



HAL
open science

Contributions to workload equity for Vehicles Routing Problems

Maxime Agius

► **To cite this version:**

Maxime Agius. Contributions to workload equity for Vehicles Routing Problems. Operations Research [math.OA]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2023. English. ⟨NNT : 2023EMSEM028⟩. ⟨tel-04822058v2⟩

HAL Id: tel-04822058

<https://hal.science/tel-04822058v2>

Submitted on 29 Apr 2026

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.



HAL Authorization

THÈSE DE DOCTORAT

de Mines Saint-Etienne - Une école de l'IMT

Ecole Doctorale N°488
(Sciences, Ingénierie, Santé)

Spécialité de doctorat : Génie industriel

Soutenue publiquement le 13 septembre 2023 par :

Maxime Agius

**Contributions to workload equity for Vehicle Routing
Problems**

**Contributions à l'équité dans la charge de travail pour les
problèmes de tournées de véhicules**

Devant un jury composé de :

Thibaut Vidal	Professeur, Polytechnique Montréal	Rapporteur
Pamela Nolz	Professeure, St. Pölten University of Applied Sciences	Rapporteure
Claudia Archetti	Professeure, ESSEC Business School	Examinatrice
Nicolas Jozefowicz	Professeur, Université de Lorraine	Examineur (Président)
Nabil Absi	Professeur, Mines Saint-Etienne	Directeur de thèse
Dominique Feillet	Professeur, Mines Saint-Etienne	Co-directeur de thèse
Thierry Garaix	Maître assistant, Mines Saint-Etienne	Encadrant de thèse

Affidavit

Je soussigné, Maxime Agius, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Nabil Absi et Dominique Feillet, dans le respect des principes d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect de la charte nationale de déontologie des métiers de la recherche.

Ce travail n'a pas été précédemment soumis dans sa globalité en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Marseille, le 9 juillet 2023

Ce travail de thèse est une œuvre de l'esprit, protégée par le droit d'auteur, tel que prévu aux articles L111-1 du CPI et suivants disposant que « *L'auteur d'une œuvre de l'esprit jouit sur cette œuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous. [...]* »

Il est rappelé que par exception au droit d'auteur, la loi française autorise l'utilisation d'une œuvre divulguée, sans autorisateur de son auteur, suivant les conditions définies dans l'article L122-5 du CPI disposant que « *Lorsque l'œuvre a été divulguée, l'auteur ne peut interdire [...] la représentation ou la reproduction d'extraits d'œuvres, [...] sous réserve que soient indiqués clairement le nom de l'auteur et la source [...] les analyses et courtes citations justifiées par le caractère critique, polémique, pédagogique, scientifique ou d'information de l'œuvre à laquelle elles sont incorporées [...]* »

Acknowledgments

First and foremost, I would like to thank my advisors Nabil Absi, Dominique Feillet and Thierry Garaix for giving me the opportunity to work with them on an exciting research topic and for their support and guidance during these thesis years. My sincere appreciation extends to Nabil for his constant monitoring, to Dominique for always finding solutions to my issues with column generation, and to Thierry for his constant involvement despite the distance. I feel truly privileged and grateful for the opportunity to collaborate with all three of them.

I would also like to thank the members of the jury, the reviewers Pamela Nolz and Thibaut Vidal, and the examiners Claudia Archetti and Nicolas Jozefowicz, for their careful reading of the manuscript and their questions.

Furthermore, I would like to thank my colleagues and friends from the SFL department, especially Manuel, Rebecca, Mario, Thibault, Lucas, and Diana, for the good moments we shared both within and beyond the lab. Special thanks to Melek, with whom I had the pleasure of sharing my office, and to Pierre for the interesting conversations about football and programming.

Then, I would like to thank my colleagues from my year as an ATER at IUT de Metz, particularly Zsuzsanna and Anne, for their support.

I need to extend my gratitude to my favorite teacher, Yann Kieffer, for his invaluable introduction to the captivating world of Operations Research. His guidance ignited my passion for this field and played an important role in shaping my academic direction.

Also, I express my deepest appreciation to Marine, Léa, and Laura for being the best roommates I could hope for. Your support during my bad days was precious.

Last but not least, I would like to thank my dearest friends Akim, Mohamed, and my brother Samuel for the good moments shared and their support during these doctoral years. I am also grateful to the colleagues and friends not mentioned here, as well as my family.

Contents

General introduction	1
1 Thesis context and relevant literature	7
1.1 Vehicle Routing Problem and relevant variants	8
1.1.1 Definition and formulation of the Vehicle Routing Problem with Time Windows (VRPTW)	8
1.1.2 Other VRP variants relevant to this thesis	9
1.2 Branch-and-Price and Column Generation for VRP	10
1.2.1 Exact solution methods and limits of the compact formulation	10
1.2.2 Set partitioning formulation	10
1.2.3 Column Generation	11
1.2.4 Pricing problem	11
1.2.5 Branch-and-price algorithm	12
1.3 Workload equity in VRP	13
1.3.1 Illustrative example and relevant questions	13
1.3.2 Vehicle Routing Problem with Route Balancing (VRPRB)	14
1.3.3 Modeling equity, relevant definitions and contributions	15
2 A branch-and-price algorithm for a routing problem with inbound and outbound requests	19
2.1 Introduction	21
2.2 Literature review	22
2.3 Problem description and mathematical formulation	25
2.3.1 Problem description	25
2.3.2 Mathematical formulation	26
2.4 Solution method	29
2.4.1 Master problem	29
2.4.2 Column generation	30
2.4.3 Branching rules	30
2.4.4 Management of infeasible linear programs	32
2.5 Multi-Trip Shortest Path Problem with Resource Constraints	33
2.5.1 Label definition	33
2.5.2 Label initialization	34
2.5.3 Extension rules	34
2.5.4 Dominance rule	37

2.6	Computational experiments	37
2.6.1	Comparison with the VRPSPDTW literature	37
2.6.2	Generation of instances for the MTMPD-RD	40
2.6.3	Comparison against a pickup and delivery branch-and-price algorithm	41
2.6.4	Detailed analysis of MT-BP	42
2.6.5	Impact of the fleet size	45
2.6.6	Other sensitivity analyses	49
2.7	Conclusion	51
3	Workload equity in vehicle routing with a medium-term perspective	53
3.1	Introduction	55
3.2	Literature review	56
3.3	Problem description	59
3.3.1	Data discovery and decision process	60
3.3.2	Daily routing problem	60
3.3.3	Assignment problem	61
3.3.4	Equity metric and function	62
3.4	Solution frameworks	62
3.4.1	Best Equity Assignment strategy	63
3.4.2	Solution frameworks	64
3.4.3	Route planning and driver assignment with perfect information	65
3.5	Solution of the daily routing problem	66
3.5.1	Branch-and-price algorithm for the MTMPD-RD	67
3.5.2	Adaptation of the branch-and-price algorithm for driver-independent equity constraints	68
3.5.3	Adaptation of the branch-and-price algorithm for driver-dependent equity constraints	68
3.6	Computational experiments	70
3.6.1	Generation of instances	70
3.6.2	Feasibility	71
3.6.3	Trade-off between equity and cost	72
3.6.4	Daily equity and evolution along the time horizon	75
3.6.5	Computing time	77
3.6.6	Synthesis: main findings	78
3.7	Conclusion	78
4	Workload equity for a routing problem with a variable-sum equity metric	81
4.1	Introduction	83
4.2	Literature review	84
4.2.1	Equity in VRP	84
4.2.2	Constraint-based approaches for workload equity	84
4.2.3	Equity in medical transportation	85
4.3	Problem description	85
4.3.1	The MTMPD-RD: a reminder	85
4.3.2	Measure of equity	88
4.4	Solution methods	89

4.4.1	Branch-and-price algorithm for the MTMPD-RD: a reminder	90
4.4.2	Weakness of the initial model for the MTMPD-RD-WE	91
4.4.3	A Dichotomy-based Heuristic	92
4.4.4	A branch-and-price approach based on a driver-indexed model	99
4.5	Computational experiments	103
4.5.1	Benchmark instances	103
4.5.2	Results and analysis	105
4.6	Conclusion	108
5	Conclusion and perspectives	113
5.1	Conclusion	113
5.2	Perspectives	114
5.2.1	Different perception of painfulness or preferences	114
5.2.2	New exact solution method for the MTMPD-RD-WE	114
5.2.3	Application of the solution methods to other problems	115
5.2.4	Dynamic version/emergency	115
5.2.5	Equity in collaborative vehicle routing	115
A	Appendix	117
	Extended summary in French	125
	List of Figures	133
	List of Tables	135
	List of Algorithms	137
	Bibliography	139

General introduction

General context

The Vehicle Routing Problem (VRP) is one of the most studied type of problems in Operations Research both for practical and theoretical reasons. Among the large variety of applications of VRPs, equity has gained interest in the recent literature. Equity might be either service and client-oriented or provider-oriented. In the former, equity is often the main concern while in the later, both equity and cost are optimized simultaneously. In provider-oriented applications, equity consists in fairly dispatching workload among drivers. The main challenge is to find a fair distribution of workload minimizing the impact on the routing costs. To measure equity, it is necessary to define the workload metric. Workload metrics can be classified either as constant-sum, where the sum remains constant across all solutions (*e.g.*, number of customers served in the route), or variable-sum, where the sum depends on the specific solution (*e.g.*, traveled distance).

Despite the growing interest of workload equity in the recent literature, the choices of models and algorithms are not always well justified. Also, the current literature essentially concerns daily equity while drivers can accept unbalanced days if equity is ensured on a longer horizon (*e.g.*, one week, one month). In addition, a recurring weakness of the current literature is the lack of evaluation of the impact of equity on the routing costs. Hence, many research directions remain open.

This is where the thesis contributions lie: Proposing new models and algorithms to integrate equity in solutions of routing problems and evaluate the impact of equity on cost. We propose solutions for both constant-sum and variable-sum metrics and use non-emergency transportation of patients as a real-world application. We solve most of routing problems to optimality implementing branch-and-price (and column generation) algorithms.

The contributions of this thesis are (i) definition of a new routing problem motivated by the non-emergency transportation of patients and a dedicated branch-and-price algorithm to solve it, (ii) consider equity for the multi-period version of the initial routing problem with a constant-sum metric, (iii) extends the daily routing problem involving equity on a variable-sum metric. In this manuscript, we propose and evaluate new equity constraints and techniques to integrate them within a branch-and-price algorithm. Solutions are evaluated through several performance indicators (KPIs).

Project FITS (ANR)

This thesis is part of project FITS - “Flexible and Intelligent Transportation Systems”, supported by the French National Research Agency (ANR - Agence Nationale de la Recherche) under grant ANR-18-CE22-0014. The project FITS aims at proposing novel mathematical models and solution approaches in order to develop intelligent tools to help optimizing the delivery of services.

In this context, service provider platforms manage the assignment of customers' demands to human resources. Generally, these platforms need intelligent tools to guarantee a smart assignment, especially when the demand varies strongly or for large scale transportation services. The project FITS aims at tackling the scientific challenges raised by the optimization of these service provider platforms.

A common non-desired consequence of the current platforms is that the assignment creates inequities between drivers in terms of workload balancing. The objective of project FITS is to find efficient solutions while improving workload balancing. This is in this context that this thesis lies.

Structure of the thesis

Chapter 1 provides definitions, problems and algorithms related to this thesis and reviews the main contributions of the literature related to this thesis. It is divided into three parts. First, the Vehicle Routing Problem (VRP) and its variants related to our work are briefly reviewed. Then, the principle of branch-and-price and column generation algorithms are explained. Finally, the literature related to workload equity in VRP is reviewed.

Chapter 2 presents a new routing problem motivated by the non-emergency transportation of patients. Patients have appointments at the hospital and make transportation requests to be transported either from home to the hospital (inbound request) or from the hospital to home (outbound request). There are two types of requests and the problem can be modeled either as a pickup and delivery problem or a multi-trip problem. The problem is called Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD). A Branch-and-Price algorithm (BP) is developed to solve this problem. We show that the multi-trip formulation outperforms the pickup and delivery formulation. Both the MTMPD-RD and the BP algorithm are used as basis for the definition of the problems and algorithms of Chapters 3 and 4.

Chapter 3 presents the multi-period version of the MTMPD-RD integrating it within a horizon of several days where equity between drivers is considered. The problem is dynamic in the sense that requests are daily revealed. Each day, both a routing problem and a route-driver assignment problem are solved. The goal is to achieve equity on the whole horizon rather than on every day. The workload to balance fairly between drivers is the painfulness of requests, this metric is constant-sum (the total painfulness of a solution is the same for any solution). Five frameworks are proposed to ensure equity lying on an assignment strategy (potentially) coupled with daily equity constraints. The BP algorithm used to solve the daily routing problems is adapted accordingly to the type of equity constraints. The frameworks are extensively compared through numerical experiments where several KPIs are used to assess the solutions.

Chapter 4 extends the work of Chapters 2 and 3 by considering a variable-sum metric (the driving time of drivers) for the MTMPD-RD. An equity constraint is added to the model to limit the excess of driving time for each driver compared to the average driving time. An exact solution method and a heuristic both based on the BP algorithm of Chapter 2 are proposed and compared through numerical experiments. In the exact solution method, columns in the master problem are indexed by route and driver, which imposes different pricing problems for drivers. In the heuristic, the branch-and-price algorithm is iteratively solved with a fixed equity constraint and the structure of the solution guides the search for the next step. The impact of equity on the routing cost is measured through numerical experiments.

Finally, Chapter 5 summarizes the main contributions of the thesis and provides promising perspectives for future research.

Communications

The work of this thesis lead to several communications.

Journal papers

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). A branch-and-price algorithm for a routing problem with inbound and outbound requests. *Computers & Operations Research*, 146, 105896.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2023). Workload equity in vehicle routing with a medium-term perspective. (submitted to *Computers & Operations Research*, under review)

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2023). Heuristic and exact algorithms for a vehicle routing problem with route cost equity constraints. (to be submitted soon)

Conferences

International conferences

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). A Branch-and-Price algorithm for a routing problem with inbound and outbound requests. 8th meeting of the EURO Working Group on Vehicle Routing and Logistics Optimization (VeRoLog), Hamburg, Germany.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). Workload equity for a dynamic multi-period routing problem in the context of medical transportation. 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI), Mauritius Island.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2023). Branch-and-price algorithms for a vehicle routing problem with route cost equity constraints (NOW), La Spezia, Italy.

National conferences

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2020). Equity in medical transportation. 21ème congrès annuel de la ROADEF, Montpellier, France.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2021). A Branch-and-Price algorithm for a medical transportation problem. 22ème congrès annuel de la ROADEF, Mulhouse, France.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). A Branch-and-Price algorithm for a routing problem with inbound and outbound requests. 23ème congrès annuel de la ROADEF, Lyon, France.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). Workload equity for a dynamic multi-period routing problem in the context of medical transportation. 23ème congrès annuel de la ROADEF, Lyon, France.

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2023). Workload equity in vehicle routing with a medium-term perspective. 24ème congrès annuel de la ROADEF, Rennes, France.

Awards

Winner of the best poster award for:

Agius, M., Absi, N., Feillet, D., & Garaix, T. (2022). Workload equity in vehicle routing with a medium-term perspective. Colloque IMT « Les sciences de l'information au service des nouvelles mobilités », Palaiseau, France.

Thesis context and relevant literature

In this chapter, we present and review important works related to the content of this thesis. The goal is not to provide an exhaustive review of the literature but rather to give important definitions and an overview of problems and algorithms related to the thesis. First, we present the Vehicle Routing Problem (VRP) and briefly review literature on the VRP and its variants related to our work, *i.e.* Pickup and Delivery Problem (PDP), multiple trip VRP (MT-VRP) and Dial-a-Ride Problem (DARP). Then, we introduce the principle of branch-and-price and column generation algorithms and present some important contributions that improved the efficiency of these algorithms. Finally, we review the literature related to workload equity in VRP, we synthesise the main contributions in the field and introduce some important definitions.

Contents

1.1	Vehicle Routing Problem and relevant variants	8
1.1.1	Definition and formulation of the Vehicle Routing Problem with Time Windows (VRPTW)	8
1.1.2	Other VRP variants relevant to this thesis	9
1.2	Branch-and-Price and Column Generation for VRP	10
1.2.1	Exact solution methods and limits of the compact formulation	10
1.2.2	Set partitioning formulation	10
1.2.3	Column Generation	11
1.2.4	Pricing problem	11
1.2.5	Branch-and-price algorithm	12
1.3	Workload equity in VRP	13
1.3.1	Illustrative example and relevant questions	13
1.3.2	Vehicle Routing Problem with Route Balancing (VRPRB)	14
1.3.3	Modeling equity, relevant definitions and contributions	15

1.1 Vehicle Routing Problem and relevant variants

The Vehicle Routing Problem (VRP) was introduced by [Dantzig and Ramser \(1959\)](#) and consists basically in planning vehicle routes to serve customer requests with a minimal total routing cost.

The most classic and studied VRP is the Capacitated Vehicle Problem (CVRP) in which the fleet size is fixed and vehicles have a homogeneous capacity that limits the total load served by a single vehicle ([Ralphs et al., 2003](#); [Marinakis et al., 2018](#)). A well-known extension of the CVRP is the Vehicle Routing Problem with Time Windows (VRPTW). In the VRPTW, a time window $[a_i, b_i]$ is associated with each customer request i : customer i must be served after time a_i and before time b_i ([Cordeau et al., 2000](#); [Desaulniers et al., 2014](#)).

1.1.1 Definition and formulation of the Vehicle Routing Problem with Time Windows (VRPTW)

Formally, the VRPTW is defined on a complete directed graph $G = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_0, \dots, v_n\}$. Node v_0 corresponds to the depot where vehicles are located while nodes $v_i \neq v_0$ correspond to customer locations. Travel times t_{ij} and travel costs c_{ij} are defined for every arc $(v_i, v_j) \in \mathcal{A}$. With every customer $v_i \in \mathcal{V} \setminus \{v_0\}$, is associated a positive demand d_i , a time window $[a_i, b_i]$ and a positive service time s_i . A homogeneous fleet of K vehicles of capacity Q is available for serving the customers. All customers must be served by exactly one vehicle route and all routes must be valid. To be valid a route must respect the following constraints: the vehicle must begin and end its route at the depot within a time horizon $[a_0, b_0]$; the vehicle load must not exceed its capacity Q ; the service of a customer has to start within its time window (and service time s_i must be counted). The vehicle is allowed to arrive at customer v_i before time a_i but has to wait this time to start servicing. Conventionally we set $d_0 = 0$ and $s_0 = 0$. The objective of the VRPTW is to find a minimum cost set of at most K routes visiting exactly once each customer with respect to the previous constraints. The VRPTW can be formulated as a Mixed-Integer Linear Program (MILP) as follows:

$$\min \sum_{(v_i, v_j) \in \mathcal{A}} \sum_{1 \leq k \leq K} c_{ij} x_{ij}^k \quad (1.1)$$

subject to

$$\sum_{1 \leq k \leq K} \sum_{v_j \in \mathcal{V} \text{ s.t. } (v_i, v_j) \in \mathcal{A}} x_{ij}^k \geq 1 \quad \forall v_i \in \mathcal{V} \setminus \{v_0\} \quad (1.2)$$

$$\sum_{v_i \in \mathcal{V} \text{ s.t. } (v_0, v_i) \in \mathcal{A}} x_{0i}^k \leq 1 \quad 1 \leq k \leq K \quad (1.3)$$

$$\sum_{v_j \in \mathcal{V} \text{ s.t. } (v_i, v_j) \in \mathcal{A}} x_{ij}^k = \sum_{v_j \in \mathcal{V} \text{ s.t. } (v_j, v_i) \in \mathcal{A}} x_{ji}^k \quad \forall v_i \in \mathcal{V}, 1 \leq k \leq K \quad (1.4)$$

$$\sum_{(v_i, v_j) \in \mathcal{A}} d_i x_{ij}^k \leq Q \quad 1 \leq k \leq K \quad (1.5)$$

$$\tau_j^k \geq \tau_i^k + s_i + t_{ij} - M(1 - x_{ij}^k) \quad \forall (v_i, v_j) \in \mathcal{A}, v_j \neq v_0, 1 \leq k \leq K \quad (1.6)$$

$$b_0 \geq \tau_i^k + s_i + t_{i0} - M(1 - x_{i0}^k) \quad \forall (v_i, v_0) \in \mathcal{A}, 1 \leq k \leq K \quad (1.7)$$

$$a_i \leq \tau_i^k \leq b_i \quad \forall v_i \in \mathcal{V}, 1 \leq k \leq K \quad (1.8)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (v_i, v_j) \in \mathcal{A}, 1 \leq k \leq K \quad (1.9)$$

where x_{ij}^k and τ_i^k are the decision variables. Variable x_{ij}^k equals 1 if route k uses arc (v_i, v_j) , 0 otherwise. If customer v_i is served by route k then, variable τ_i^k corresponds to the starting time of the service of customer i by route k , otherwise τ_i^k is meaningless. Value M must be large enough to ensure validity of constraints (1.6) and (1.7).

The objective function (1.1) minimizes the total travel cost. Constraints (1.2) ensure that every customer is visited exactly once. Constraints (1.3) and (1.4) are the flow conservation constraints. Constraints (1.5) are the capacity constraints while Constraints (1.6)-(1.8) compute the times of service of customers and check the time windows constraints. Finally (1.9) define the arc flow variables. In the literature many other model are proposed for the VRPTW (Irnich et al., 2014; Marinakis et al., 2018; Feillet, 2010). Note that a model for the CVRP can be obtained from (1.1)-(1.9) by removing variables τ_i^k and time constraints (1.6)-(1.8).

