploite la structure à deux niveaux du réseau. Ses performances sont liées à un seuil du taux de remplissage des camions. Les demandes au-dessus de ce seuil peuvent être acheminées directement. Celles en dessous de ce seuil doivent suivre la structure hiérarchique du réseau. L'acheminement des deux types de demandes est optimisé, d'abord séparément puis conjointement via plusieurs étapes dans lesquelles les sous-problèmes sont résolus avec le PLM. Le sous-problème interne peut être résolu avec un PLM car il est plus petit. L'extension de sa solution se fait aussi grâce à un PLM. Le point difficile de cette étape de résolution est le partage de la capacité des véhicules entre les demandes sur chaque lien opérationnel. Cela induit l'ajout de nouvelles contraintes de capacité. Enfin, l'ajout des grandes demandes à la solution construite hiérarchiquement est fait avec un PLM. On peut noter que contrairement à l'algorithme précédent, celui-ci résout le problème intercluster avant les problèmes intraclusters.

Différents seuils sont testés pour déterminer lequel donne les meilleures solutions et temps de calcul. Ces tests montrent qu'un meilleur taux de remplissage n'aboutit pas à un plan de transport moins cher dans notre cas. De plus, l'algorithme hiérarchique permet d'avoir des plans de transport nettement meilleurs que ceux appliqués sur le terrain, ceux obtenus via une utilisation directe du PLM et même ceux obtenus avec l'algorithme k -Clusters.

Cela montre que le paradigme diviser-pour-régner est efficace pour la conception de réseau de services lorsqu'il s'applique à un problème industriel de grande taille. En conclusion, dans cette thèse nous avons défini et modélisé un problème industriel: le PTLCC. Nous avons proposé un PLM qui modélise ce problème. Nous avons testé ce modèle avec des jeux de données fournis par une entreprise postale et cela nous a permis de voir que le PLM permet de trouver des solutions qui permettent d'économiser 10 à 20 % du coût total de la solution actuelle pour livrer les colis. Comme nous n'avions pas atteint la solution optimale, pour aller encore plus loin, nous avons proposé deux algorithmes diviser-pour-régner basés respectivement sur des méthodes de clustering standard et sur la hiérarchie naturelle du réseau. Ces méthodes nous ont permis de prouver l'importance de respecter la hiérarchie naturelle du réseau et nous ont permis de trouver des solutions encore meilleures qu'avec le PLM seul. Comme perspectives, on pourrait envisager de créer d'autres instances de tests avec des propriétés ciblées afin de tester les limites de chacun des deux algorithmes. On pourrait envisager de calculer différemment le seuil de l'algorithme hiérarchique et/ou d'utiliser des méthodes approchées pour résoudre les sous-problèmes, notamment le problème intercluster de l'algorithme k -clusters. Enfin, on pourrait ajouter d'autres opérations logistiques au modèle du PTLCC.

Abstract

In this PhD, we design a network for parcel long-haul transportation for a postal company. The long-haul transportation is from the collection sorting centers to delivery depots. We do not consider how the parcels are brought from their departure post office to their collection sorting center, neither how their are transported to the post offices and then to individuals. Parcels are sent from all the origin sites to all the destination sites. In our case study, parcels are transported in trucks with one or two containers. In these trucks, parcels for different destinations are mixed. Thus they are sorted in sorting centers in order to reach their destination and this sorting has a cost. Our objective is to choose how many trucks go from each site to another, with how many containers attached to them and with how many parcels in them while minimizing the total cost. To achieve this, we propose a linear model and two divide-and-conquer algorithms.

Keywords: Operational Research, Long-Haul Parcel Transportation, Network Design, Divide-and-Conquer, Tactical planning

Résumé

Dans cette thèse, nous concevons un réseau de transport de colis long-courrier pour une entreprise postale. Le transport long-courrier se fait des centres de tri aux dépôts de livraison. Ni l'acheminement des colis de leur bureau de poste de départ à leur centre de tri de collecte, ni l'acheminement vers les bureaux de poste puis vers les particuliers ne sont considérés. Les colis sont envoyés de tous les sites de départ vers tous les sites de destination. Dans notre étude de cas, les colis sont transportés dans des camions avec un ou deux conteneurs. Dans ces camions, les colis pour différentes destinations sont mélangés. Ainsi ils sont triés dans des centres de tri pour arriver à destination et ce tri a un coût. Notre objectif est de choisir combien de camions vont de chaque site à chaque site, avec combien de conteneurs et avec combien de colis et lesquels, tout en minimisant le coût total. Pour ce faire, nous proposons un modèle linéaire et deux algorithmes diviser-pour-régner.

Mots-clés : Recherche Opérationnelle, Transport long-courrier de colis, Conception de réseau, Diviser-pour-régner, Planification tactique