1.1.2 Other VRP variants relevant to this thesis

Despite the academic interest of the CVRP and the VRPTW, real life applications of VRP require the definition of more specific problems. Hence, many variants of the VRP have raised in the literature and VRPs have become one of the most studied class of problems in Operations Research (Toth and Vigo, 2014; Laporte, 2009).

Variants of VRPs can be classified according to many characteristics such as the type of requests, the fleet composition and location, the optimization objective or the network structure as explained by Irnich et al. (2014). Usually a variant is characterized by specific constraints that enrich/modify the simple model (1.1)-(1.9). Many survey on variants of the VRP exists in the literature. Among others, Marinakis et al. (2018); Irnich et al. (2014) present the most important variants with mathematical formulations for some of them while the recent review from Vidal et al. (2020) classify the existing and emerging variants.

In this thesis, the studied problem of Chapter 2 (that is reused in Chapters 3 and 4) can be modeled either as Pickup and Delivery Problem (PDP) or as a Multi-trip VRP (MT-VRP).

Basically, a PDP is a VRP where each request is characterized by two nodes, a pickup node where the customer commodity is loaded in the vehicle and a corresponding delivery node where it is dropped off (Desaulniers et al., 2002). Behind this generic definition, the PDP can be classified into three families as surveyed in Berbeglia et al. (2007):

- Many-to-many problems (M-M): commodities may have multiple origins and destinations and a location may be the origin or destination of multiple commodities.
- One-to-many-to-one (1-M-1): some commodities are delivered from the depot to customers and other commodities are picked up at the customers' locations and returned back to the depot.
- One-to-one problems (1-1): a commodity has a single origin and a single destination.

Readers are referred to Battarra et al. (2014) for a recent survey on PDP. The studied problem in Chapter 2 is classified within the 1-M-1 family and will be reviewed in Section 2.2.

In some applications, demands are high, vehicle capacity limited and travel times are short. So, with a limited number of vehicles available, they need to be reused and perform multiple trips. This class of problems is called Multi-trip VRP (MT-VRP) and was introduced by Fleischmann (1990). Cattaruzza et al. (2016) surveys the MT-VRP.

This thesis is applied to the context of non-emergency transportation of patients: patients need to go to hospital for appointment and come back home but they cannot travel by their own. In this context,

transportation companies ensure the service. This type of problem can be classified as a Dial-A-Ride Problem (DARP). [Cordeau and Laporte \(2003\)](#) states that the DARP is a special pickup and delivery problem with time windows for transportation of people and with extra client convenience constraints: i) the ride time of a user must not exceed a maximal duration L and routes must respect a maximum route duration. [Cordeau and Laporte \(2007\)](#) provides mathematical formulations of the DARP. Literature on the DARP has been largely enriched with variants for each specific applications, [Molenbruch et al. \(2017\)](#) and [Doerner and Salazar-González \(2014\)](#) provide comprehensive classifications of the variants and solution methods used in the DARP literature.

PDP, MT-VRP and DARP will be reviewed in more details in Chapter 2 (Section 2.2).

1.2 Branch-and-Price and Column Generation for VRP

For solving a vehicle routing problem, we distinguish two types of solution methods, exact methods and (metha)heuristics ([Toth and Vigo, 2014](#)). The former ensure to find an optimal solution while the latter do not. (Metha)heuristics are to be preferred when finding a good solution within a short amount of time is required, typically for solving real-world applications. In this thesis, we rather prefer to ensure finding optimal solutions when possible. Hence, we choose to develop an exact solution method to solve the initial daily routing problem, the MTMPD-RD (see Chapter 2). This algorithm is then adapted to suit all studied variants of the MTMPD-RD involving equity constraints (see Chapters 3 and 4).

1.2.1 Exact solution methods and limits of the compact formulation

VRPs are known to be NP-hard problems. Solving to optimality a VRP using a classic branch-and-bound algorithm is usually not efficient ([Laporte and Nobert, 1983](#); [Fischetti et al., 1994](#); [Toth and Vigo, 2002](#)). Formulation (1.1)-(1.9) is called a compact formulation. This formulation and all its variants have a weak linear relaxation. So, when solving a VRP with a branch-and-bound algorithm based on this type of formulation, the tree search requires to explore a huge number of nodes ([Feillet, 2010](#)).

To get a stronger bound, other models obtained using Dantzig-Wolfe decomposition have been investigated. These models are called extended formulations and have the structure of set partitioning formulations.

1.2.2 Set partitioning formulation

In the set partitioning formulation, we have a single type of decision variables corresponding to feasible routes. Let Ω be the set of feasible routes. These routes satisfy capacity and time constraints. Customer locations are visited at most once. Let $a_i^r = 1$ if route $r \in \Omega$ serves request $i \in \mathcal{V} \setminus \{v_0\}$, 0 otherwise. Let $b_{ij}^r = 1$ if route r uses arc $(v_i, v_j) \in \mathcal{A}$, 0 otherwise ($c_r = \sum_{(v_i, v_j) \in \mathcal{A}} b_{ij}^r c_{ij}$ for all $r \in \Omega$). We introduce a non-negative integer decision variable θ_r equals to the number of times route $r \in \Omega$ is selected in the solution. The set partitioning formulation (SP) is defined as follows:

$$\min \sum_{r \in \Omega} c_r \theta_r \tag{1.10}$$

subject to

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{V} \setminus \{v_0\} \quad (1.11)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (1.12)$$

$$\theta_r \in \mathbb{N} \quad \forall r \in \Omega \quad (1.13)$$

The objective function (1.10) minimizes the total cost. Constraints (1.11) are the set partitioning constraints, they ensure that each customer is served exactly once. Constraint (1.12) imposes to build at most K routes.

The linear relaxation of (1.10)-(1.13) defines the master problem. In practice, the master problem (MP) provides a strong lower bound but it cannot be solved directly due to the huge amount of variables (one per feasible route). This is in this context that Desrosiers et al. (1984) use column generation to solve the linear relaxation.

1.2.3 Column Generation

As the master problem contains too many variables, they cannot be explicitly enumerated. The column generation algorithm tackles a restricted master problem that it iteratively enriches until an optimality criterion is met. Given an iteration t and a subset $\Omega_t \subseteq \Omega$, we call $MP(\Omega_t)$, the master problem restricted to the subset Ω_t of routes. $MP(\Omega_t)$ can be solved (by the simplex algorithm), which also provides a feasible solution to the (unrestricted) master problem. The condition for this solution to be optimal is that the master problem does not contain any variable with a negative reduced cost. Having a negative reduced cost can occur for variables associated with routes in $\Omega \setminus \Omega_t$, as they were not considered in the solution of the restricted master problem. Finding such variables relies on the solution of a pricing problem (see next paragraph). If some are found, they are added to the restricted master problem and a new iteration is started. The process ends when no variable with negative reduced cost is found. Figure 1.1 describes the principle of the column generation algorithm.

1.2.4 Pricing problem

From the master problem (1.10)-(1.13), we deduce the reduced cost of a route. Let λ_i be the unsigned dual variables associated with Constraints (1.11), one variable per customer $v_i \in \mathcal{V} \setminus \{v_0\}$, and λ_0 be the nonpositive dual variable associated with Constraint (1.12). The reduced cost of a route $r \in \Omega$ is:

$$\bar{c}_r = c_r - \sum_{v_i \in \mathcal{V} \setminus \{v_0\}} a_i^r \lambda_i - \lambda_0 = \sum_{(v_i, v_j) \in \mathcal{A}} b_{ij}^r (c_{ij} - \lambda_i)$$

In the column generation algorithm, looking for routes with a negative reduced cost is called the subproblem or pricing problem. Generally, the pricing problem is modeled either as a Shortest Path Problem with Resource Constraints (SPPRC) (Desrochers et al., 1992) or an Elementary Shortest Path Problem with Resource Constraints (ESPPRC) (Feillet et al., 2004). In both models, the principle is to find paths from the depot to the depot, satisfying capacity and time constraints with a negative cost, where costs on arcs are set to $c_{ij} - \lambda_i$. The difference is that in the SPPRC, routes are not necessarily elementary so, a customer can be visited several times in a single route. The advantage of the SPPRC is to be only weakly NP-hard contrary to the ESPPRC. However, solving the pricing problem as an SPPRC decreases the quality of the linear relaxation of the master problem.

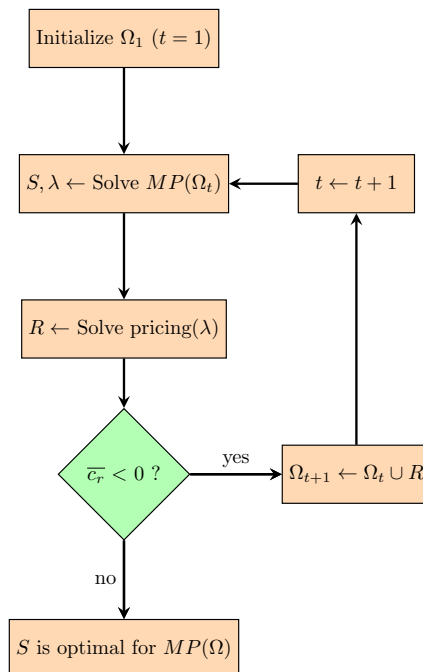


Figure 1.1: Column generation principle

In practice, dynamic programming algorithms are efficient to solve the pricing problem. Also, they allow finding in the same iteration many routes with a negative reduced cost. These algorithms are based on labels that correspond to partial paths starting from the depot. A label is characterized by its level of resource consumption, basically: the last node visited, the cumulative load, the time of last service and the cost of the current path, in the context of the VRPTW. Labels are extended checking resource constraints until the best feasible paths are obtained. Dominance rules are used to discard dominated labels and accelerate the algorithm. Given two labels L^1 and L^2 corresponding to two partial paths arriving at the same node, L^1 dominates L^2 if all resources are less or equally consumed in L^1 compared to L^2 .

1.2.5 Branch-and-price algorithm

A branch-and-price algorithm (BP) consists in solving a branch-and-bound algorithm (BB) where the linear relaxation is solved involving column generation (CG). The branching rules have to be compatible with the column generation algorithm. The classic branching rule for VRPs consists in: first computing flows on arcs, $f_{ij} = \sum_{r \in \Omega} b_{ij}^r \theta_r$ for any $(v_i, v_j) \in \mathcal{A}$; then, selecting an arc $(v_i, v_j) \in \mathcal{A}$ with a fractional flow ($0 < f_{ij} < 1$) and deriving two branches:

- one branch where arc (v_i, v_j) is forbidden in the solution.
- one branch where arc (v_i, v_j) is enforced in the solution.

The advantage of this scheme is that these branching rules can be handled without introducing new constraints and dual variable in the master problem.

Readers are referred to [Feillet \(2010\)](#) for a more complete tutorial on Branch-and-Price algorithms. Since the seminal paper of [Desrosiers et al. \(1984\)](#), many papers contributed improving Branch-and-Price

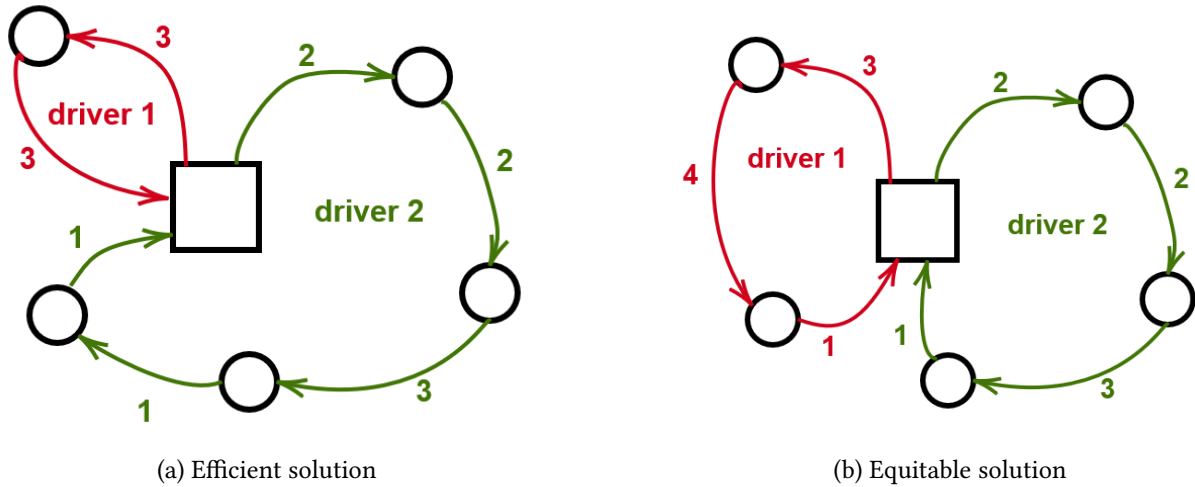


Figure 1.2: Example with two solutions: an efficient and an equitable

and Column Generation algorithms. A classic technique consists in adding valid cuts to strengthen the bound of the relaxation. Among others, [Kohl et al. \(1999\)](#) and [Jepsen et al. \(2008\)](#) introduce the k -path cuts and the subset-row inequalities respectively. With cut generation, the algorithms are usually called branch-and-cut-and-price. [Baldacci et al. \(2012\)](#) reviews exact methods for solving the CVRP and the VRPTW. A more recent paper from [Costa et al. \(2019\)](#) surveys the main contributions to exact methods for VRPs, whether they are generic or specific to a VRP variant.

1.3 Workload equity in VRP

The goal of this section is not to do a complete review of equity for VRPs. The recent survey by [Matl et al. \(2018\)](#) does it extensively. We rather try to synthesize the main contributions in the field and give definitions, properties and tools required to read the rest of this manuscript and particularly chapters 3 and 4. Readers are referred to [Matl et al. \(2018\)](#) for more details.

We start illustrating equity in VRP through a simple example and enumerate important questions arising from equity in VRPs (Section 1.3.1). Then, we present the VRP with Route Balancing (VRPRB) which is the most studied related problem in the literature (Section 1.3.2) and show the weaknesses of this model. Then, we introduce important definitions, models and properties as well as the main equity metrics and equity functions from the literature (Section 1.3.3).

1.3.1 Illustrative example and relevant questions

When solving a routing problem minimizing routing costs, we get an optimal solution that can suffer from inequity, that is, workload might be distributed unfairly between drivers. Yet, there might exist more equitable solutions with a small additional total travel cost. Figure 1.2 illustrates this through a simple example on a small instance composed of two drivers, a single depot and 5 customers. The square corresponds to the depot, circles correspond to customers.

Figure 1.2 shows two solutions for this instance: on the left, the efficient solution and on the right, the equitable solution. In the efficient solution, the total routing cost is 15, driver 1 serves 1 customer and their route cost is 6 while driver 2 serves 4 customers and their route cost is 9. In the equitable solution, the total

routing cost is 16, driver 1 serves 2 customers and their route cost is 8 while driver 2 serves 3 customers and their route cost is 8. So, the routing cost of the efficient solution is slightly better (15 against 16) but it is clearly less equitable both in terms of number of customers served and routing cost. In a context where equity is a crucial criterion, which is often true in practice, the equitable solution should be preferred. Even though the conclusions are obvious on the simple example of Figure 1.2, several questions need to be answered:

- Which workload resource has to be balanced fairly?
- How to measure equity (or inequity) on the selected workload resource?
- In which application workload is equity relevant?
- What is the impact of equity on the properties, structure and cost of solutions?
- Is it necessary to have equity every day on a planning horizon of several days?
- How to integrate equity in mathematical models and solution methods?
- What if drivers have different perceptions of equity?

Some of these questions have largely been answered in the literature (see Section 1.3.3). Others remain at least partially unanswered. The goal of this thesis is to study some of them.

1.3.2 Vehicle Routing Problem with Route Balancing (VRPRB)

In the literature on equity for VRPs, the VRP with Route Balancing (VRPRB) is probably the most studied problem. The VRPRB is the bi-objective version of the CVRP where the second objective is to minimize the range of route length (difference between the cost of the longest and the shortest route of the solution). The problem was introduced by Jozefowicz et al. (2002) and has been largely studied for almost two decades (Jozefowicz et al., 2007; Borgulya, 2008; Lacomme et al., 2015). Oyola and Løkketangen (2014) provides a formal definition and a mathematical formulation of the VRPRB.

By extension, the VRPRB with time windows is the bi-objective version of the VRPTW and has also been investigated in the literature (Baños et al., 2013). Basically, the range equity objective can be integrated to model (1.1)-(1.9) by adding two new non negative variables L and S corresponding to the cost of the longest and shortest route of the solution respectively, set with constraints

$$S \leq \sum_{(v_i, v_j) \in \mathcal{A}} c_{ij} x_{ij}^k \leq L \quad 1 \leq k \leq K \quad (1.14)$$

and adding the secondary objective $\min L - S$.

Jozefowicz et al. (2002) point out that for Pareto optimal solutions, some routes can be artificially worsen to improve the workload balance. This undesirable property actually occurs in many VRPs with equity and reveals some modelling issues. These issues are discussed in Section 1.3.3. The VRPRB has been studied with other relevant equity functions: the min-max in Bertazzi et al. (2015) or the lexicographic min-max in Lehuédé et al. (2020).

1.3.3 Modeling equity, relevant definitions and contributions

When considering equity in a routing problem, one needs to decide which workload resource must be balanced fairly. Following [Matl et al. \(2018\)](#), we call it the equity metric. The most studied metrics are the routing cost (either distance or time), the number of customers served, the total customer load served or the service time at customer location. The choice of the equity metric should be taken appropriately according to the application, for example the number of stops is probably the most relevant metric for last mile delivery applications while the service time is more relevant in home maintenance applications. Equity metrics can be classified within two families:

- Constant-sum metric: for any feasible solution, the sum of workload assigned to drivers remains constant (e.g., number of customers served, total customer load served, service time).
- Variable-sum metric: the sum of workload assigned to drivers differs between solutions (e.g., routing cost/distance/time).

Generally, constant-sum metrics are expressed with values associated with the nodes of the network while variable-sum metrics are expressed with values associated with the edges/arcs ([Matl et al., 2019](#)).

When introducing equity, the VRP formally becomes a biobjective problem ([Jozefowicz et al., 2008](#); [Zajac and Huber, 2021](#)). The first objective is the classic cost minimization and the second is the equity objective (also named equity function or equity measure), which can take different forms. The two objectives can however be considered in different ways:

- Bi-objective model: two explicit objectives and generation of the Pareto front.
- Weighted-sum: the two objectives are combined within a single objective function with weights $\alpha \times \text{obj}_1 + \beta \times \text{obj}_2$. Usually, $\alpha + \beta = 1$.
- Constraint-based: cost minimization is the single objective and equity is handled with constraints.
- Primary objective: the equity objective is the single objective.

The choice of an equity function is not trivial. Some papers in the literature focus their study on a specific equity function (like [Golden et al. \(1997\)](#) and [Bertazzi et al. \(2015\)](#) with the min-max objective and [Lehuédé et al. \(2020\)](#) with the the lexicographic min-max), others compare several equity objectives within a bi-objective model (like [Halvorsen-Weare and Savelsbergh \(2016\)](#) and [Matl et al. \(2018\)](#)).

[Halvorsen-Weare and Savelsbergh \(2016\)](#) studied the bi-objective version of the Mixed Capacitated General Routing Problem (MC-GRP) ([Pandi and Muralidharan, 1995](#)) where the classic min-cost objective is combined with an equity objective. They studied the route length as metric and selected four equity functions and compare the generated Pareto fronts. Authors show that the choice of the equity objective has a significant impact on the Pareto front and suggest that the range function is a robust choice.

More recently, the work of [Matl et al. \(2018\)](#) largely contributed to the literature on workload equity for VRPs: i) they review the literature, ii) they conduct a theoretical analysis of the most relevant equity functions, iii) they conduct numerical experiments on small instances generating the Pareto fronts for each equity function and compare them. This paper is up to now the reference paper about equity for routing problems. We present in the following paragraphs the main definitions, properties and results of this paper.

Their review summarizes for VRP with workload equity papers, the important modelling choices: the type of equity function (range, min-max or other), the equity metric (route length or demand served), the

optimization model (multi-objective, weighted-sum, constraint-based or primary objective) and the type of method (exact or heuristic).

In their theoretical analysis, the authors provide a list of eight desirable properties (axioms) for equity functions. After that, the authors list six commonly used functions and determine which axioms are respected by which functions. The studied equity functions are the min-max, the lexicographic min-max, the range, the mean absolute deviation, the standard deviation and the Gini coefficient.

Before presenting the main results, let us introduce some notations. Given a routing problem and an instance with n customers and K vehicles, a feasible solution S is a set $\{r_1, r_2, \dots, r_K\}$ of K feasible routes that cover the customers; we denote $C(S)$ the routing cost of the solution. With regards to a given equity metric, each route/driver k has a workload w_k . We call $W = \{w_1, w_2, \dots, w_K\}$, the workload allocation of the solution. We denote $I(W)$ the equity function to be minimized.

Among the studied equity functions in the literature, the range and the min-max are probably the most recurrent. Let us define them:

- range: minimization of the difference between the maximal and the minimal workload

$$\min I(W) = \max_{k \in \{1, \dots, K\}} w_k - \min_{k \in \{1, \dots, K\}} w_k$$

- min-max: minimization of the maximal workload

$$\min I(W) = \max_{k \in \{1, \dots, K\}} w_k$$

Among the eight axioms presented by [Matl et al. \(2018\)](#), two are particularly important: monotonicity and the Pigou-Dalton Transfer Principle:

- Monotonicity: given two solutions S and S' with workload allocation W and W' , and introducing $\delta_k = w'_k - w_k$ for all k in $\{1, \dots, K\}$. An equity function satisfies monotonicity if and only if: $\delta_k \geq 0$ for all k implies $I(w') \geq I(w)$. Informally, it states that if one or more outcomes are strictly worsened while leaving all others unchanged, then inequity should increase (strong monotonicity) or at least not be reduced (weak monotonicity).
- Pigou-Dalton Transfer Principle: given two solutions S and S' with workload allocation W and W' being formed as follows: $w'_i = w_i - \delta$, $w'_j = w_j + \delta$ and $w'_k = w_k$ for all $k \notin \{i, j\}$. An equity function satisfies the Pigou-Dalton Transfer Principle if and only if: $\delta \geq 0$ and $w_j \geq w_i$ implies $I(w') \geq I(w)$. Informally, it states that taking from the poorer and giving to the richer should lead to a numerical increase in inequality (strong version), or at least not reduce it (weak version).

Authors point out some important definitions and properties:

- The monotonicity axiom is relevant only when the metric is variable-sum while Pigou-Dalton Transfer Principle is relevant only when the metric is constant-sum. Table 1.1 sums-up these properties.
- The monotonicity axiom is respected only for the min-max and the lexicographic min-max objectives, these measures are said monotonic, the others non-monotonic.
- For any combination of a variable-sum metric with a non-monotonic equity function, an optimal solution can contain non-TSP optimal routes (e.g., range).
- A solution S of workload allocation W is said workload inconsistent if there exists another solution S' of workload allocation W' such that $C(S) > C(S')$ and $w_k \geq w'_k$ for all k but $I(W) < I(W')$.

metric	Monotonicity	Transfert Principle
constant-sum	irrelevant	useful
variable-sum	critical	marginal

Table 1.1: Satisfaction of the monotonicity axiom and the Pigou–Dalton transfer principle according to the type of equity metric

- For any combination of a variable-sum metric with a non-monotonic equity function, to avoid non-TSP optimal tours, one needs to impose TSP-optimal routes. This does not guarantee workload consistency anyway.

Important conclusions of [Matl et al. \(2018\)](#) are that none of the inequity measure respect all axioms hence, there exists no "perfect" inequity measure. In contradiction with [Halvorsen-Weare and Savelsbergh \(2016\)](#), they claim that the choice of the range measure combined with a variable-sum metric is not appropriate because it causes non-TSP routes and workload inconsistency. These conclusions are consistent with the remark of [Jozefowicz et al. \(2002\)](#) that noted that solutions of the VRPRB artificially deteriorate the solution cost to improve the equity measure.

[Matl et al. \(2018\)](#) enrich their paper with numerical experiments where they generate the optimal Pareto front on small instances (at most 14 customers) with all equity functions and the route cost as equity metric. Numerical results reinforce their conclusions.

Another paper from the same authors ([Matl et al., 2019](#)) generalizes their work to different equity metrics. They solve larger instances (up to 200 customers) with a heuristic and evaluate both variable-sum and constant-sum equity metrics. Their study reveals that the equity metric has more impact on the solutions than the equity function. All these definitions and properties must be used carefully when modeling equity in VRP.

A branch-and-price algorithm for a routing problem with inbound and outbound requests

In this chapter, we present a new problem arising in the context of non-emergency transportation of patients. We consider a hospital (the depot) and a set of patients with a medical appointment. Patients require either to go from home to hospital (inbound request) or from hospital to home (outbound request). The problem can be addressed as a Pickup and Delivery Problem, but the fact that all transportation requests are connected to the depot also allows tackling it as a special Multi-Trip Vehicle Routing Problem. We adopt the second standpoint and call it the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates. We propose a specialized branch-and-price algorithm and demonstrate computationally that our approach outperforms a classical branch-and-price algorithm based on the Pickup and Delivery Problem modeling. We also show how our algorithm can be adapted to the solution of the Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Windows, and obtain new optimal solutions on benchmark instances.

Contents

2.1	Introduction	21
2.2	Literature review	22
2.3	Problem description and mathematical formulation	25
2.3.1	Problem description	25
2.3.2	Mathematical formulation	26
2.4	Solution method	29
2.4.1	Master problem	29
2.4.2	Column generation	30
2.4.3	Branching rules	30
2.4.4	Management of infeasible linear programs	32
2.5	Multi-Trip Shortest Path Problem with Resource Constraints	33
2.5.1	Label definition	33
2.5.2	Label initialization	34

2.5.3	Extension rules	34
2.5.4	Dominance rule	37
2.6	Computational experiments	37
2.6.1	Comparison with the VRPSPDTW literature	37
2.6.2	Generation of instances for the MTMPD-RD	40
2.6.3	Comparison against a pickup and delivery branch-and-price algorithm	41
2.6.4	Detailed analysis of MT-BP	42
2.6.5	Impact of the fleet size	45
2.6.6	Other sensitivity analyses	49
2.7	Conclusion	51

2.1 Introduction

Since the seminal paper of [Dantzig and Ramser \(1959\)](#) that introduced the Vehicle Routing Problem (VRP), thousands of research papers dealing with different variants of the VRP have been published. A wide range of formulations and solution approaches have been proposed and allow to tackle many types of real-world applications for the transportation of goods or people. Readers are referred to the book by [Toth and Vigo \(2014\)](#) for a relatively recent overview of the field.

In this chapter, we are interested in the problem of transporting patients to hospitals for medical appointments, and back home after their appointments. Vehicle routing problems in the context of medical transportation have been widely studied, especially in the context of emergency services. Works dealing with non-emergency transportation, like ours, are scarcer. We consider a hospital and a set of patients with a medical appointment, that have either to be picked up at home and transported to the hospital or the other way around. Appointments define time constraints that we call release dates (time at which patients are available for transportation) and due dates (time at which they must be arrived at their destination). Vehicles are available at the hospital to achieve these tasks and for more efficiency, patients can be regrouped in the same vehicle, as long as time constraints and the vehicle capacity are satisfied.

Clearly, the studied problem is a special Pickup and Delivery Problem (PDP), with every patient being moved from a pickup point to a delivery point. As transportation requests concern persons, it is also a special Dial-A-Ride Problem (DARP). However, the fact that all transportation requests start or end at the hospital makes this PDP very specific. Actually, the problem can also be viewed as a special multi-trip VRP (MTVRP), with vehicles performing multiple trips from the depot (the hospital). We adopt this standpoint and demonstrate that it helps solving the problem more efficiently.

Seen as a multi-trip VRP, the problem, however, raises some important issues. First, trips combine two types of requests: inbound requests, for patients that will be picked-up at their home location and transported to the hospital, and outbound requests, for patients that are initially at the hospital and need to be transported back home. It implies a complex management of the vehicle capacity, similar to that encountered in PDPs. Second, operations at home locations are not only constrained in time by the release date (resp., due date) at this node, but also by the due date (resp., release date) at the hospital, which implies a complex time management. Given the existing literature, our problem can be identified as a Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates. In the following, we coin it MTMPD-RD.

The main contributions of this paper are threefold. First, we introduce a new variant of the VRP, the MTMPD-RD, which is motivated by the non-emergency patient transportation. Second, we present a branch-and-price algorithm specifically designed for this problem and we evaluate it through an extensive computational campaign; in particular we show the benefits of our approach against an off-the-shelf PDP branch-and-price algorithm and against a two-phase approach in which mono-trip routes are generated first and a trip-to-vehicle assignment procedure is applied second. Third, we show how our algorithm can be used to solve the Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Windows, and we close some benchmark instances for this problem.

The paper is organized as follows: Section 2.2 reviews the literature related to the MTMPD-RD. Section 2.3 describes the problem and provides a mathematical formulation. Section 2.4 describes the proposed solution method, a branch-and-price algorithm with a column generation to solve the linear relaxation. Section 2.5 describes the pricing problem used in the column generation algorithm. Section 2.6 details the used instances and the experiments and discusses obtained results. Finally, Section 2.7 concludes this paper, summarizing our findings and suggesting possible extensions to this work.

2.2 Literature review

In this section, we position the MTMPD-RD in the literature. As briefly introduced in Section 2.1, the MTMPD-RD is a variant of the PDP, and of the MTRVP. Also, it concerns the non-emergency transportation of patients and can be assimilated to a DARP. In the following, we respectively position our problem regarding the PDP, the MTRVP, the DARP and the non-emergency transportation literature.

Positioning regarding the PDP literature

PDP have met a considerable interest in the literature. For complete states of the art, interested readers are referred to [Berbeglia et al. \(2007\)](#) and [Battarra et al. \(2014\)](#).

[Berbeglia et al. \(2007\)](#) propose a classification of the different variants of the PDP. PDPs are classified according to the route structure to be considered and the type of demand. Each variant can further be divided into two variants, a single vehicle and a multiple-vehicle variant. The three main classes classified according to the route structure are the many-to-many problems (M-M), the one-to-many-to-one problems (1-M-1), and the one-to-one problems (1-1). In M-M problems, a commodity may have multiple origins and destinations and a location may be the origin or destination of multiple commodities. This class is the least studied in the literature as it is the one with the least concrete applications. In 1-M-1 problems, some commodities are delivered from the depot to customers and other commodities are picked up at the customers' locations and returned back to the depot. The 1-M-1 has several applications. For example, in the beverage industry, full bottles are delivered to customers while empty ones are returned to the depot ([Privé et al., 2006](#)). In 1-1 problems, a commodity has a single origin and a single destination. This problem arises in the context of urban carrier operations and maritime shipping, among others, and also in the context of passenger transportation, where it is defined as a DARP ([Cordeau and Laporte, 2003](#); [Cordeau, 2006](#)). A recent review ([Molenbruch et al., 2017](#)) provides a comprehensive classification of the DARP literature.

Clearly, the MTMPD-RD is classified within the 1-M-1 class, since commodities are transported either from the depot to customers or from customers to the depot. The 1-M-1 problems can be classified according to the demand type. If customers ask for a simultaneous service, *i.e.*, a combined service of pickup and delivery with at least one customer asking for both a pickup and a delivery, the problem is called the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). The VRPSPD is the most generic and studied variant of 1-M-1 problems. Interested readers are referred to [Koç et al. \(2020\)](#) for a complete literature review. If every customer requires either a pickup or a delivery, but not both, the problem is said single demand. The MTMPD-RD is a single-demand problem.

In the 1-M-1 with single demand, two classes can be derived depending on the way commodities are handled. When all deliveries must be served first followed by all pickups, the problem is said with backhaul. When pickups and deliveries can be mixed, the problem is said mixed (*e.g.*, Vehicle Routing Problem with Mixed Pickup and Delivery (VRPMPD), see for example [Mosheiov \(1998\)](#)). The VRPMPD is actually a special case of the VRPSPD in which no customer makes a combined demand. So a valid solution method for the VRPSPD is also valid for the VRPMPD.

Very few exact solution methods have been developed for the VRPSPD and the VRPMPD. [Subramanian et al. \(2013\)](#) propose a branch-and-cut-and-price algorithm to solve the VRPSPD. Instances with up to 100 customers are solved optimally. To the best of our knowledge, it is currently the best exact solution for the VRPSPD, as well as for the VRPMPD. The MTMPD-RD is a VRPMPD in which we add release and due dates and the possibility of performing multiple trips.

In the literature, the VRP with Simultaneous Pickup and Delivery and Time Windows (VRPSPDTW) is the closest problem to the MTMPD-RD. However, the VRPSPDTW does not allow multiple trips. To the best of our knowledge, [Angelelli and Mansini \(2002\)](#) were the first authors to propose a method to solve the VRPSPDTW and until now, they are the only ones who proposed an exact method. They developed a branch-and-price algorithm to solve the VRPSPDTW minimizing the total cost. They tested their method on modified Solomon's instances (from [Solomon \(1987\)](#), initially designed for the Vehicle Routing Problem with Time Windows (VRPTW)) with up to 20 customers.

Since then, several authors have proposed heuristics to solve the VRPSPDTW. In these heuristics, the VRPSPDTW is usually defined as a hierarchical bi-objective problem: a primary objective minimizes the number of vehicles used and a secondary objective minimizes the total distance. [Wang and Chen \(2012\)](#) introduce new instances from Solomon's instances with up to 100 customers and presents a genetic algorithm to solve the problem. Several other heuristics have been proposed and evaluated on the same instances: a parallel simulated annealing algorithm in [Wang et al. \(2015\)](#), a Tabu Search algorithm in [Shi et al. \(2018\)](#), an Adaptive Large Neighborhood Search with Path Relinking in [Hof and Schneider \(2019\)](#), and a lexicographic-based two-stage algorithm in [Shi et al. \(2020\)](#).

Positioning regarding the MTRP literature

The multi-trip Vehicle Routing Problem (MTRP) was introduced in [Fleischmann \(1990\)](#), which argues on the necessity for vehicles to perform multiple trips when demands are high compared to vehicle capacities and travel times are short. As explained in the survey on multi-trip vehicle routing problems by [Cattaruzza et al. \(2016\)](#), the interest for these problems in the literature grew in the last years, essentially due to the development of new city logistic distribution systems and the usage of small eco-friendly vehicles.

In the literature, different naming have been used to refer to the use of multiple trips. In this chapter, we use the terms trip and route. A trip is a sequence of customer location visits starting and ending at the depot and without any intermediate stop at the depot. A route is a succession of one or several trips processed by a single vehicle.

Despite the gain of interest for the use of multiple trips, publications on exact methods remain limited. [Koc and Karaoglan \(2011\)](#) propose a branch-and-cut algorithm, while [Mingozzi et al. \(2013\)](#) describe an exact algorithm for the MTRP based on two set-partitioning formulations and column-and-cut generation procedures for the linear relaxation. Hernandez et al. developed two branch-and-price algorithms: for the MTRP with time windows ([Hernandez et al., 2016](#)) and for the MTRP with time windows and limited trip duration ([Hernandez et al., 2014](#)). [Hernandez et al. \(2016\)](#) also compare route-based and trip-based formulations *i.e.*, formulations where columns define complete routes or single trips. Clearly, the latter limits the size of the search space in the pricing problem because the search is limited to trips instead of sequence of trips; but conversely, it complicates the master problem in which the selected trips have to be able to combine into a feasible set of routes. Computational experiments show a slight advantage for the trip-based formulation on some instances, at the price of a much more complex implementation. In this work, we adopt the route-based model. More recently, [Paradiso et al. \(2020\)](#) report a seven-step solution framework that allows solving efficiently different variants of the MTRP. The framework relies on the combination of many ideas: the concept of structure (roughly speaking, a structure is a trip whose departure time is free), the generation of a relevant set of structure thanks to column generation and thanks to different relaxations, a branch-and-cut algorithm to optimally combine structures, ng-routes, subset-row inequalities, among others.

Positioning regarding the DARP and the non-emergency transportation literature

The literature on non-emergency transport of patients is relatively poor, the emergency case having attracted more attention. [Fogue et al. \(2016\)](#) studie a problem that shares many similarities with ours. The study focuses on the inconveniences of patients with wheelchairs or on stretchers. It deals with several health facilities and the goal is to minimize both the average waiting time of patients and the ambulance usage. The authors proposed a two-phase heuristic: first, a genetic algorithm assigns requests to drivers, and second, a scheduling algorithm builds driver routes. The authors tested the algorithm on different scenarios obtained from a real ambulance company. The experiments show that the solution provided by the proposed algorithm outperforms human experts, reducing the average waiting time of patients and increasing the ambulance usage.

[Lim et al. \(2017\)](#) studie a non-emergency ambulance service met by Hong Kong public hospitals. The objective is threefold and hierarchical: first, maximizing the number of requests served; second, minimizing the total cost and third, balancing the workload for the staff. The authors proposed a local search metaheuristic using a variable neighborhood descent procedure. They applied their algorithm to real-world instances. The results show that the algorithm outperforms the human experts, improving the number of requests served.

As previously stated, non-emergency transportation of patients can be defined as a DARP. Indeed, a DARP is a pickup and delivery problem with time windows for transportation of people with extra client convenience constraints. The standard definition of the DARP ([Cordeau and Laporte, 2003](#)) states that, the ride time of a user must not exceed a maximal duration L and routes must respect a maximum route duration. The MTMPD-RD distinguishes from this definition in three points. First, we propose a formulation with a release date at the pickup node and a due date at the delivery node instead of classic time windows and a constraint to limit ride time of a user. Second, in the MTMPD-RD route duration is not limited. Finally, the main specificity of the MTMPD-RD is that all requests are linked with the depot, either the pickup or the delivery node is the depot and multi-trip is allowed. Several variants of the DARP, however, show some similarities with our problem.

[Parragh et al. \(2012\)](#) include staff/accompanying persons which have restrictions on availability and working time. This enforces to come back regularly at the depot where are located the staff members. Other constraints can enforce to come back regularly at the depot, like lunch breaks or vehicles disinfection ([Zhang et al., 2015](#); [Liu et al., 2015](#)). In these cases, multi-trip is allowed but requests are still characterized by a pickup and a delivery node.

[Liu et al. \(2015\)](#) point out that in real-world applications, several requests might have the depot as the pickup or delivery location. They adapted their model by allowing the pickup or drop-off operations on these requests with large time windows to be grouped together which reduces the size of the graph and, the service time at the new node is equal to the sum of the concerned clients' service time.

[Qu and Bard \(2013\)](#) present a problem motivated by the route planning for Program of All-Inclusive Care for the Elderly (PACE) organisation. It consists in transporting elderly people between their home and activity centers for socializing, or to go to a medical appointment. The main specificity of the problem is that patients have different mobility type, so they might require a simple seat, or more space for a walker or a wheelchair. Hence PACE owns different types of vehicles whose interior can be modified. The problem is defined as a heterogeneous Pickup and Delivery Problem with configurable vehicle capacity. The paper presents a Mixed Integer Linear Program (MILP) formulation of the problem and an Adaptive Large Neighborhood Search algorithm. The same authors extended their work in [Qu and Bard \(2015\)](#) providing a branch-and-price-and-cut algorithm. The algorithm is tested on real instances and randomly generated instances and can solve instances with up to 50 customers to optimality. Other DARP variant

problems are solved with exact methods (Cordeau, 2006; Luo et al., 2019).

2.3 Problem description and mathematical formulation

2.3.1 Problem description

The goal of the problem is to optimize the way customers are transported to or from a central depot on a time horizon $[0, T_{max}]$. This depot is denoted v_0 and customer locations v_1 to v_n . When a customer needs to be transported from their location to the depot, we call it an inbound request. Otherwise, when the customer is transported from the depot to their location, it is called an outbound request. We denote by \mathcal{N}^+ the set of inbound requests and by \mathcal{N}^- the set of outbound requests. $\mathcal{N} = \mathcal{N}^+ \cup \mathcal{N}^-$ is the union of these two sets. We also equivalently denote $\mathcal{N} = \{1, \dots, n\}$, where the customer location of request i is v_i . In what follows, terms request, patient and customer will be used indifferently.

With each request $i \in \mathcal{N}$ is associated a demand $q_i \geq 0$, a release date r_i , a due date d_i and a service time s_i . The demand indicates the number of people involved in the request. The release date is the time at which these people are available at the pickup node. The due date is the latest allowed arrival time at the delivery node. The service time is the time needed at the customer location for pickup (inbound request) or drop off (outbound request). A time might also be spent at the depot, for pickups and drop-offs as explained below.

The transportation network can be represented by a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ and $\mathcal{A} = \{(v_i, v_j) | v_i, v_j \in \mathcal{V}, i \neq j\}$. A travel time t_{ij} and a travel cost c_{ij} are defined on each arc $(v_i, v_j) \in \mathcal{A}$. A homogeneous fleet $\mathcal{M} = \{m_1, \dots, m_K\}$ of K vehicles of capacity Q is available to satisfy transportation requests. Vehicle routes consist of one or several successive trips, starting and coming back to the depot. Each trip is allowed to mix inbound and outbound requests. When leaving the depot (starting a new trip), a vehicle can either be empty (the trip will not fulfill any outbound request) or not. In the second case, a constant pickup time T_P is counted, before the vehicle can leave and after all release dates associated with the outbound requests served in the trip are passed. When returning to the depot (ending a trip), a vehicle can again either be empty (the trip did not fulfill any inbound request) or not. In the second case, a constant time T_D is counted for drop-off.

Figure 2.1 illustrates on a simple example the time constraints that have to be satisfied. Signs $+$ and $-$ respectively indicate inbound and outbound requests. The route is made of three trips. In Trip 1, the vehicle has to serve two outbound requests for patients 3 and 6. The vehicle thus leaves the depot not before time $\max(r_3, r_6) + T_P$, when the two patients are ready and picked-up. It first goes to the location of inbound patient 1. If it arrives earlier than r_1 , it waits until the patient is ready (time r_1). Then, a time s_1 is spent for pickup. The vehicle then successively goes to locations v_3 and v_6 for drop-offs. These locations respectively need to be reached before due dates d_3 and d_6 . Once these locations reached, drop-offs times s_3 and s_6 are respectively spent. At worst, the trip continues from v_6 at time $d_6 + s_6$. Then, patient 4 is picked-up (not before r_4 , with service time s_4) and the vehicle returns to the depot with patients 1 and 4 inside. Due dates for these patients impose that the depot is reached before time $\min(d_1, d_4)$. When the depot is reached, the two patients leave the vehicle, for a drop-off time T_D . The vehicle is now empty and ready for a second trip. Trip 2 serves only one outbound request (patient 2), so the vehicle might have to wait for time $r_2 + T_P$ before starting. After having dropped-off patient 2 (before d_2 , with service time s_2), the vehicle returns to the depot empty. The arrival time is only constrained by T_{max} and no drop-off time is incurred. Furthermore, as Trip 3 only contains an inbound request (patient 5), the vehicle can restart immediately. Patient 5 is picked-up (not before r_5 , with service time s_5), the vehicle returns to the depot

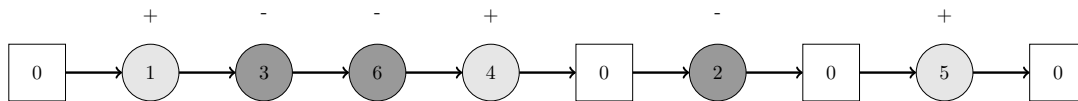


Figure 2.1: Route timing example

before d_5 , a drop-off time T_D is added and the vehicle route is finished, at worst at time T_{max} .

A solution is feasible if it contains at most K routes, serves all requests and is such that all routes are valid, where a route r is valid if:

- r starts and ends at the depot (with one or several trips),
- the vehicle capacity is never exceeded in the route,
- the timing of the route is consistent with release dates, due dates, service times and the time horizon, as illustrated with Figure 2.1.

The objective is to minimize the total travelling cost. Table 2.1 reports the notation introduced in the definition of the MTMPD-RD.

In the following, we assume that instances are not trivially unfeasible. In particular, $r_k + s_k + t_{k0} \leq d_k$ holds for all inbound customers and $r_k + T_P + t_{0k} \leq d_k$ holds for all outbound customers. Furthermore, for inbound customers, we assume that $d_k \leq T_{max} - T_D$ to leave time for drop-off before the end of the time horizon. Finally, we assume that the triangle inequality holds for the travel time and the travel cost matrices.

2.3.2 Mathematical formulation

We now propose a mathematical formulation for the MTMPD-RD. Usually in the literature, 1-M-1 pickup and delivery problems are formulated either using a commodity flow model or a vehicle flow model (Koc et al., 2020). Instead of using these formulations, we adapt the vehicle flow formulation presented in Cattaruzza et al. (2016) for the MTVRP.

In order to represent the trips, we introduce set $\mathcal{H} = \{0, \dots, n-1\}$ to number trips within a vehicle route. In the worst case, a vehicle can use up to n trips. The decision variables are reported in Table 2.2. Note that for time variables τ_i^{kt} and for load variables u_i^{kt} , we introduce a duplicate of the depot, v_{n+1} , to differentiate between the start and the end of the trip.

Using parameters and variables described in Tables 2.1 and 2.2, we propose the following mathematical formulation to model the MTMPD-RD.

$$\min \sum_{(v_i, v_j) \in \mathcal{A}} c_{ij} \sum_{m_k \in \mathcal{M}} \sum_{t \in \mathcal{H}} x_{ij}^{kt} \quad (2.1)$$

Table 2.1: Notation

Sets	
$\mathcal{N} = \{1, \dots, n\}$	set of requests
$\mathcal{V} = \{v_0, v_1, \dots, v_n\}$	vertices of the graph
$\mathcal{A} = \{(v_i, v_j) v_i, v_j \in \mathcal{V}, i \neq j\}$	set of arcs
$\mathcal{G} = (\mathcal{V}, \mathcal{A})$	transportation network
$\mathcal{M} = \{m_1, \dots, m_K\}$	fleet of vehicles
\mathcal{N}^+	set of inbound requests
\mathcal{N}^-	set of outbound requests
Parameters	
n	number of requests
K	number of vehicles
Q	vehicle capacity
q_i	demand of customer i
r_i	release date of customer i
d_i	due date of customer i
t_{ij}	travel time between nodes v_i and v_j
c_{ij}	travel cost between nodes v_i and v_j
s_i	service time at location v_i of customer i
T_P	pickup service time at the depot
T_D	delivery service time at the depot
T_{max}	ending time of the horizon

Table 2.2: Decision variables

Variables	
$x_{ij}^{kt} \in \{0, 1\}$	1 if trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ uses arc $(v_i, v_j) \in \mathcal{A}$, 0 otherwise
$y_i^{kt} \in \{0, 1\}$	1 if trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ serves request $i \in \mathcal{N}$, 0 otherwise
$i^{kt} \in \{0, 1\}$	1 if at least one inbound request is served in trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$, 0 otherwise
$o^{kt} \in \{0, 1\}$	1 if at least one outbound request is served in trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$, 0 otherwise
$\tau_i^{kt} \geq 0$	time at which trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ starts service at node $v_i \in \mathcal{V} \cup \{v_{n+1}\}$
$u_i^{kt} \geq 0$	load of vehicle $m_k \in \mathcal{M}$ on trip $t \in \mathcal{H}$ leaving node $v_i \in \mathcal{V} \cup \{v_{n+1}\}$

subject to

$$\sum_{m_k \in \mathcal{M}} \sum_{t \in \mathcal{H}} y_i^{kt} = 1 \quad \forall i \in \mathcal{N} \quad (2.2)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_i\}} x_{ji}^{kt} = y_i^{kt} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.3)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_i\}} x_{ij}^{kt} = y_i^{kt} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.4)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_0\}} x_{0j}^{kt} \leq 1 \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.5)$$

$$\sum_{v_i \in \mathcal{V} \setminus \{v_0\}} x_{i0}^{kt} \leq 1 \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.6)$$

$$i^{kt} \leq \sum_{i \in \mathcal{N}^+} y_i^{kt} \leq |\mathcal{N}^+| i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.7)$$

$$o^{kt} \leq \sum_{i \in \mathcal{N}^-} y_i^{kt} \leq |\mathcal{N}^-| o^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.8)$$

$$u_0^{kt} = \sum_{i \in \mathcal{N}^-} q_i \times y_i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.9)$$

$$u_j^{kt} \geq u_i^{kt} + q_j - Q(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, j \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.10)$$

$$u_j^{kt} \geq u_i^{kt} - q_j - Q(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, j \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.11)$$

$$u_{n+1}^{kt} = \sum_{i \in \mathcal{N}^+} q_i \times y_i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.12)$$

$$u_i^{kt} \leq Q \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.13)$$

$$\tau_j^{kt} \geq \tau_0^{kt} + T_P \times o^{kt} + t_{0j} - T_{max}(1 - x_{0j}^{kt}) \quad \forall (v_0, v_j) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.14)$$

$$\tau_j^{kt} \geq \tau_i^{kt} + s_i + t_{ij} - T_{max}(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, i \neq 0, j \neq 0, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.15)$$

$$\tau_{n+1}^{kt} \geq \tau_i^{kt} + s_i + t_{i0} - T_{max}(1 - x_{i0}^{kt}) \quad \forall (v_i, v_0) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.16)$$

$$\tau_{n+1}^{kt} \geq \tau_0^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.17)$$

$$\tau_0^{k,t+1} \geq \tau_{n+1}^{kt} + T_D \times i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \setminus \{n-1\} \quad (2.18)$$

$$T_{max} \geq \tau_{n+1}^{k,n-1} + T_D \times i^{k,n-1} \quad \forall m_k \in \mathcal{M} \quad (2.19)$$

$$\tau_i^{kt} \geq r_i \times y_i^{kt} \quad \forall i \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.20)$$

$$\tau_{n+1}^{kt} \leq d_i + T_{max}(1 - y_i^{kt}) \quad \forall i \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.21)$$

$$\tau_0^{kt} \geq r_i \times y_i^{kt} \quad \forall i \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.22)$$

$$\tau_i^{kt} \leq d_i + T_{max}(1 - y_i^{kt}) \quad \forall i \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.23)$$

$$x_{ij}^{kt} \in \{0, 1\} \quad \forall (v_i, v_j) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.24)$$

$$y_i^{kt} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.25)$$

$$i^{kt} \in \{0, 1\} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.26)$$

$$o^{kt} \in \{0, 1\} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.27)$$

$$0 \leq \tau_i^{kt} \leq T_{max} \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.28)$$

$$u_i^{kt} \geq 0 \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (2.29)$$

The objective function (2.1) minimizes the total cost. Constraints (2.2) ensure that each request is assigned to exactly one vehicle and one trip. Constraints (2.3) and (2.4) are the flow conservation constraints. Constraints (2.5) and (2.6) ensure that trips start and end at the depot. Constraints (2.7) (resp., constraints (2.8)) ensure i^{kt} (resp., o^{kt}) equals to 1 when at least one inbound (resp., outbound) request is served in the trip, and equals to 0 otherwise. Constraints (2.9) to (2.13) are flow capacity constraints, where constraints (2.9) ensure that the load of a vehicle starting a trip is equal to the sum of loads of the outbound requests served in the trip, constraints (2.10) and (2.11) ensure vehicle load flow conservation when visiting an inbound (resp., outbound) customer, constraints (2.12) ensure that the load of a vehicle ending a trip is equal to the sum of loads of the inbound requests served in the trip, and constraints (2.13) check that vehicle capacity is never exceeded. Time precedence relations are ensured with constraints (2.14) to (2.19). Constraints (2.14) ensure that the pickup service time T_P is added when leaving the depot if the trip contains at least one outbound request. Constraints (2.15) model travel and service times between two customers while constraints (2.16) model the return to the depot. Constraints (2.17) ensure that the ending time of a trip is later than its starting time. Constraints (2.18) are used to add the delivery service time T_D when returning to the depot if the trip contained at least one inbound request. Constraints (2.19) ensure that the service of a route ends before the end of the time horizon. Release and due date constraints are respected with constraints (2.20) to (2.23). Constraints (2.20) and (2.21) respectively check that an inbound customer is picked up at home after their release date and dropped off at the depot before their due date, while constraints (2.22) and (2.23) check that an outbound customer is picked up at the depot after their release date and dropped off at home before their due date. Finally, constraints (2.24) to (2.29) define the variables.

2.4 Solution method

Solving the compact formulation (2.1)-(2.29) using a standard branch-and-cut solver is not efficient because of the weakness of the lower bound provided by the linear relaxation. In this section, we propose an extended formulation based on a set partitioning formulation (SP) which provides a much stronger lower bound. This formulation is characterized by a large number of decision variables. To cope with this drawback, we use a column generation approach to solve the linear relaxation. This linear relaxation of the extended formulation is called the master problem (MP). The master problem limited to the available set of columns is called the restricted master problem (RMP). After solving the RMP, a subproblem (called pricing problem) is solved in order to find new columns to add to the RMP. These columns correspond to routes with negative reduced costs. In our case, the pricing problem is a multi-trip elementary shortest path problem with resource constraints. It is solved using a specific label correcting algorithm (Feillet et al., 2004). The column generation algorithm ends when the pricing problem does not find any column with a negative reduced cost. Column generation is embedded into a branch-and-price algorithm. Due to the multi-trip aspect of the problem formulation, and because we later compare it with another implementation of branch-and-price, we call the algorithm the multi-trip branch-and-price algorithm (MT-BP).

2.4.1 Master problem

Let Ω be the set of feasible routes, as defined in Section 2.3.1. These routes satisfy capacity and time constraints. Customer locations are visited at most once. Let $a_i^r = 1$ if route $r \in \Omega$ serves request $i \in \mathcal{N}$, 0 otherwise. Let $b_{ij}^r = 1$ if route r uses arc $(v_i, v_j) \in \mathcal{A}$, 0 otherwise. We denote by $c_r = \sum_{(v_i, v_j) \in \mathcal{A}} b_{ij}^r c_{ij}$ the cost of route $r \in \Omega$. We introduce a non-negative integer decision variable θ_r equal to the number of

times route $r \in \Omega$ is selected in the solution. The problem is described with the following set partitioning formulation (SP):

$$\min \sum_{r \in \Omega} c_r \theta_r \quad (2.30)$$

subject to

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (2.31)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (2.32)$$

$$\theta_r \in \mathbb{N} \quad \forall r \in \Omega \quad (2.33)$$

The objective function (2.30) minimizes the total cost. Constraints (2.31) are the set partitioning constraints, they ensure that each request is served. Constraint (2.32) imposes to build at most K routes. The linear relaxation of (2.30)-(2.33) defines the master problem. We call $MP(\Omega_t)$, the master problem, restricted to the subset of routes $\Omega_t \subseteq \Omega$.

2.4.2 Column generation

Column generation consists in finding columns with negative reduced cost and adding them to the restricted master problem using dual variables of the current solution of the restricted master problem. Let $\forall v_i \in \mathcal{V} \setminus \{v_0\}$, λ_i unsigned be the dual variables associated with constraints (2.31) and $\lambda_0 \leq 0$ the dual variable associated with constraint (2.32). The algorithm starts with an initial set of feasible columns Ω_0 to solve the restricted master problem $MP(\Omega_0)$. Then, the pricing problem searches for columns with negative reduced costs and add them to the restricted master problem $MP(\Omega_1)$. This process is repeated until no column with a negative reduced cost is found. The reduced cost of route r is defined as follows:

$$\bar{c}_r = c_r - \left(\sum_{i \in \mathcal{N}} a_i^r \lambda_i \right) - \lambda_0 = \sum_{(v_i, v_j) \in \mathcal{A}, v_j \neq v_0} b_{ij}^r (c_{ij} - \lambda_j) + \left(\sum_{(v_i, v_0) \in \mathcal{A}} b_{i0}^r c_{i0} \right) - \lambda_0 \quad (2.34)$$

Note that dual variable λ_0 should not be subtracted every time the depot is visited.

Algorithm 2.1 describes the column generation scheme. $MP(\Omega_t)$ corresponds to solving the master problem restricted to Ω_t and is used to determine the solution S and the dual variable vector λ . Pricing(λ) corresponds to the call to the algorithm that solves the pricing problem with dual variables λ . It returns a set of routes with negative reduced costs. The pricing problem as well as an associated exact solution approach are described in detail in Section 2.5.

2.4.3 Branching rules

Before describing the branching strategy, let us introduce some definitions. Given a solution S of $MP(\Omega_t)$, we call *flow* on arc $(v_i, v_j) \in \mathcal{A}$, the sum of variables using this arc in S , we denote it as $f_{ij} = \sum_{r \in \Omega} b_{ij}^r \theta_r$. A well-known branching strategy for VRPs consists in looking for an arc with a fractional flow and branch on it. When all arcs have an integer flow, the solution is feasible for the integer problem. In our problem, due to the use of multiple trips, it may happen that a solution is non-feasible for the integer problem but,

Algorithm 2.1 Column generation()

```

1:  $t \leftarrow 0$ 
2:  $\Omega_0 \leftarrow$  initial set of routes
3: repeat
4:    $S, \lambda \leftarrow MP(\Omega_t)$ 
5:    $R \leftarrow$  Pricing( $\lambda$ )
6:    $\Omega_{t+1} \leftarrow \Omega_t \cup R$ 
7:    $t \leftarrow t + 1$ 
8: until  $R = \emptyset$ 
9: return  $S$ 

```

has integer flows for all arcs. To deal with this issue, we introduce the notion of triplet. We call a *triplet* the succession of three vertices with v_0 as second vertex, *i.e.*, (v_i, v_0, v_j) is a triplet with $1 \leq i, j \leq n$. Triplet (v_i, v_0, v_j) can be seen as the succession of arcs (v_i, v_0) and (v_0, v_j) ; actually, it represents the transition between the ending part of a trip and the starting of the next trip in a route. Similarly to the definition of arc flow, the flow on a triplet (v_i, v_0, v_j) is defined as the sum of variables using this triplet in S , we denote it as $f_{i0j} = \sum_{r \in \Omega} b_{i0j}^r \theta_r$, where $b_{i0j}^r = 1$ if subsequence (v_i, v_0, v_j) exists in route r , 0 otherwise.

We show that any solution with integer flows both on arcs and on triplets is integer. We then explain how we use triplets for branching. But first, let us emphasize that a solution might have integer arc flows on all arcs and some fractional flows on triplets. We illustrate that with an example. Let us consider a solution of the master problem defined as a set of 4 routes covering 5 requests: $r_1 = (v_0 v_1 v_0 v_2 v_0)$ with $\theta_1 = 0.5$, $r_2 = (v_0 v_1 v_0 v_3 v_0 v_4 v_5 v_0)$ with $\theta_2 = 0.5$, $r_3 = (v_0 v_4 v_5 v_0)$ with $\theta_3 = 0.5$, and $r_4 = (v_0 v_2 v_0 v_3 v_0)$ with $\theta_4 = 0.5$. In this solution, $f_{ij} \in \{0, 1\}$ on all arcs $(v_i, v_j) \in \mathcal{A}$, but triplet (v_1, v_0, v_2) , for example, has a fractional flow $f_{102} = 0.5$ (supplied by route r_1).

Proposition 1. *Let S be an optimal solution of MP. S is integer if and only if flows on all arcs and triplets are integer.*

Proof. The fact that all flows on arcs and triplets are integer when a solution is integer is trivial because flows are sums of variables θ_r . We show that having integer flows on arcs and triplets ensures that the solution is integer.

We consider a solution S that is not integer. We show that there exists at least one arc or one triplet for which the flow is fractional. We proceed by contradiction, assuming that the flow is integer for all arcs and triplets. Let r_1 be a route such that $0 < \theta_{r_1} < 1$ and let (v_0, v_u) be the first arc of this route. Let r_2 be another route in the solution (*i.e.*, $\theta_{r_2} > 0$) that also traverses arc (v_0, v_u) . Route r_2 exists because the flow on arc (v_0, v_u) is integer and $0 < \theta_{r_1} < 1$.

We first assume that route r_2 does not start with (v_0, v_u) and call (v_l, v_0) the arc that precedes (v_0, v_u) in r_2 . We know that $f_{l0u} = \sum_{r \in \Omega} b_{l0u}^r \theta_r \geq \theta_{r_2} > 0$ and also that f_{l0u} is integer. Then, $f_{l0u} + \theta_{r_1} > 1$, while $f_{l0u} + \theta_{r_1} \leq \sum_{r \in \Omega} b_{l0u}^r \theta_r \leq \sum_{r \in \Omega} a_u^r \theta_r = 1$, which makes a contradiction.

We now consider the case when r_2 starts with the same arc as r_1 (v_0, v_u) . Let then call v_i the first node after which r_1 and r_2 diverge. Node v_i exists because the two routes start with the same arc and are different. We consider two cases:

- $v_i \in \mathcal{V} \setminus \{v_0\}$. We denote v_{j_1} the node that follows v_i in route r_1 and v_{j_2} the node that follows v_i in route r_2 . We know that: $\sum_{r \in \Omega} a_i^r \theta_r = 1$, $\sum_{r \in \Omega} a_i^r \theta_r \geq \sum_{r \in \Omega} (b_{ij_1}^r + b_{ij_2}^r) \theta_r = f_{ij_1} + f_{ij_2}$, $f_{ij_1} > 0$ and $f_{ij_2} > 0$. It follows $0 < f_{ij_1} < 1$ and $0 < f_{ij_2} < 1$, which contradicts the initial assumption.

- $v_i = v_0$. We know, by definition, that v_i is not the first vertex of the two routes. We also know that it is not the last for at least one route, say route r_2 , because the two routes are different. So, v_i corresponds to an intermediate stop at the depot for route r_2 . We call v_l the customer before v_i in the two routes and v_u the customer after v_i in route r_2 . We know that $f_{l0u} = \sum_{r \in \Omega} b_{l0u}^r \theta_r \geq \theta_{r_2} > 0$ and also that f_{l0u} is integer. Then, $f_{l0u} + \theta_{r_1} > 1$, while $f_{l0u} + \theta_{r_1} \leq \sum_{r \in \Omega} b_{l0}^r \theta_r \leq \sum_{r \in \Omega} a_l^r \theta_r = 1$, which makes a contradiction.

□

Based on Proposition 1, we propose the following branching scheme. If S is not integer and there exists at least one arc (v_i, v_j) with a fractional flow f_{ij} , then we derive two branches: in the first branch, arc (v_i, v_j) is forbidden in the solution ($f_{ij} = 0$); in the second branch, arc (v_i, v_j) is enforced ($f_{ij} = 1$). To forbid the use of (v_i, v_j) , variables using this arc are removed from the master problem and the arc is removed in the pricing problem. To enforce the use of (v_i, v_j) , variables using arcs (v_i, v_l) with $v_l \neq v_j$ and (v_l, v_j) with $v_l \neq v_i$ are removed from the master problem and these arcs are removed in the pricing problem.

If a solution S is not integer but all arcs have an integer flow, then there exists a fractional flow on a triplet (Proposition 1). Let (v_i, v_0, v_j) be such a triplet. We derive two branches: in the first branch, the triplet (v_i, v_0, v_j) is forbidden in the solution ($f_{i0j} = 0$); in the second branch, the triplet (v_i, v_0, v_j) is enforced ($f_{i0j} = 1$). To forbid the use of (v_i, v_0, v_j) , variables using it are removed from the master problem. In the pricing problem (see Section 2.5), a label that finishes with (v_i, v_0) is never extended to v_j . This constraint has some consequences on the dominance rule as explained in Section 2.5.4. To enforce the use of (v_i, v_0, v_j) , variables using arcs (v_i, v_l) with $v_l \neq v_0$ and (v_l, v_j) with $v_l \neq v_0$ are removed from the master problem. Also, variables using arc (v_i, v_0) not followed by (v_0, v_j) and variables using arc (v_0, v_j) not preceded by (v_i, v_0) are removed from the master problem. In the pricing problem, arcs (v_i, v_l) with $v_l \neq v_0$ and (v_l, v_j) with $v_l \neq v_0$ are removed. Also, during the execution of the labeling algorithm, all triplets (v_i, v_0, v_l) with $v_l \neq v_j$ and (v_l, v_0, v_j) with $v_l \neq v_i$ are forbidden. To forbid a triplet, a label that finishes with the first arc of the triplet cannot be extended using the second arc. Again, it impacts the dominance rule as explained in Section 2.5.4.

2.4.4 Management of infeasible linear programs

At each node of the branch-and-price tree, it may happen that the current set of columns does not allow finding a feasible solution in the RMP. To deal with that issue, we introduce a non-negative decision variable z with a coefficient 1 in the left-hand side of Constraints (2.31). This variable is penalized in the objective function with a large coefficient M_z . With this variable a feasible solution always exists. When a LP is solved, variable z might then be equal to 0 or be positive. If $z = 0$, the algorithm continues normally. If $z > 0$, it is probable that the RMP does not admit any feasible solution (in the sense of the formulation without z). However, we cannot be sure: it might happen that a solution with $z = 0$ exists but that this solution is not as good as the solution with $z > 0$. We propose to deal with that as follows.

At the root node, M_z is set to a large initial value M_z^0 . When the column generation at a given node N terminates with $z > 0$, the node is pruned, but a duplicate node N' is inserted in the pending-nodes queue, with exactly the same characteristics except that $M_z' = \alpha_z M_z$, with $\alpha_z > 1$. With this mechanism, the LP bound will progressively increase when nodes are duplicated, until $z = 0$ or a large-enough value is reached proving that the instance is not feasible. For each instance we compute a feasibility bound FB , if no feasible solution is found and the pending nodes all have a lower bound greater than FB , it proves that the instance is unfeasible.

2.5 Multi-Trip Shortest Path Problem with Resource Constraints

The pricing problem for vehicle routing problems is classically reduced to an Elementary Shortest Path Problem with Resource Constraints (ESPPRC), with ad hoc resource definitions. The ESPPRC is then solved using a labeling algorithm (see for example Feillet et al. (2004)). A labeling algorithm consists in using labels to represent partial routes. Partial routes are obtained by extending initial labels from a source node (the depot) in order to reach a final node (the depot). The labels are obtained using an extension function from one node to its neighbors. The goal is to find feasible routes with negative reduced costs. A dominance rule is used in order to eliminate partial routes that cannot create an optimal route.

Different resource definitions have been proposed in the context of pickup and delivery problems, but these definitions are always supported by a graph in which two nodes are introduced for each request: a pickup node and a drop-off node. When one of these nodes is traversed, a specific discrete resource is updated, to indicate that the associated request is either started or finished. In the MT-BP, we propose a completely different approach. In the graph, we only introduce a node for customer home locations, in addition to the depot. It drastically reduces the size of the graph but induces a complex resource management because we cannot identify easily in a label which requests are ongoing.

The main issues can be summarized as follows. When a label reaches a customer location v_i , it means that the associated request is added to the trip. If the request is inbound, it doesn't raise any special difficulty: the new patient is loaded into the vehicle, the time is updated to take account of the release date and a new deadline is defined to make sure that the due date will be satisfied. Outbound requests are much more difficult to manage. If v_i is the drop-off point of an outbound request, it reveals that the patient is in the vehicle from the beginning of the trip. We introduce a dedicated resource \tilde{q} to manage that issue and efficiently check that adding this patient complies with the vehicle capacity. Furthermore, the starting time of the trip is potentially delayed because of the release date r_i (at which the patient was ready at the depot). In order to manage that second difficulty, we anticipate delays: without deciding which outbound requests will be added to the trip, we generate labels that leave the depot with delayed departures. Then, further delays are not allowed when extending labels: if the starting time of the trip is too early for an outbound request, this request can't be added to the trip. The information on trip starting times is not kept with a new resource: the vector of reachable requests (classically introduced to avoid that the same request is satisfied several times in a route) allows keeping the information on acceptable outbound requests. This vector is, however, somehow difficult to manage because a request could be unreachable in the current trip but could become reachable in the next trip.

In the following we introduce the main components of the labeling algorithm: label definition, label initialization, extension rules and dominance rules.

2.5.1 Label definition

A label L represents a path P starting at depot v_0 in graph \mathcal{G} . It is defined by a vector $L = (v, c, t, l, q, \tilde{q}, R = \{R_1, \dots, R_n\})$ of size $n + 6$, with:

- v : the last node visited in P .
- c : the current reduced cost of P .
- t : the time at which node v is left (service time at node v included).
- l : the latest time allowed to be back to the depot in the current trip.

- q : the load induced by inbound requests in the current trip (including node v).
- \tilde{q} : the maximal load reached up to node v (included) in the current trip.
- R : the reachability vector.

Resource l is introduced to limit the arrival time at the depot and ensure satisfaction of the due date for inbound requests served in the trip. Resources q and \tilde{q} allow managing the capacity constraint in presence of inbound and outbound requests. Resource \tilde{q} is actually enough to check the capacity constraint, but q is needed to update \tilde{q} . Resource vector R is defined with three possible states for each request. Given a request $i \in \mathcal{N}$: (i) $R_i = C$ if v_i is reachable in the current trip (ii) $R_i = N$ if v_i is unreachable in the current trip because of resource constraints but is reachable in the next trip (iii) $R_i = U$ if node v_i cannot be reached anymore in the route because it has already been reached or because resource constraints make it unreachable. Other fields in L are standard.

2.5.2 Label initialization

Usually, labeling algorithms are initialized with a single label that corresponds to the path $P = (v_0)$, ready for extension at time 0. In the MT-BP, we propose to initialize the algorithms with $|\mathcal{N}^-|+1$ labels:

- $L_0 = (v_0, -\lambda_0, 0, T_{max}, 0, 0, R)$ with:
 - $\forall k \in \mathcal{N}^+$: if $\max(r_k, t + t_{0k}) + s_k + t_{k0} \leq d_k$, $R_k = C$; $R_k = U$ otherwise
 - $\forall k \in \mathcal{N}^-$, $R_k = N$
- $L_i = (v_0, -\lambda_0, r_i + T_P, T_{max}, 0, 0, R)$ for $i \in \mathcal{N}^-$ with:
 - $\forall k \in \mathcal{N}^+$: if $\max(r_k, t + t_{0k}) + s_k + t_{k0} \leq d_k$, $R_k = C$; $R_k = U$ otherwise
 - $\forall k \in \mathcal{N}^-$: if $r_k \leq r_i$ and $t + t_{0k} \leq d_k$, $R_k = C$; if $r_k > r_i$ and $t + t_{0k} \leq d_k$, $R_k = N$; $R_k = U$ otherwise

Label L_0 represents a vehicle that would start empty from the depot at time 0. Pickup time T_P is not consumed before departure. Therefore, this vehicle can only serve inbound requests in this trip. Other labels correspond to later departures from the depot. We create one label for each time instant $r_i + T_P$ (label L_i), with $i \in \mathcal{N}^-$. These labels simulate vehicles that would start at time $r_i + T_P$. Patients of outbound requests with a release date not larger than r_i are apt to be in the vehicle, but none is necessarily in. No other labels are needed because there is no interest to start a route at any other time instant.

Resource R indicates which customers might be reached in the trip or in next trips. This condition depends on the deadline d_k for outbound customers. For inbound customers, $R_k = U$ if the deadline cannot be met, otherwise $R_k = C$ or $R_k = N$ depending on condition $r_k \leq r_i$.

2.5.3 Extension rules

In this section, we describe resource extension functions when a label is extended to a customer and when it is extended to the depot. These functions differ between inbound and outbound requests for some resources.

Extension to a customer

A label $L^1 = (v^1, c^1, t^1, l^1, q^1, \tilde{q}^1, R^1)$ attached to a node $v^1 = v_i \in \mathcal{V}$ is extended to a node $v_j \in \mathcal{V} \setminus \{v_0\}$ if and only if v_j is reachable in the current trip, i.e., $R_j^1 = C$. The new label L^2 is given by $L^2 = (v^2, c^2, t^2, l^2, q^2, \tilde{q}^2, R^2)$ such that:

- $v^2 \leftarrow v_j, c^2 \leftarrow c^1 + c_{ij} - \lambda_j$
- if $j \in \mathcal{N}^+$:
 - $t^2 \leftarrow \max(r_j, t^1 + t_{ij}) + s_j$
 - $l^2 \leftarrow \min(l^1, d_j)$
 - $q^2 \leftarrow q^1 + q_j$
 - $\tilde{q}^2 \leftarrow \max(\tilde{q}^1, q^2)$

The service cannot start before the release date (resource t). The deadline is updated if needed (resource l). The load induced by inbound requests is increased (resource q). Resource \tilde{q} is increased if the load q^2 at node v is larger than the highest load up to that node.

- if $j \in \mathcal{N}^-$:
 - $t^2 \leftarrow t^1 + t_{ij} + s_j$
 - $l^2 \leftarrow l^1$
 - $q^2 \leftarrow q^1$
 - $\tilde{q}^2 \leftarrow \tilde{q}^1 + q_j$

Resource t is simply updated. Neither the deadline (resource l) nor resource q are modified. Resource \tilde{q} is increased by q_j because visiting v_j reveals that the patient was in the vehicle from the beginning of the trip.

- $R_j^2 \leftarrow U$
- $\forall k \in \mathcal{N}^+ \setminus \{j\}$:
 - if $R_k^1 = C$ and $\max(r_k, t^2 + t_{jk}) + s_k + t_{k0} \leq \min(l^2, d_k)$, then $R_k^2 \leftarrow C$
 - else if $R_k^1 \in \{C, N\}$, $q^2 = 0$ and $\max(r_k, t^2 + t_{j0} + t_{0k}) + s_k + t_{k0} \leq d_k$, then $R_k^2 \leftarrow N$
 - else if $R_k^1 \in \{C, N\}$, $q^2 > 0$ and $\max(r_k, t^2 + t_{j0} + T_D + t_{0k}) + s_k + t_{k0} \leq d_k$, then $R_k^2 \leftarrow N$
 - else $R_k^2 \leftarrow U$

These conditions straightforwardly take account of the waiting times implied by release dates and the deadlines derived from due dates. Time T_D at the depot is introduced or not depending on the presence of inbound customers in the trip.

- $\forall k \in \mathcal{N}^- \setminus \{j\}$:
 - If $R_k^1 = C$ and $t^2 + t_{jk} \leq d_k$ and $t^2 + t_{jk} + s_k + t_{k0} \leq l^2$, then $R_k^2 \leftarrow C$
 - else if $R_k^1 \in \{C, N\}$ and $q^2 = 0$ and $\max(r_k, t^2 + t_{j0}) + T_P + t_{0k} \leq d_k$, then $R_k^2 \leftarrow N$
 - else if $R_k^1 \in \{C, N\}$ and $q^2 > 0$ and $\max(r_k, t^2 + t_{j0} + T_D) + T_P + t_{0k} \leq d_k$, then $R_k^2 \leftarrow N$

- else $R_k^2 \leftarrow U$

Computationally, the feasibility of label L^2 is checked before computing vector R^2 . The only condition to check is $\tilde{q} \leq Q$. Time constraints are necessarily satisfied because the extension was conditioned by $R_j^1 = C$.

Extension to the depot

Note that contrary to classical implementations of the ESPPRC for vehicle routing problems, labels that have been extended to the depot are still eligible for future extensions. Furthermore, as for the label initialization, several new labels are created, with different starting times. These labels represent the different times at which the vehicle might start its new trip.

A label $L^1 = (v^1, c^1, t^1, l^1, q^1, \tilde{q}^1, R^1)$ attached to a node $v^1 = v_i \in \mathcal{V} \setminus \{v_0\}$ can always be extended to the depot. Indeed, deadlines are checked in L_1 thanks to resources l and R_i (the label could not have reached v_i if a direct return to the depot was not possible). Such an extension means that the current trip is finished and that a new trip can start. We generate one label L_0^2 for which the new trip will start as early as possible, with the vehicle empty, and one label L_j^2 for each departure time $r_j + T_p$ obtained from requests $j \in \mathcal{N}^-$ such that $R_j^1 \in \{C, N\}$.

Label $L_0^2 = (v_0, c^1 + c_{i0}, a_t, T_{max}, 0, 0, R^2)$ where:

- a_t indicates when the vehicle is ready: $a_t = t^1 + t_{i0} + T_D$ if $q^1 > 0$ (a drop-off time is needed for inbound requests), $a_t = t^1 + t_{i0}$ otherwise
- $\forall k \in \mathcal{N}^+$:
 - if $R_k^1 \in \{C, N\}$ and $\max(r_k, a_t + t_{0k}) + s_k + t_{k0} \leq d_k$, then $R_k^2 \leftarrow C$
 - else $R_k^2 \leftarrow U$
- $\forall k \in \mathcal{N}^-$:
 - if $R_k^1 \in \{C, N\}$, then $R_k^2 \leftarrow N$
 - else $R_k^2 \leftarrow U$

Label $L_j^2 = (v_0, c^1 + c_{i0}, t^2 = \max(r_j, a_t) + T_p, T_{max}, 0, 0, R^2)$ where:

- a_t indicates when the vehicle is ready and is defined as above
- $\forall k \in \mathcal{N}^+$:
 - if $R_k^1 \in \{C, N\}$ and $\max(r_k, t^2 + t_{0k}) + s_k + t_{k0} \leq d_k$, then $R_k^2 \leftarrow C$
 - else $R_k^2 \leftarrow U$
- $\forall k \in \mathcal{N}^-$:
 - if $R_k^1 \in \{C, N\}$, and $t^2 + t_{0k} \leq d_k$, and $r_k \leq \max(r_j, a_t)$, then $R_k^2 \leftarrow C$
 - else if $R_k^1 \in \{C, N\}$, and $r_k > \max(r_j, a_t)$, then $R_k^2 \leftarrow N$
 - else $R_k^2 \leftarrow U$

2.5.4 Dominance rule

We introduce the following dominance rule in order to improve the efficiency of the algorithm. By definition, given two labels L^1 and L^2 attached to the same node, L^1 dominates L^2 if, (i) L^1 has a better cost and (ii) any feasible extension of L^2 is also feasible for L^1 (i.e., each resource is less consumed by L^1 than L^2 and, each node is in a better (or equivalent) reachability state in L^1 than in L^2). In the following, the label identifier is added as a superscript to the parameters of the labels. Formally speaking, L^1 dominates L^2 if:

- $v^1 = v^2, c^1 \leq c^2, t^1 \leq t^2, l^1 \geq l^2, q^1 \leq q^2, \tilde{q}^1 \leq \tilde{q}^2$,
- for every $i \in \mathcal{N}$: if $R_i^2 = C$, then $R_i^1 = C$ and if $R_i^2 = N$, then $R_i^1 \in \{C, N\}$

Note that the dominance rule is not applied in a very specific situation, due to the fact that branching constraints on triplets are non-robust with regard to the pricing problem. Let us consider that a constraint on a triplet (v_i, v_0, v_j) is active at the current branch-and-price node. Let also consider two labels L_1 and L_2 , both attached to the depot and such that L_1 originates from v_i . Then, L_1 is subject to a constraint that does not apply to L_2 and is thus more restricted in its extensions: L_1 cannot dominate L_2 .

2.6 Computational experiments

In this section, we describe the benchmarks of instances used for our experiments and we analyze the computational results.

As mentioned in Section 2.2, the MTMPD-RD was not addressed in the literature. However, the MTMPD-RD has several similarities with the VRP with Simultaneous Pickup and Delivery and Time Windows (VRPSPDTW) for which benchmark instances are available (Wang and Chen, 2012). To allow a comparison on the basis of these instances, we adapted our branch-and-price algorithm to solve the VRPSPDTW.

We also generated a set of instances for the MTMPD-RD. In order to show the effectiveness of our algorithm (MT-BP), we developed a standard branch-and-price algorithm to solve the classical pickup and delivery problem. In the following we coin this algorithm PD-BP. In order to allow a comparison between MT-BP and PD-BP, the generated instances were slightly simplified.

All algorithms are implemented in the C++ programming language and Linear Programs are solved using the commercial solver IBM CPLEX 12.9. Experiments are executed on a computer with a 3.6 GHz CPU and 32 GB of RAM running on Windows 10. All computational times are given in seconds. Unless explicitly indicated, the computational time is limited to 7,200 seconds.

This section is organized in six parts. First, we explain the adaptation of our algorithm (MT-BP) to solve the VRPSPDTW and we evaluate it on the existing benchmark instances. Second, we explain how we generated benchmark instances for the MTMPD-RD. Third, we briefly describe the PD-BP algorithm and compare it to MT-BP on a slightly simplified version of the new benchmark instances. Fourth, we deeply analyze the behavior of MT-BP. Fifth, we study the impact of the size of the fleet of vehicles on the solution cost and on the computing time. Finally, we conduct other sensitivity analyses on challenging instances characterized by a larger interval between release and due dates and/or a larger time horizon.

2.6.1 Comparison with the VRPSPDTW literature

As mentioned previously, the VRPSPDTW has several similarities with the MTMPD-RD except for the few following points. In the VRPSPDTW, all customers require both a delivery (outbound) and a pickup

(inbound) request. When visiting a customer location, a vehicle has to serve both requests, starting with the delivery request. Customer visits are subject to time windows at the customer location, instead of release dates and due dates for the MTMPD-RD. A service time is only considered at customer locations and includes both the time for delivery and for pickup. Finally, multiple trips are not allowed.

In the literature, the VRPSPDTW is usually defined as a lexicographic bi-objective problem. The first objective minimizes the number of vehicles, while the second objective minimizes the total traveling cost. Readers are referred to [Wang and Chen \(2012\)](#) for a formal definition of the VRPSPDTW.

In order to solve the VRPSPDTW, we adapted the MT-BP algorithm as follows:

- In the pricing problem, we do not allow extending labels that have reached the depot. It prevents generating routes with multiple trips. When initializing labels and applying extension rules, all values N (reachable in next trip) of the reachability vector are marked as unreachable U in the reachability vector.
- The VRPSPDTW introduces a strict precedence between the inbound and outbound requests associated with the same customer. Recall that these two requests are considered independently in our model, *i.e.*, two nodes are introduced in \mathcal{V} even if they correspond to the same physical location. To deal with the strict precedence between inbound and outbound requests, we define the inbound node of the customer location as the unique successor of the outbound node, and inversely, the outbound node as the unique predecessor of the inbound node. The distance and the travel time between these two nodes are fixed to zero.
- Time windows at the customer locations are converted into release and due dates as follows. Given a customer v_i in a VRPSPDTW instance with a time window $[a_i, b_i]$ and a service time s_i , we denote i^- (resp., i^+) the associated outbound (resp., inbound) request and we set $r_{i^-} = 0, d_{i^-} = b_i, s_{i^-} = 0, r_{i^+} = a_i, d_{i^+} = T_{max}$ and $s_{i^+} = s_i$. T_{max} is set to the closing time b_0 of the time window associated with the depot. T_P and T_D are set to 0.

For the sake of comparison, we set for each tested instances the number of vehicles equals to the number of vehicles obtained in the best known solution in the literature and our single objective is to minimize the total traveling distance.

To evaluate our algorithm, we use the benchmark instances of [Wang and Chen \(2012\)](#). They consist of three 10-customers instances, three 25-customers instances, three 50-customers instances and fifty-six 100-customers instances. Optimal solutions are only known for the three instances with 10 customers, for one instance with 25 customers and for one instance with 50 customers. These solutions were found by Wang and Chen using a commercial solver. Since 2012, several papers have addressed these instances with different heuristic solution methods. [Shi et al. \(2020\)](#) reports best-known solutions.

Table 2.3 shows our results on the nine small instances (10, 25 and 50 customers). Column “Instance” states the instance name; “#customers” the number of customers; “#vehicles” the number of vehicles used in the best known solution from the literature; “BKS” the value of the best known solution; “opt” the optimality status in the literature; “MT-BP” the value of the solution obtained by MT-BP, and “CPU time” the computing time in seconds of MT-BP.

Table 2.3 shows that, except for instance RCdp5004, we could solve all instances very quickly. Instance RCdp5004 could only be solved to optimality after more than 27 hours. We also conducted complementary

¹A value of 725.59 was mistakenly reported in [Wang and Chen \(2012\)](#); the detail of the solution, provided by the authors, permitted to see that this solution is not feasible

Table 2.3: Comparison with the best-known solutions of the small-size instances for the VRPSPDTW

Instance	#customers	#vehicles	BKS	opt	MT-BP	CPU time
RCdp1001	10	3	348.98	yes	348.98	0.01
RCdp1004	10	2	216.69	yes	216.69	0.01
RCdp1007	10	2	310.81	yes	310.81	0.02
RCdp2501	25	5	551.05	yes	551.05	0.09
RCdp2504	25	4	473.46		473.46	0.25
RCdp2507	25	5	540.87		540.87	0.28
RCdp5001	50	9	994.18	yes	994.18	4.41
RCdp5004	50	6	733.21 ¹		733.21	99234.80
RCdp5007	50	7	809.72		809.72	76.83

experiments by imposing to use one less vehicle. We proved that the problem becomes infeasible in all cases (including instance RCdp5004). We thus closed the benchmark for small instances.

For instances with 100 customers, MT-BP was able to solve to optimality six instances out of the 56. Table 2.4 presents the results for these six instances. For these six instances, we found the same value as the best-known solution in the literature; we also proved for four of them infeasibility when using one less vehicle. Thus we proved the optimality of four best-known solutions. When using one less vehicle, we were unable to solve Rdp106 and Cdp106 or to prove their infeasibility. For the fifty remaining instances, no feasible solution was found within two hours of computing time.

Table 2.4: Comparison with the best-known solutions of the 100-customers instances for the VRPSPDTW

Instance	#vehicles	BKS	MT-BP	CPU time
Rdp101	19	1650.80	1650.80	11.34
Rdp102	17	1486.12	1486.12	76.59
Rdp105	14	1377.11	1377.11	598.54
Rdp106	12	1252.03	1252.03	7117.61
Cdp105	10	1053.12	1053.12	257.114
Cdp106	10	963.45	963.45	4932.52

The benchmark instances of Wang and Chen (2012) are derived from the instances of Solomon (1987). Instances are classified according to two factors:

- Customer distribution: with clustered customers (C), with customer locations generated uniformly randomly over a square (R), with a combination of randomly placed and clustered customers (RC).
- Time window type and vehicle capacity: 1 for narrow time windows and small vehicle capacity, 2 for large time windows and large vehicle capacity.

Small-size instances of Wang and Chen (2012) are all of type RC1, which means a combination of randomly placed and clustered customers and, with narrow time windows and small vehicle capacity. 100-customers

instances contain instances of any type (C1, R1, RC1, C2, R2, RC2) but MT-BP was able to solve only instances of type 1.

Even though we closed the benchmark for some instances, we can conclude that the MT-BP is not very suitable to address VRPSPDTW instances with 100 customers, as it did not find any feasible solution for most of such instances within the considered time limit. In particular, instances involving large time windows were harder to solve, as none of them could be solved. This behaviour is often encountered when using column generation to solve vehicle routing problems. In fact, enlarging time windows increases the number of feasible solutions in the dynamic programming process when solving the subproblems, which is a key element in the effectiveness of the solution method. We also notice that MT-BP is able to solve instances of different types of customer distribution (randomly located (R) and clustered customers (C)).

2.6.2 Generation of instances for the MTMPD-RD

In this section, we describe how we generated a set of realistic instances for the MTMPD-RD. The instances were generated using real data from the city of Aix-en-Provence, France. The depot is the Public Hospital of the city and the demands are unitary: they correspond to the transportation of single patients from or to this hospital. The road network of the city is extracted from OpenStreetMap and represented by a graph denoted \mathcal{G}_C . The travel times are expressed in minutes and rounded up to integer values. The speeds considered are half of the maximal speeds indicated on each road segment in OpenStreetMap, in order to match with the urban context.

To create an instance with n requests, we randomly select n nodes from \mathcal{G}_C . From these n nodes plus the depot node, we compute the travel time matrix with the Dijkstra's shortest path algorithm minimizing traveling time and we construct the complete graph \mathcal{G} and the input data t_{ij} . We set $c_{ij} = t_{ij}$. The time horizon is set to $[0 \text{ min.}, 240 \text{ min.}]$ corresponding to a working period of 4 hours. Pickup and drop-off times at the depot (T_P and T_D) are fixed to 3 minutes, as well as service times at customer locations.

To generate release and due dates, we suppose that an acceptable ride time for a customer is the time needed for a direct ride between pickup and drop-off locations plus 20 minutes. To ensure feasibility for each request i , we randomly generate r_i and d_i as follows:

- if $i \in \mathcal{N}^+$, then $r_i \geq t_{0i}$, $d_i \leq T_{max} - T_D$, and $d_i - r_i = s_i + t_{i0} + 20$.
- if $i \in \mathcal{N}^-$, then $r_i \geq 0$, $d_i \leq T_{max} - s_i - t_{i0}$, and $d_i - r_i = T_P + t_{0i} + 20$.

Four series of instances are generated with 50 or 100 customers and two types of vehicles: sedan ($Q = 2$) and minivan ($Q = 4$). Series of instances are named $S_{n/t}$ where, n is the number of requests (50 or 100) and t is the vehicle type (s for sedan or m for minivan). The fleet size differs from an instance to another and is determined in a way that guarantees both feasibility and a high utilization rate of vehicles (see details below).

Given n and t , series $S_{n/t}$ is composed of 3 subsets of instances corresponding to different distributions of inbound and outbound requests. The probability that a request is inbound is fixed to: 25%, 50% or 75%. Each subset is composed of 5 instances identified from 1 to 5. We then obtain a total of 15 instances per series. Instances are named $I_{n/t/p/i(k)}$, with n the number of customers, t the vehicle type, p the probability that a request is inbound, i the instance id, and k the number of vehicles.

We first generated the set of instances $S_{50/s}$. This set of instances is used to generate $S_{50/m}$ by setting Q to 4. In both sets we determine for each instance I , the minimum number of needed vehicles to ensure feasibility. So, the size of the fleet of vehicles K can differ from one instance to another.

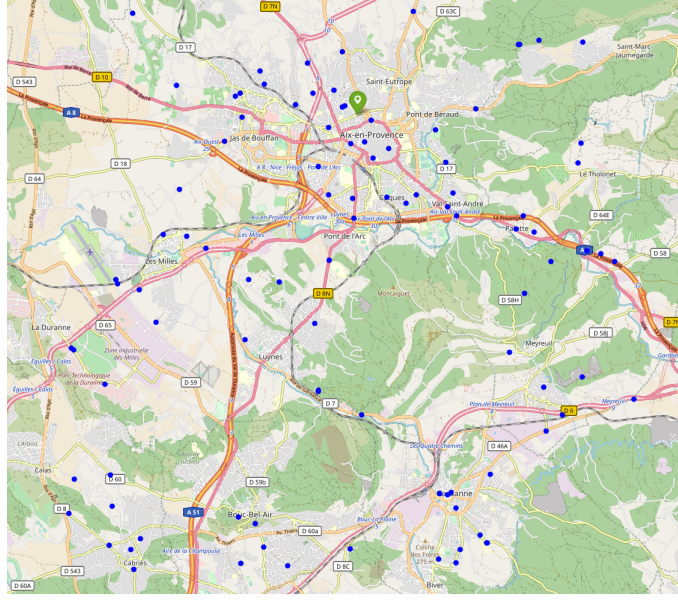


Figure 2.2: Customers distribution of instance $I_{100/s/25/5}(15)$

100-customers instances are generated by combining 50-customers instances. Given a vehicle type t and a probability p that a request is inbound, an instance $I_{100/t/p/i}(k)$ is a combination of instances $I_{50/t/p/i_1}(k_1)$ and $I_{50/t/p/i_2}(k_2)$ where $i = i_1, i_2 = i_1 \pmod{5} + 1$ and $k = k_1 + k_2$ to ensure feasibility.

So, we generate 4 series of instances characterized by the number of requests and the type of vehicles: $S_{50/s}, S_{50/m}, S_{100/s}$ and $S_{100/m}$ with 15 instances each.

Even though patient locations are selected randomly and uniformly among the graph \mathcal{G}_C , their distribution is not uniform in space. The graph \mathcal{G}_C represents a real road network and the denser an area is, the more nodes it contains. Consequently, nodes are more likely to be selected in urban areas than in the countryside. As an illustration, Figure 2.2 shows the customer distribution of instance $I_{100/s/25/5}(15)$ (and of instance $I_{100/m/25/5}(15)$ as only the vehicle capacity and fleet size change). The green flag corresponds to the Public Hospital of Aix-en-Provence, while the small blue points are customer locations. We can see that the distribution mixes clusters and random nodes. The biggest cluster is in the city of Aix-en-Provence, other smaller clusters exist in smallest cities like Gardanne or Bouc-Bel-Air, but also some nodes are more isolated.

2.6.3 Comparison against a pickup and delivery branch-and-price algorithm

As mentioned previously, the MTMPD-RD problem is formulated as a multi-trip vehicle routing problem but it can also be seen as a Pickup and Delivery Problem, with the pickup (resp., delivery) node as the depot of outbound (resp., inbound) requests. In order to transform the MTMPD-RD into a pickup and delivery problem, we duplicate the depot for each request, giving rise to a graph with $2n + 1$ nodes. In order to convert the MTMPD-RD problem into a Pickup and Delivery Problem, we need to simplify the MTMPD-RD by removing the service time at the depot location (*i.e.*, $T_D = T_P = 0$). Indeed, the service time is associated with the depot and cannot be duplicated for each equivalent node when moving to the Pickup and Delivery formulation. With this representation, the MTMPD-RD problem can be solved using a standard branch-and-price for pickup and delivery problems.

To solve the problem, we implemented a branch-and-price algorithm based on [Garaix et al. \(2010\)](#). This algorithm is denoted PD-BP for Pickup and Delivery Branch-and-Price. For a fair comparison, the symmetries induced by the $n + 1$ copies of the depot is broken by removing arcs (v_i, v_j) with $i > j$ where v_i and v_j correspond to the depot.

Tables 2.5 and 2.6 provide experimental results on the generated instances with respectively 50 customers and 100 customers (recall that in the experiments of this section the service time at the depot location is ignored). They show for both solution approaches, the value of the lower bound at the root node (“RLB”), the computing time at the root node (“Time^{RLB}”) in seconds, the objective value of the best integer solution found at the end of the branch-and-price algorithm (“UB”) and the total computing time (“Time”) in seconds. For “RLB” and “UB”, the symbol “-” means that no solution has been found within the allocated computing time. For the computing time, the symbol “-” means that the time limit of two hours was reached. Upper bounds are given in bold when optimality is proven.

The two tables show that MT-BP significantly outperforms PD-BP. Both algorithms found the root lower bound for all instances but, the running time is most of the time better for the MT-BP (for 59 instances out of 60). One might note that this difference grows significantly for the 100-customers instances. This shows that the column generation of MT-BP is much more efficient than the one of PD-BP. PD-BP found a feasible solution for all 50-customers instances and proved optimality for 26 of them, while MT-BP proved optimality for all instances. For the 100-customers instances, PD-BP found a feasible solution for 13 out of 30 instances and proved optimality for 4 of them, while MT-BP found a feasible solution for 22 out of 30 instances and proved optimality for 16 of them. Those figures highlight the efficiency of MT-BP against PD-BP.

2.6.4 Detailed analysis of MT-BP

In this section, we conduct a deep analysis on the behaviour of MT-BP. Tables 2.7 and 2.8 provide results of the MT-BP algorithm on the benchmark instances that we generated. Note that contrary to the instances used in Section 2.6.3, service times T_P and T_D are not ignored. These tables show for each instance, the value of the lower bound (“RLB”) at the root node, the computing time (“Time^{RLB}”) in seconds to reach “RLB”, the objective function value of the best integer solution found (“UB”) and the total computing time (“Time”) in seconds of MT-BP. The symbol “-” for “Time^{RLB}” and “Time” means that the process did not finish within the time limit of two hours. For the other parameters, the symbol “-” means that no value was found within the time limit of two hours. Value “UB” is highlighted in bold if optimality is proven. We also provide the gap at the root node (“root gap”) between RLB and the optimal solution and, the final gap (“final gap”) when MT-BP terminates with a non-proven optimum. The value “root gap” is only provided if optimality is proven. Columns “#trips”, “min”, and “max” respectively provide the total number of trips, the minimum number of trips per vehicle, and the maximum number of trips per vehicle in the obtained solution.

From the two tables, we can notice that the root lower bound is found for all instances and the gap at the root node is very small ; it rarely exceeds 2% and never exceeds 3% (except for instance $I_{50/m/25/4(7)}$ with a gap of 5.09%) which denotes the high quality of the lower bound provided by the set partitioning formulation and the efficiency of the column generation algorithm. When a feasible solution is found but optimality not proven, the final gap is lower than 1% which also shows the high quality of the solutions found within two hours of computing time.

All 50-customers instances are solved optimally and, mostly with a computing time of less than 10 seconds. For 100-customers instances, 21 feasible solutions are found and optimality is proven for 18 of these instances. These figures show that MT-BP is still efficient on instances with up to 100 customers.

Table 2.5: Comparison of the PD-BP and the MT-BP on 50-customers instances

Instance	PD-BP				MT-BP			
	RLB	Time ^{RLB}	UB	Time	RLB	Time ^{RLB}	UB	Time
$I_{50/s/25/1(8)}$	1079.50	1.6	1092	4.9	1079.50	1.8	1092	8.3
$I_{50/s/25/2(7)}$	933.19	4.6	934	5.0	933.19	2.4	934	3.5
$I_{50/s/25/3(7)}$	957.00	3.9	959	12.1	957.00	2.7	959	11.5
$I_{50/s/25/4(8)}$	903.00	3.0	911	155.4	903.00	2.3	911	8.4
$I_{50/s/25/5(7)}$	859.50	3.1	865	9.7	859.50	2.7	865	9.4
$I_{50/s/50/1(7)}$	882.33	2.5	886	-	882.33	1.2	886	6.1
$I_{50/s/50/2(8)}$	769.79	4.2	771	4.5	769.79	0.9	771	1.3
$I_{50/s/50/3(9)}$	996.50	1.4	997	1.6	996.50	1.0	997	1.3
$I_{50/s/50/4(7)}$	917.50	2.1	928	3.9	917.50	1.7	928	3.2
$I_{50/s/50/5(7)}$	849.50	2.7	855	3.5	849.50	1.7	855	3.4
$I_{50/s/75/1(7)}$	942.25	1.7	948	2.7	942.25	1.1	948	2.5
$I_{50/s/75/2(7)}$	878.00	2.8	878	2.8	878.00	1.0	878	1.0
$I_{50/s/75/3(7)}$	903.00	1.6	913	10.3	903.00	1.2	913	11.9
$I_{50/s/75/4(7)}$	922.00	1.6	933	8.5	922.00	0.7	933	4.8
$I_{50/s/75/5(8)}$	995.00	4.0	999	-	995.00	0.6	999	1.5
$I_{50/m/25/1(8)}$	1021.50	1.8	1031	3.9	1021.50	1.6	1031	3.4
$I_{50/m/25/2(7)}$	824.27	5.9	829	10.0	824.27	2.7	829	9.7
$I_{50/m/25/3(7)}$	906.50	7.2	914	47.9	906.50	2.8	914	8.7
$I_{50/m/25/4(7)}$	874.50	4.7	886	25.7	874.50	3.3	886	82.0
$I_{50/m/25/5(7)}$	811.00	4.5	811	4.5	811.00	2.8	811	3.3
$I_{50/m/50/1(7)}$	868.50	3.1	875	12.8	868.50	1.2	875	7.9
$I_{50/m/50/2(7)}$	736.50	11.9	742	41.8	736.50	1.6	742	4.5
$I_{50/m/50/3(9)}$	969.00	3.1	974	4.4	969.00	1.0	974	1.9
$I_{50/m/50/4(7)}$	902.54	2.9	913	4.5	902.54	1.9	913	3.1
$I_{50/m/50/5(7)}$	774.00	3.7	779	5.1	774.00	1.9	779	3.3
$I_{50/m/75/1(6)}$	888.40	14.0	897	32.6	888.40	1.7	897	3.0
$I_{50/m/75/2(7)}$	840.00	5.2	842	196.5	840.00	0.9	842	1.4
$I_{50/m/75/3(7)}$	862.50	4.0	867	-	862.50	1.2	867	2.9
$I_{50/m/75/4(7)}$	863.00	3.5	871	342.9	863.00	0.7	871	4.4
$I_{50/m/75/5(7)}$	931.83	15.3	951	-	931.83	0.7	949	2.9

Table 2.6: Comparison of the PD-BP and the MT-BP on 100-customers instances

Instance	PD-BP				MT-BP			
	RLB	Time ^{RLB}	UB	Time	RLB	Time ^{RLB}	UB	Time
$I_{100/s/25/1(15)}$	1823.50	94.7	-	-	1823.50	63.0	-	-
$I_{100/s/25/2(14)}$	1710.75	258.8	-	-	1710.75	133.1	1721	-
$I_{100/s/25/3(15)}$	1643.00	176.6	1655	-	1643.00	107.6	-	-
$I_{100/s/25/4(15)}$	1557.25	142.0	1559	151.0	1557.25	97.5	1559	152.0
$I_{100/s/25/5(15)}$	1710.00	88.5	1734	-	1710.00	67.1	1736	-
$I_{100/s/50/1(15)}$	1400.50	139.8	-	-	1400.50	32.2	-	-
$I_{100/s/50/2(17)}$	1546.00	39.7	1564	-	1546.00	34.3	1566	-
$I_{100/s/50/3(16)}$	1687.83	64.2	1692	-	1687.83	44.6	1692	1822.4
$I_{100/s/50/4(14)}$	1552.83	95.1	-	-	1552.83	53.6	1575	-
$I_{100/s/50/5(14)}$	1510.53	89.4	-	-	1510.53	39.8	1519	535.1
$I_{100/s/75/1(14)}$	1673.00	74.0	1675	77.7	1673.00	26.0	1675	35.4
$I_{100/s/75/2(14)}$	1581.33	156.3	-	-	1581.33	32.2	1589	919.7
$I_{100/s/75/3(14)}$	1632.00	108.4	-	-	1632.00	24.5	-	-
$I_{100/s/75/4(15)}$	1750.50	72.5	1759	356.1	1750.50	16.1	1759	58.1
$I_{100/s/75/5(15)}$	1756.50	89.5	-	-	1756.50	23.2	-	-
$I_{100/m/25/1(15)}$	1560.00	302.6	-	-	1560.00	60.8	1573	651.3
$I_{100/m/25/2(14)}$	1471.97	419.5	-	-	1471.97	98.1	1480	-
$I_{100/m/25/3(14)}$	1481.92	387.7	1491	-	1481.92	102.3	1490	1758.4
$I_{100/m/25/4(14)}$	1422.67	232.2	1426	263.9	1422.67	102.1	1426	244.3
$I_{100/m/25/5(15)}$	1525.43	319.0	-	-	1525.43	55.7	-	-
$I_{100/m/50/1(14)}$	1291.75	545.9	1309	-	1291.75	44.5	1306	-
$I_{100/m/50/2(16)}$	1401.46	312.3	-	-	1401.46	39.8	-	-
$I_{100/m/50/3(16)}$	1559.43	402.2	-	-	1559.43	39.0	1575	1612.5
$I_{100/m/50/4(14)}$	1445.94	168.1	1456	-	1445.94	52.9	1456	216.6
$I_{100/m/50/5(14)}$	1419.43	300.4	1425	-	1419.43	45.3	1425	79.9
$I_{100/m/75/1(13)}$	1483.50	481.6	-	-	1483.50	24.4	1497	502.0
$I_{100/m/75/2(14)}$	1458.20	548.3	-	-	1458.20	29.4	1471	515.5
$I_{100/m/75/3(14)}$	1486.50	211.9	1491	-	1486.50	26.2	1491	53.2
$I_{100/m/75/4(14)}$	1547.33	625.0	-	-	1547.33	18.0	1554	117.9
$I_{100/m/75/5(13)}$	1477.00	590.9	-	-	1477.00	24.6	-	-

One might notice that the algorithm is more efficient with minivans (13 feasible solutions with 11 optimal solutions) than sedans (8 feasible solutions with 7 optimal solutions). As we can notice from the last three columns of Table 2.8, using sedan vehicles requires more trips than using minivans because of the smaller vehicle capacity. It may explain why the algorithm is more efficient with minivans. From these instances, the impact of the distribution of inbound and outbound requests does not seem to have an impact on the computing time or the quality of obtained solutions.

Table 2.9 summarizes computational results of MT-BP for the four series of instances $S_{n/t}$ with $n \in \{50, 100\}$ and $t \in \{s, m\}$ (15 instances per set). The reported statistics are: the number of feasible solutions (“#Feas.”), the number of optimal solutions (“#Opt.”), the average number of trips per vehicle (“#trips/veh.”), the average number of requests per trip (“#req/trip”), and the average number of requests per vehicle (“#req/veh.”). Note that “#trips/veh.”, “#req/trip” and “#req/veh.” are computed only if a feasible solution is obtained.

Table 2.9 highlights the efficiency of MT-BP both on 50-customers instances and 100-customers instances. Note that theoretically, the maximum number of requests served in a trip can reach twice the vehicle capacity Q (Q outbound requests and Q inbound requests). For sedans ($Q = 2$), we can notice that the average filling rate of vehicles is not far from the maximal theoretical capacity. However, this average filling rate is relatively low for minivans ($Q = 4$). This behaviour can be explained by the value of the time horizon (4 hours) that makes the number of requests per hour relatively small. This is also due to the ride time that limits the detours to pickup or drop off of other patients. We can also notice that the number of trips per vehicle varies between 2 and 3. Note finally that the average number of requests per vehicle #req/veh is equal to #trips/veh \times #req/trip.

2.6.5 Impact of the fleet size

In this section, we study the impact of the fleet size on the total cost and on the computing time. One objective is also to evaluate the opportunity to address this problem by first considering an infinite fleet of vehicles and second assigning the obtained trips to vehicles. This approach should simplify the solution process. Indeed, when the fleet is infinite, the set of feasible routes Ω can equivalently be reduced to the set of mono-trip routes: a multi-trip route can be considered as several mono-trip routes performed by several vehicles. To deal with the case of an infinite fleet, we thus implemented a mono-trip variant of MT-BP (following what is already explained in Section 2.6.1).

In order to perform this study, we selected four instances, one from each series of instances and run MT-BP with different fleet sizes. The selected instances are $I_{50/s/25/3(7)}$, $I_{50/m/25/4(7)}$, $I_{100/s/25/4(15)}$ and, $I_{100/m/25/5(15)}$. Note that for each instance and each fleet size MT-BP is able to prove either optimality or infeasibility. Obtained results are summarized in Tables 2.10, 2.11, 2.12 and, 2.13.

Tables 2.10, 2.11, 2.12 and, 2.13 contain one row per fleet size. Columns show for each fleet size: the solution cost (“cost”), the running time in seconds (“Time”), the total number of trips in the solution (“#trips”), the average number of trips per vehicle (“#trips/veh.”), the average number of requests per trip (“#req/trip”), and the average number of requests per vehicle (“#req/veh.”). The tables report the results with an unlimited number of vehicles and a single trip per vehicle (denoted ∞), then all fleet sizes $K \in \{K_1, \dots, K_2\}$ where K_1 is the minimum fleet size required to obtain the minimal cost equivalent to an infinite size fleet, and K_2 is the maximum fleet size which enforces unfeasibility.

Table 2.10 shows that the best cost is 995 and is achievable with 10 vehicles. Then, reducing the fleet size first slightly increases the cost (from 995 to 1002 between 10 and 8 vehicles), then highly increases to 1054 with 7 vehicles. The instance is infeasible for $K \leq 6$. We notice that reducing the fleet size also

Table 2.7: Results of the MT-BP algorithm on 50-customers instances

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{50/s/25/1(8)}$	1129.53	1.2	1146	6.6	1.44	0.00	23	2	4
$I_{50/s/25/2(7)}$	976.17	2.5	988	7.3	1.20	0.00	21	2	5
$I_{50/s/25/3(7)}$	1027.78	2.8	1054	21.6	2.49	0.00	23	3	4
$I_{50/s/25/4(8)}$	949.75	2.0	956	9.4	0.65	0.00	24	2	4
$I_{50/s/25/5(7)}$	889.83	2.7	902	16.1	1.35	0.00	23	2	5
$I_{50/s/50/1(7)}$	903.50	1.4	910	10.3	0.71	0.00	21	2	4
$I_{50/s/50/2(8)}$	771.00	0.9	771	0.9	0.00	0.00	18	1	4
$I_{50/s/50/3(9)}$	1036.00	1.0	1047	2.6	1.05	0.00	22	1	4
$I_{50/s/50/4(7)}$	964.00	1.5	976	5.6	1.23	0.00	20	2	4
$I_{50/s/50/5(7)}$	881.50	1.8	884	2.7	0.28	0.00	21	2	4
$I_{50/s/75/1(7)}$	948.00	1.1	952	2.0	0.42	0.00	23	2	4
$I_{50/s/75/2(7)}$	879.00	1.1	879	1.2	0.00	0.00	19	2	4
$I_{50/s/75/3(7)}$	905.50	1.4	915	9.3	1.04	0.00	23	2	5
$I_{50/s/75/4(7)}$	943.71	0.8	967	21.2	2.41	0.00	22	2	5
$I_{50/s/75/5(8)}$	1026.63	0.8	1040	11.6	1.29	0.00	24	2	4
$I_{50/m/25/1(8)}$	1065.00	1.5	1069	2.2	0.37	0.00	19	2	4
$I_{50/m/25/2(7)}$	892.50	2.1	908	21.7	1.71	0.00	18	2	4
$I_{50/m/25/3(7)}$	968.38	2.7	993	45.4	2.48	0.00	21	2	4
$I_{50/m/25/4(7)}$	962.43	3.0	1014	4.5	5.09	0.00	23	2	4
$I_{50/m/25/5(7)}$	857.02	2.6	864	6.0	0.81	0.00	19	2	3
$I_{50/m/50/1(7)}$	874.00	1.2	877	1.8	0.34	0.00	20	2	4
$I_{50/m/50/2(7)}$	743.33	1.4	745	2.8	0.22	0.00	16	1	3
$I_{50/m/50/3(9)}$	1015.00	0.8	1026	1.7	1.07	0.00	19	2	3
$I_{50/m/50/4(7)}$	949.00	1.6	962	6.1	1.35	0.00	19	2	4
$I_{50/m/50/5(7)}$	809.00	1.6	809	1.6	0.00	0.00	18	2	3
$I_{50/m/75/1(6)}$	924.75	1.7	935	4.5	1.10	0.00	23	3	5
$I_{50/m/75/2(7)}$	851.25	1.1	857	2.1	0.67	0.00	18	2	4
$I_{50/m/75/3(7)}$	863.00	1.3	867	2.8	0.46	0.00	20	2	4
$I_{50/m/75/4(7)}$	869.00	0.9	879	2.0	1.14	0.00	19	1	4
$I_{50/m/75/5(7)}$	971.50	1.0	995	7.7	2.36	0.00	22	3	4

Table 2.8: Results of the MT-BP algorithm on 100-customers instances

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{100/s/25/1(15)}$	1868.24	48.1	-	-	-	-	-	-	-
$I_{100/s/25/2(14)}$	1746.50	86.2	-	-	-	-	-	-	-
$I_{100/s/25/3(15)}$	1712.00	75.5	1718	289.2	0.35	0.00	41	1	4
$I_{100/s/25/4(15)}$	1584.67	68.7	1591	142.9	0.40	0.00	39	1	5
$I_{100/s/25/5(15)}$	1753.50	47.3	1757	95.1	0.20	0.00	38	2	4
$I_{100/s/50/1(15)}$	1413.75	26.3	-	-	-	-	-	-	-
$I_{100/s/50/2(17)}$	1575.22	24.8	-	-	-	-	-	-	-
$I_{100/s/50/3(16)}$	1726.90	30.6	1751	-	-	0.70	35	1	3
$I_{100/s/50/4(14)}$	1588.67	36.5	-	-	-	-	-	-	-
$I_{100/s/50/5(14)}$	1537.50	29.4	1547	205.2	0.61	0.00	34	1	3
$I_{100/s/75/1(14)}$	1688.83	22.6	1693	507.5	0.25	0.00	40	2	4
$I_{100/s/75/2(14)}$	1595.88	33.8	1606	898.4	0.63	0.00	40	1	4
$I_{100/s/75/3(14)}$	1637.50	21.3	-	-	-	-	-	-	-
$I_{100/s/75/4(15)}$	1755.00	14.9	1766	4780.3	0.62	0.00	42	1	6
$I_{100/s/75/5(15)}$	1762.75	22.2	-	-	-	-	-	-	-
$I_{100/m/25/1(15)}$	1646.29	50.2	1658	1104.0	0.71	0.00	34	1	3
$I_{100/m/25/2(14)}$	1582.06	72.0	-	-	-	-	-	-	-
$I_{100/m/25/3(14)}$	1569.31	77.4	1582	1526.5	0.80	0.00	35	1	3
$I_{100/m/25/4(14)}$	1486.92	79.2	1497	971.6	0.67	0.00	34	1	4
$I_{100/m/25/5(15)}$	1599.61	50.9	1612	1077.3	0.77	0.00	34	1	4
$I_{100/m/50/1(14)}$	1307.82	33.9	1327	-	-	0.59	28	1	3
$I_{100/m/50/2(16)}$	1451.83	32.3	1472	-	-	0.20	28	1	3
$I_{100/m/50/3(16)}$	1605.81	30.1	1614	117.5	0.51	0.00	30	1	3
$I_{100/m/50/4(14)}$	1495.67	35.9	1505	117.1	0.62	0.00	29	1	3
$I_{100/m/50/5(14)}$	1481.67	32.3	1484	48.4	0.16	0.00	31	1	4
$I_{100/m/75/1(13)}$	1494.17	22.2	1506	123.3	0.79	0.00	32	1	4
$I_{100/m/75/2(14)}$	1470.99	27.7	1485	377.3	0.94	0.00	35	1	4
$I_{100/m/75/3(14)}$	1490.00	26.2	1500	786.3	0.67	0.00	30	1	4
$I_{100/m/75/4(14)}$	1549.28	16.1	1561	264.8	0.75	0.00	32	1	3
$I_{100/m/75/5(13)}$	1490.27	21.5	-	-	-	-	-	-	-

Table 2.9: Summary of obtained solutions with MT-BP

Instance set	#feas.	#opt.	#trips/veh.	#req/trip	#req/veh.
$S_{50/s}$	15	15	2.95	2.29	6.76
$S_{50/m}$	15	15	2.74	2.55	7.00
$S_{100/s}$	8	7	2.62	2.59	6.78
$S_{100/m}$	13	11	2.20	3.16	6.95

Table 2.10: Impact of fleet size for instance $I_{50/s/25/3}$

Fleet size (K)	cost	Time	#trips	#trips/veh.	#req/trip	#req/veh.
∞	995	0.1	22	1.00	2.27	2.27
10	995	1.7	22	2.20	2.27	5.00
9	996	2.1	22	2.44	2.27	5.56
8	1002	4.2	22	2.75	2.27	6.25
7	1054	30.7	23	3.29	2.17	7.14
6	\emptyset	4.3	-	-	-	-

Table 2.11: Impact of fleet size for instance $I_{50/m/25/4}$

Fleet size (K)	cost	Time	#trips	#trips/veh.	#req/trip	#req/veh.
∞	920	0.4	22	1.00	2.27	2.27
10	920	4.2	22	2.20	2.27	5.00
9	922	9.6	22	2.44	2.27	5.56
8	932	66.9	22	2.75	2.27	6.25
7	1014	7	23	3.29	2.17	7.14
6	\emptyset	4.2	-	-	-	-

Table 2.12: Impact of fleet size for instance $I_{100/s/25/4}$

Fleet size (K)	cost	Time	#trips	#trips/veh.	#req/trip	#req/veh.
∞	1591	7.7	39	1.00	2.56	2.56
15	1591	142.7	39	2.60	2.56	6.67
14	1595	218.7	39	2.79	2.56	7.14
13	1607	418.1	39	3.00	2.56	7.69
12	1642	473.4	42	3.25	2.38	8.33
11	\emptyset	151.5	-	-	-	-

Table 2.13: Impact of fleet size for instance $I_{100/m/25/5}$

Fleet size (K)	cost	Time	#trips	#trips/veh.	#req/trip	#req/veh.
∞	1606	16.8	33	1.00	3.03	3.03
17	1606	156.8	33	1.94	3.03	5.88
16	1607	238.1	33	2.06	3.03	6.25
15	1612	1075.5	34	2.27	2.94	6.67
14	1617	844.2	34	2.43	2.94	7.14
13	1631	401.8	34	2.62	2.94	7.69
12	1728	1779.1	38	3.17	2.63	8.33
11	\emptyset	82.0	-	-	-	-

impacts the computing time. An interesting remark is that the number of trips is not drastically impacted by the fleet size.

Results on tables 2.11, 2.12 and, 2.13 are similar. They show a slight impact on the cost when the fleet size is near K_1 and a high impact when the fleet size is close to K_2 , especially for instance $I_{100/m/25/5}$ where the cost increases from 1631 to 1728 when passing from 13 to 12 vehicles.

As expected, the computing time is significantly lower when the fleet is infinite and that the mono-trip variant of MT-BP is used. This approach, combined with a post-assignment of trips to vehicles, is clearly to be preferred when the fleet is large enough. However, the tables show its limits. Indeed, it would never give a solution with less than K_1 vehicles. Our results suggest that reducing the fleet size by a few vehicles by allowing multiple trips does not have a high impact on transportation costs. Furthermore, if the number of available vehicles is a critical resource for the company providing the transportation service, the use of multiple trips can be necessary to satisfy all requests. It is also important to note that K_1 and K_2 are not known in advance and depend on the instance characteristics. Hence, it is difficult to be sure in advance that the solution provided by the mono-trip variant would comply with the fleet size.

2.6.6 Other sensitivity analyses

In this section, we present other sensitivity analyses. We generated new challenging instances for the MTMPD-RD based on instances presented in Section 2.6.2 with modified parameters to assess the limits of the MT-BP algorithm.

Let us recall that, for all instances presented in Section 2.6.2, the time horizon is $[0 \text{ min.}, 240 \text{ min.}]$ which corresponds to a working period of 4 hours, and a ride is considered acceptable for a customer if its duration is the time needed for a direct ride between the pickup and drop-off locations plus 20 minutes (*i.e.*, 20 minutes corresponds to the maximal acceptable detour).

In this section we consider a larger time horizon and a larger acceptable ride time to evaluate how the MT-BP algorithm scales up. We propose three new sets of instances:

- Larger acceptable ride time: allowing detours of 40 minutes and keeping a time horizon of 240 minutes.
- Larger time horizon: $[0 \text{ min.}, 480 \text{ min.}]$ (8 hours) and keeping acceptable detours of 20 minutes.
- Both a larger time horizon $[0 \text{ min.}, 480 \text{ min.}]$ and larger acceptable detours (40 minutes).

Let E_0 be the initial set of instances presented in Section 2.6.2. E_0 is composed of 60 instances: 30 instances with 50 requests and 30 instances with 100 requests. The three new sets are derived from E_0 , especially customers remain the same, and are defined as follows:

- E_1 : instances of E_0 with modified release and due dates to extend the maximal detour from 20 to 40 minutes: $r_i \leftarrow r_i - 10$ and $d_i \leftarrow d_i + 10$; however, if the new values are not consistent with the conditions defined in Section 2.6.2, we rather set $r_i \leftarrow r_i$ and $d_i \leftarrow d_i + 20$, or $r_i \leftarrow r_i - 20$ and $d_i \leftarrow d_i$, when the new release date is too small or the new due date too large, respectively.
- E_2 : instances of E_0 with the time horizon enlarged from $[0 \text{ min.}, 240 \text{ min.}]$ to $[0 \text{ min.}, 480 \text{ min.}]$ and the release and due dates regenerated.
- E_3 : instances of E_2 with release and due dates modified similarly to instance set E_1 .

Fleet sizes are determined the same way as they were for instances of set E_0 . We run the MT-BP algorithm on all these new instances. Similarly to Table 2.9 for instances of set E_0 , tables 2.14, 2.15 and 2.16 summarize computational results for instances of sets E_1 , E_2 and, E_3 .

Table 2.14: Summary of obtained solutions with MT-BP for the E_1 instances

Instance set	#feas.	#opt.	#trips/veh.	#req/trip	#req/veh.
$S_{50/s}$	15	13	3.16	2.58	8.15
$S_{50/m}$	15	14	2.64	3.64	9.62
$S_{100/s}$	0	0	-	-	-
$S_{100/m}$	5	2	2.13	4.50	9.62

Table 2.15: Summary of obtained solutions with MT-BP for the E_2 instances

Instance set	#feas.	#opt.	#trips/veh.	#req/trip	#req/veh.
$S_{50/s}$	15	15	5.29	2.15	11.36
$S_{50/m}$	15	15	4.69	2.25	10.56
$S_{100/s}$	10	5	4.44	2.53	11.24
$S_{100/m}$	13	10	3.77	2.80	10.57

Table 2.16: Summary of obtained solutions with MT-BP for the E_3 instances

Instance set	#feas.	#opt.	#trips/veh.	#req/trip	#req/veh.
$S_{50/s}$	15	14	5.45	2.50	13.64
$S_{50/m}$	15	13	3.96	3.32	13.16
$S_{100/s}$	2	1	4.77	3.23	15.38
$S_{100/m}$	2	1	3.19	3.92	12.50

Tables 2.14 and 2.16 show the limits of MT-BP to solve instances with large maximal detour allowed. Both for E_1 and E_3 , optimality could not be proven for three 50-customers instances. For 100-customers instances, MT-BP found feasible solutions for 5 and 4 for instances respectively in E_1 and E_3 , and was able to prove optimality for 2 instances in both sets. Similarly to the issue of large time windows for the VRPSPDTW, enlarging the maximal detour in instances of the MTMPD-RD increases the number of feasible solutions in the dynamic programming process when solving the subproblems and drastically impacts the effectiveness of the solution method. Table 2.15 shows results for E_2 comparable to those of E_0 (see Table 2.9) with all 50-customers instances solved to optimality and 23 feasible solutions found (of which 15 are proven optimal) for 100-customers instances. These results highlight that the time horizon doesn't seem to have an important impact on efficiency of the MT-BP algorithm. However, increasing the maximal detour highly deteriorates its efficiency. Appendix A provides detailed results on instances of sets E_1 , E_2 and E_3 .

2.7 Conclusion

In this chapter, we introduced a new problem arising from the context of non-emergency transportation of patients. The main originality of the studied problem is that there are two types of requests, a patient is either picked-up at home and transported to the hospital or the other way around. Requests are thus always connected to the hospital, and routes combine both types of requests. In addition, the quality of service is preserved with riding time constraints that enforce the vehicles to regularly come back to the hospital. Consequently, the problem can be addressed either as a special Pickup and Delivery Problem or as a special multi-trip VRP. We call it the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD).

The main goal of the chapter was then to evaluate if tackling the problem as a multi-trip VRP should be preferred. We proposed a branch-and-price algorithm (MT-BP). The complex time and capacity constraints of the problem make the definition and the management of resources complex in the pricing problem. Also, they implied some difficulties in the branching scheme.

As the MTMPD-RD is a new problem, we generated a new set of realistic instances. We also implemented a branch-and-price algorithm based on the pickup and delivery formulation (PD-BP). Experiments showed that the MT-BP outperforms the PD-BP and prove the efficiency of a multi-trip model for this problem. MT-BP proved its efficiency by solving to optimally instances involving up to 100 requests.

To further validate the MT-BP, we adapted it for the solution of the VRPSPDTW. Our experiments on benchmark instances closed the benchmark for small instances and closed four 100-customers instances. We also detected that an infeasible solution was published in the literature. In addition, we evaluated the opportunity to tackle the problem as a mono-trip vehicle routing problem followed by a trip-to-vehicle assignment. Experiments show the computational interest but also the limits of this standpoint. Finally, we conducted other sensitivity analyses generating new challenging instances with a larger time horizon and/or a larger acceptable ride time. Results show that the computational time of the MT-BP algorithm is strongly impacted when the maximal detour is increased, but remains relatively stable for larger time horizons.

A first perspective for this work would be to improve the solution method, by adding valid cuts or using a bidirectional approach for the pricing problem. Also, one might propose heuristics to solve the MTMPD-RD to be able to solve larger instances in a short amount of time, which would be appreciated in real-life applications. An important real-life constraint is also missing in this study. Results show a strong imbalance in the duration of routes. It is inappropriate in our context because the transportation

is subcontracted to independent paramedics that should be considered equitably. In addition, the easiness of patient management may vary depending on their pathology or their lodging accessibility, which also potentially generates inequity among drivers. An interesting perspective would thus be to integrate equity into the problem with both regards.

Workload equity in vehicle routing with a medium-term perspective

In chapter 2, we introduced the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD), a routing problem motivated by the non-emergency transportation of patients. In this chapter, we study the multi-period version of the problem. We consider a longer horizon (several days) for which every day, the MTMPD-RD is solved and routes are assigned to drivers. The contribution of this chapter is to propose techniques to integrate equity between drivers over the time horizon.

In the vehicle routing literature, equity for drivers is mostly considered on a daily basis. In this chapter, we extend the literature by addressing equity on a longer term. We consider a time horizon of a few weeks and a test-bed routing problem motivated by healthcare logistics, with customer demand progressively revealed. Having unbalanced routes at some periods (days) is accepted, but some degree of equity is imposed on the complete horizon. We propose five solution frameworks with different levels of control on equity. A dedicated branch-and-price algorithm is used to solve the daily routing problem and is adapted for the different solution frameworks. Numerical experiments are conducted on a benchmark of realistic instances to evaluate how addressing equity this way impacts the routing cost, equity and feasibility.

Contents

3.1	Introduction	55
3.2	Literature review	56
3.3	Problem description	59
3.3.1	Data discovery and decision process	60
3.3.2	Daily routing problem	60
3.3.3	Assignment problem	61
3.3.4	Equity metric and function	62
3.4	Solution frameworks	62
3.4.1	Best Equity Assignment strategy	63
3.4.2	Solution frameworks	64

3.4.3	Route planning and driver assignment with perfect information	65
3.5	Solution of the daily routing problem	66
3.5.1	Branch-and-price algorithm for the MTMPD-RD	67
3.5.2	Adaptation of the branch-and-price algorithm for driver-independent equity constraints	68
3.5.3	Adaptation of the branch-and-price algorithm for driver-dependent equity constraints	68
3.6	Computational experiments	70
3.6.1	Generation of instances	70
3.6.2	Feasibility	71
3.6.3	Trade-off between equity and cost	72
3.6.4	Daily equity and evolution along the time horizon	75
3.6.5	Computing time	77
3.6.6	Synthesis: main findings	78
3.7	Conclusion	78

3.1 Introduction

Starting from the Truck Dispatching Problem introduced by [Dantzig and Ramser \(1959\)](#) more than 50 years ago, the vehicle routing problem literature has grown considerably with two main objectives: developing methods that can efficiently solve these complex optimization problems (and ultimately problems of similar nature) and close the gap between academic contributions and actual practitioner needs. Following the latter trend, equity has gained a lot of interest in the last two decades, as acknowledged by the recent survey of [Matl et al. \(2018\)](#). Workload equity is indeed especially relevant in many practical situations, where drivers should be assigned distribution routes with comparable workloads.

The most common way of addressing equity in the routing context is to introduce an equity metric, and to balance vehicle routes according to this metric when computing the routing plan. This approach is, however, somehow myopic because, most often, routing plans are computed on a daily basis while the perception that drivers may have of inequity is on a longer term: In practice, a driver may indeed accept unbalanced routes at some periods (days), as long as equity is preserved on a longer range. The purpose of this paper is to answer two important related research questions. To the best of our knowledge, these questions have almost never been investigated yet:

1. Limits of the “myopic” approach? The approach found in the literature is susceptible to be overly restrictive and costly because it imposes balanced routes at every time period (day). Based on a test-bed problem selected in the field of healthcare logistics and a time horizon of a few weeks, we conduct computational experiments that evaluate the impact of this “myopic” approach against approaches considering equity with a more global perspective.
2. Alternative approaches? Considering equity with a global perspective, we introduce and compare five solution frameworks that aim at providing equitable routing plans in the medium-term with limited cost increases.

Several important modeling decisions need to be taken to allow addressing these two questions. A first decision is to define how customer demand is revealed. As far as we know, most authors that have considered equity in a multi-period setting assume that customer demand is known for the whole horizon at the beginning of the planning period (see [Section 3.2](#)). While considering equity in a static multi-period context makes sense for sure, we rather consider that customer demand is progressively revealed. Then, the routing plan computed at any period is computed without any information on what will happen on the next periods. This assumption fits better to our application case and, also, to important applications of vehicle routing like last-mile delivery.

A second important decision is to define how to integrate equity. By nature, a vehicle routing in which equity is introduced becomes multi-objective. When dealing with a dynamic vehicle routing problem, the solution cannot be to compute a set of efficient solutions. Instead, it is simply at the end of the time horizon that we evaluate the proposed solution schemes according to relevant KPIs (Key Performance Indicators). The way equity is introduced (or not) when computing each daily routing plan is specific to the methods.

A third, and last, decision is to define which type of equity is desired. In the literature, one can distinguish two types of equity metrics ([Matl et al., 2018](#)): constant-sum metrics, in which the sum is constant over all solutions (*e.g.*, number of customers served in the route), and variable-sum metrics, whose sum depends on the solution (*e.g.*, traveled distance). The type of metric can have important theoretical implications but also directly impacts solution schemes. In this chapter, we consider a constant-sum metric. The generalization to variable-sum metrics is left to future research.

The test-bed problem that supports this study concerns non-emergency healthcare logistics. We consider a hospital and patients with a medical appointment, that have either to be picked up at home and transported to the hospital or the other way around. This problem is addressed on a time horizon that consists of a few weeks, with an equity metric related to the painfulness of transportation requests. In [Agius et al. \(2022\)](#), we studied the mono-period version of the problem without considering equity. We developed a dedicated branch-and-price algorithm. The solution frameworks proposed in this chapter sequentially solve daily routing problems and are built upon this branch-and-price algorithm.

The contributions of this paper are threefold. First, we introduce a new dynamic multi-period routing problem involving equity for drivers motivated by the non-emergency transportation of patients. Second, we propose different solution frameworks to ensure equity globally on the time horizon and adapt the branch-and-price algorithm developed in [Agius et al. \(2022\)](#) to these frameworks. Third, we conduct extensive numerical experiments on a benchmark of realistic instances to evaluate how addressing equity with this medium-term perspective allows limiting its impact on routing costs. With these contributions, we believe that we provide important insights to the two aforementioned research questions.

The paper is organized as follows. Section 3.2 reviews the related literature. Section 3.3 details the test-bed problem. Section 3.4 describes the different solution frameworks that we developed to take care of equity with a medium-term perspective. Section 3.5 briefly recalls the branch-and-price algorithm developed in [Agius et al. \(2022\)](#) and explains how it can be adapted to the solution of the daily routing problems encountered in the different frameworks. Section 3.6 reports and analyzes computational results. Finally, Section 3.7 concludes the paper, summarizing our findings and suggesting possible extensions to this work.

3.2 Literature review

In this section, we position our work in the literature. Firstly, we briefly review equity in the Vehicle Routing Problem (VRP) literature. Second, we focus on routing problems involving multiple periods and show the novelty of addressing equity with a medium-term perspective. Finally, we quickly survey how equity has been considered so far in the medical transportation literature.

Equity in vehicle routing

The interest raised by equity in an optimization process is far from being limited to vehicle routing. Among others, [Karsu and Morton \(2015\)](#) point out this interest and survey the associated literature in operations research. More recently, [Matl et al. \(2018\)](#) reviewed this literature with a focus on vehicle routing. They show an abundant number of recent papers, that they review and categorize. They also analyze the properties of different equity measures, completing this analysis by a numerical study on small instances solved optimally.

[Matl et al. \(2018\)](#) formalize some important concepts that we will reuse in this work. The equity metric is the workload resource that has to be balanced fairly between drivers. A metric is said constant-sum if for any feasible solution, the sum of workload assigned to drivers remains constant (*e.g.*, number of customers served, total customer load served, total service time). Otherwise, a metric is said variable-sum if the sum of workload assigned to drivers differs between solutions (*e.g.*, routing cost/time). When equity is considered, the problem is bi-objective. The first objective is the classical routing cost minimization and the second objective is the equity objective, usually defined as an explicit objective using an equity function (*ex:* min-max, range). The bi-objective problem can then be addressed as a multi-objective problem,

thus seeking a Pareto front, or simplified to a mono-objective problem, using a weighted sum of the two objectives or limiting inequity with a constraint.

[Halvorsen-Weare and Savelsbergh \(2016\)](#) also studied in-depth the effects of different equity functions in a bi-objective VRP. They conclude that the equity function has significant effects on the Pareto front and suggest that the range function (difference between the drivers with the least and the most workload) is a robust choice. [Matl et al. \(2018\)](#) agree on the former but contradict the latter. Indeed, when optimizing equity with the range function on a variable-sum metric, two undesirable consequences occur. A Pareto-optimal solution can contain non TSP-optimal routes or a Pareto-optimal solution can be workload inconsistent, that is, can contain routes whose workloads are all equal or greater than those of other Pareto-optimal solutions.

[Matl et al. \(2019\)](#) extend the work of [Matl et al. \(2018\)](#) studying the impact of the equity metric on the solution. Numerical experiments underline that the trade-off structure of a solution depends primarily on the workload resource, and less on the equity function.

A clear conclusion that can be drawn from these works is that the literature mainly focuses on single period routing problems. The possibility that the unbalances generated during a period are compensated in previous or future periods is not considered. This fact suffers from a few exceptions that are surveyed in the next paragraph.

Equity in multi-period vehicle routing

To the best of our knowledge [Ribeiro and Ramalhinho Dias Lourenço \(2001\)](#) were the first to consider equity in a multi-period routing problem. In the context of commercial distribution, they proposed a Periodic Vehicle Routing Problem (PVRP) with three objectives, minimizing the total cost, balancing the workload fairly among drivers and minimizing inconsistency to improve customer service. The equity objective is non-linear and the authors were able to solve (exactly) a single instance with 3 periods, 2 drivers and at most 5 customers per period. [Blakeley et al. \(2003\)](#) are also interested in equity for a PVRP, in the context of maintenance operation planning. The objective function is a weighted sum including routing cost, overtime and equity. The equity objective aims at minimizing the difference between the shortest and the longest routes (range on the route duration). Similarly, [Gulczynski et al. \(2011\)](#) define a new variant of the PVRP, that they call PVRP with balance constraints (PVRP-BC), in which an equity objective is introduced to minimize the difference between the minimal and the maximal numbers of customers served within routes (range on the number of customers served). This objective is combined with the routing cost objective into a weighted sum.

More recently, [Nekooghadirli et al. \(2022\)](#) have explicitly questioned the fact that equity is generally defined at the scale of a single period. They introduce a multi-period Capacitated Vehicle Routing Problem (CVRP) that they solve with a two-phase approach. First, the routing problem is solved independently at each period with the VRPSolver framework from [Pessoa et al. \(2020\)](#). Second, routes are assigned to drivers. The objective of the assignment problem is to minimize the maximum workload (routing cost) accumulated on the horizon over all drivers. From their model, the authors determine a lower bound which permits to evaluate how far a solution is from the best possible workload equity. Experiments show that when the number of days is larger than or equal to 5, their approach often reaches the “perfect” workload equity, *i.e.*, the obtained workload balance matches the theoretical lower bound, otherwise the gap is below 1%.

All the above approaches consider equity in a multi-period but static context: All customer demands are known in advance. Basically, they introduce equity in a more difficult problem, a PVRP instead of a single-period VRP, but the paradigm remains the same: A static vehicle routing problem is solved with

an equity objective. In our study, on the opposite, we are interested in the solution of multi-period vehicle routing problems in which the data for new periods is progressively revealed. We thus move to the paradigm of dynamic VRPs. Readers are referred to Pillac et al. (2013) and Bektaş et al. (2014) for literature reviews on these problems. We are only aware of three papers considering equity within this paradigm.

Wen et al. (2010) address a dynamic multi-period vehicle routing problem where customer orders are revealed over the time horizon. Customer orders must be fulfilled within a set of consecutive periods which can start as early as the period after the order is placed. The objective is to minimize a weighted sum function that considers total travel costs, customers waiting and the daily workload balance. The daily workload is the total travel time. In their work, the daily workload balance is defined by the absolute value of the difference between that day's workload and the estimated average daily workload. The authors propose a three phase heuristic to solve the problem. The daily problem is solved using a Variable Neighborhood Search heuristic. Although this work deals with a dynamic multi-period vehicle routing problem, it is very different from our work. First, workload balance is based on variable-sum metric while in our work we deal with a constant-sum metric. Second, only one solution approach is proposed and is based on a Variable Neighborhood Search heuristic for each daily routing problem. In our paper, we propose several approaches based on an exact solution approach for each daily routing problem.

Khorsi et al. (2020) study a problem arising in humanitarian relief logistics in post-disaster situations. They consider realistic aspects such as multiple depots, split delivery and multiple trips. Their problem is dynamic as information is revealed gradually over a time horizon of several periods. A rolling horizon approach is used to manage the dynamics of the problem. Three objectives are taken into account, minimization of the total amount of unsatisfied demands, minimization of the total traveling time and the equity objective. Equity is considered between the demand points to ensure a fair distribution of goods and avoid unfair allocation policies. The equity objective is measured as the minimization of the sum of the absolute deviation of a fraction of unsatisfied demands between the demand points on the rolling horizon. The authors propose an epsilon constraint approach in order to deal with the multi-objective nature of their problem. The proposed approach is based on a commercial solver to solve small size instances (9 locations). Contrary to our paper, this study is very specific to humanitarian relief logistics and deals with a multi-objective problem. In addition, the small size daily problems are solved using a standard solver.

Wang et al. (2022) study a multi-period workload balancing problem arising from the last-mile urban delivery. The problem is defined as dynamic and stochastic and is formulated as a Markov Decision Process. This work is the one that comes closest to our study. They considered two types of workload: Incentive workload, that is, the delivery quantity which affects a courier's income, and effort workload, that is, the delivery time which affects a courier's health. Incentive workload is balanced over the whole time horizon (e.g., a week or a month) whereas effort workload is balanced every day. They propose five dispatching policies, namely, *balanced penalty policy*, *VRP policy with rebalancing*, *myopic policy with rebalancing*, *fixed territory policy*, and *fixed territory with rotation policy*. The *balanced penalty policy* penalizes the cumulative incentive imbalance at each decision period. The authors propose a hybrid simulation optimization algorithm to find the best policy parameter values. In the *VRP policy with rebalancing*, they first solve a VRP then they rebalance the cumulative incentive workload incentive when assigning routes to drivers. In the *myopic policy with rebalancing*, they first solve a myopic problem to minimize the single-period effort imbalance then they rebalance the cumulative incentive workload when assigning routes to drivers. In the *fixed territory policy*, each driver is assigned to a region. The *fixed territory policy with rotation* is similar to the *fixed territory policy* except that the assignment of drivers to regions can be modified every payroll cycle. They showed that the *balanced penalty policy* and the *myopic policy with rebalancing* perform very well compared to other policies. In this chapter, the authors address a stochas-

tic problem and propose a heuristic approach at each step of the proposed approaches. In our study, we study a deterministic and dynamic vehicle routing problem where requests are known only at the start of the associated period. At each step of the proposed methods, we develop an exact approach to solve the associated combinatorial optimization problem. Our study is complementary to this paper and may open several research perspectives for the stochastic version of our problem.

Equity in medical transportation

To the best of our knowledge, no paper in the literature tackles a dynamic multi-period vehicle routing problem with equity in the context of healthcare logistics as we do. However, a few papers are interested in equity in the context of home healthcare logistics.

[Liu et al. \(2013\)](#) define a weekly home healthcare logistics problem in which equity is the primary objective and consists in minimizing the longest route (min-max) in the week to ensure reasonable working hours.

[Triki et al. \(2014\)](#) define a problem in which equity concerns the total daily working time and not drivers. Each day a target working time is defined and the equity objective is to minimize the gap between the actual working time and the target working time for each day. This objective is combined with the routing cost in a weighted sum.

[Cappanera and Scutellà \(2015\)](#) study the home care services problem, a problem with scheduling, assignment and routing decisions. They propose several extensions and particularly study two equity objectives: i) maxmin, which maximizes the minimum operator utilization factor, and ii) minmax, which minimizes the maximum operator utilization factor. Results on computational experiments show that the maxmin criterion finds more balanced solutions (the difference between the maximum and the minimum operator utilization factors is smaller) but the minmax criterion has less impact on the operating costs (it finds solutions with smaller total traveled time).

[Liu et al. \(2021\)](#) consider routing cost, satisfaction of patients and satisfaction of caregivers. The latter is measured as the range of the accumulated working time (routing and service time) assigned to caregivers. Also, daily constraints limit the daily working time of caregivers. A sensitivity analysis shows that choosing an objective function involving only one sub-objective causes a high level of dissatisfaction for caregivers or patients.

3.3 Problem description

[Ulmer et al. \(2020\)](#) point out the lack of consistency in modeling stochastic dynamic vehicle routing problems in the literature and present a modeling framework which is a generalization of conventional Markov decision processes (MDPs). They call it route-based MDP. Despite the dynamic nature of our problem, this framework is not useful in our case, because routing decisions do not involve any dynamics. We iteratively solve static daily routing problems. When new information is revealed it concerns the next period and does not infer with any ongoing optimization process.

We describe our problem in four successive blocks. At first, we describe the dynamic of the problem: how data is revealed and what is the decision process. Then, we describe the daily routing problem that has to be solved at each step of the process. The third block concerns the assignment of routes to drivers. Indeed, contrary to what is usually found in vehicle routing, deciding which driver will execute which route is important once the routing plan is computed. Finally, we precise the equity metric and function.

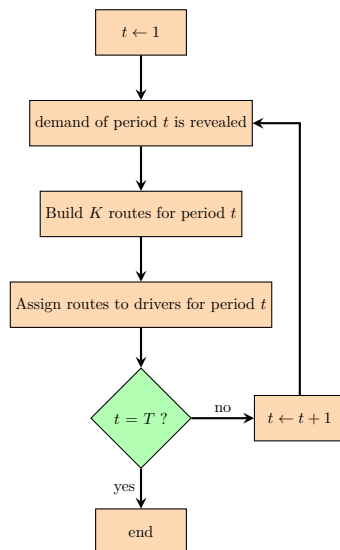


Figure 3.1: Data discovery and decision process

3.3.1 Data discovery and decision process

Our problem is defined on a planning horizon $\mathcal{T} = \{1, \dots, T\}$ of T periods. A set $\mathcal{M} = \{m_1, \dots, m_K\}$ of K drivers is available. We assume that drivers work at every period with identical vehicles of capacity Q . In what follows, terms request, patient and customer will be used indifferently, and the same for the terms period and day.

The problem is dynamic in the sense that requests are revealed day by day. The requests of the following days are unknown. However, each daily routing problem is static as requests of a day t are all known before starting the day. Each day, two decisions have to be made, routing and assignment of routes to drivers. They can be treated separately because vehicles are identical and drivers work every day. Figure 3.1 describes the data discovery and decision process. Iteratively for each day, the demand is revealed, the vehicle routes are computed and the routes are assigned to drivers, until the last period is done.

The problem is bi-objective. The first objective is to minimize the total travel cost over all vehicles and all periods. The second objective concerns equity and is detailed in Section 3.3.4. However, the dynamic of the decision process imposes that a single solution is returned, providing all routing and assignments decisions for all periods. The quality of this solution should then be evaluated according to these two objectives.

In the next two subsections, we describe under which constraints are taken the two types of decisions at each period: routing and assignment. We do not detail the objective functions that guide these decisions. Indeed, this is part of the solution process and is explained in Section 3.4. Also, the way equity is considered may imply limiting the search space. Again, it is related to the solution process and is detailed in Section 3.4.

3.3.2 Daily routing problem

In this section, we describe the problem that needs to be addressed to compute daily routes. Note that to be rigorous, each data, set and variable related to customers should be indexed by the period. However to

lighten the notation and as daily problems are solved separately, we remove the period index.

The problem consists in transporting patients between their home and hospital for appointments. We consider a single hospital and a single depot located at the hospital (v_0). We distinguish two types of requests: inbound and outbound requests. Inbound requests consist in transporting patients from their home to the hospital, while outbound requests consist in transporting patients from the hospital to their home. We denote by \mathcal{N}^+ (resp., \mathcal{N}^-) the set of inbound (resp., outbound) requests and $\mathcal{N} = \mathcal{N}^+ \cup \mathcal{N}^-$. We also equivalently denote $\mathcal{N} = \{1, \dots, n\}$, where the customer location of request i is v_i . The transportation network can then be represented by a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ and $\mathcal{A} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}, i \neq j\}$. A travel time t_{ij} and a travel cost c_{ij} are defined on each arc $(v_i, v_j) \in \mathcal{A}$. In what follows, we set $c_{ij} = t_{ij}$.

With each request $i \in \mathcal{N}$ is associated a demand $q_i \geq 0$, a release date r_i , a due date d_i and a service time s_i . The demand corresponds to the number of people involved in the request. The release and due dates correspond respectively, to the time at which these people are available at the pickup node and, the latest allowed arrival time at the delivery node. The service time is the time needed at the customer location for pickup (inbound request) or drop off (outbound request).

Due to release and due date constraints and vehicle capacity, vehicles are enforced to come back regularly to the depot; hence multiple trips are allowed. Each trip is allowed to mix inbound and outbound requests. When leaving the depot (starting a new trip), a vehicle can either be empty (the trip will not fulfill any outbound request) or not. If the vehicle is not empty, a constant pickup time T_P is counted, before the vehicle can leave and after all release dates associated with the outbound requests served in the trip are passed. When returning to the depot (ending a trip), a vehicle can again either be empty (the trip did not fulfill any inbound request) or not. If the vehicle is not empty, a constant time T_D is counted for drop-off. The daily time horizon is $[0, T_{max}]$. Vehicle routes consist of one or several successive trips, starting and coming back to the depot.

The K routes computed when solving the daily routing problem at a given period should comply with these constraints and serve all customers. With the objective of minimizing the total traveling time and without considering equity, the problem has been introduced in Chapter 2 and called the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD). Section 2.3 provides the mathematical formulation for the MTMPD-RD.

3.3.3 Assignment problem

Given a period t , once a set $\mathcal{R}^t = \{r_1^t, \dots, r_K^t\}$ of routes is available, these routes have to be assigned to drivers. As drivers are supposed identical, routes can indifferently be assigned to any driver without impacting the routing cost. Each driver should be assigned to a route, and each route be assigned to a driver. It can easily be described with a simple mathematical formulation.

Let us introduce decision variable $\gamma_{r,k}^t = 1$ if route $r \in \mathcal{R}^t$ is assigned to driver $m_k \in \mathcal{M}$, 0 otherwise. The assignment problem on period t can be defined with the following constraints:

$$\sum_{r \in \mathcal{R}^t} \gamma_{r,k}^t = 1 \quad \forall m_k \in \mathcal{M} \quad (3.1)$$

$$\sum_{m_k \in \mathcal{M}} \gamma_{r,k}^t = 1 \quad \forall r \in \mathcal{R}^t \quad (3.2)$$

$$\gamma_{r,k}^t \in \{0, 1\} \quad \forall t \in \mathcal{T}, \forall r \in \mathcal{R}^t \quad (3.3)$$

Constraints (3.1) ensure that exactly one route from the routing solution of period t is assigned to each driver, while constraints (3.2) ensure that exactly one driver is assigned to each route. Constraints (3.3) define the variables.

3.3.4 Equity metric and function

As mentioned in Section 3.2, the literature distinguishes two types of equity metrics (Matl et al., 2018). A metric is said constant-sum if for any feasible solution, the sum of workload assigned to drivers remains constant (e.g., number of customers served, total customer load served, total service time). Otherwise, a metric is said variable-sum if the sum of workload assigned to drivers differs between solutions (e.g., routing cost/time).

The most studied metric in the literature is the routing cost which aims to propose equitable routes in terms of cost or time for drivers. In this chapter, we rather consider a constant-sum metric. This is less general, but easier. The case of variable-sum metrics is left for perspectives. Our metric concerns the difficulties that might be encountered by drivers at patient homes when they help patients moving to the vehicle or moving back home. We call it painfulness and denote it π_i^t . For a request i and a period t , π_i^t is assumed to be defined based on factors, like the patient ages, their number, their mobility and the type of residential building where they live. The more painful request i , the higher π_i^t . Discussing how these values are set goes beyond this study. Note that this metric is a generalization of the number of customers served.

The equity function that we use in our equity objective is the range, defined as the difference of painfulness between the drivers with the maximal and the minimal cumulative painfulness over the T periods. This function evaluates equity on the complete time horizon. The possible unbalanced that may be observed at some periods are not considered. However, in our experiments, we will also analyze how the different methods behave with this regards (Section 3.6).

The range function is known to have some undesirable side-effects when combined with a variable-sum metric (Matl et al., 2018). In our context, with a constant-sum metric, these effects are avoided.

3.4 Solution frameworks

An important feature to deal with in the solution frameworks is the bi-objective nature of our problem. Indeed, the main purpose of these frameworks is to achieve the best possible compromise between routing costs and equity. In all our approaches, we follow the same principle. While optimizing daily routes, the objective is always to minimize travel cost. Inequity might be limited or not with a constraint, with techniques depending on the approach (see Section 3.4.2). When assigning routes to drivers, travel costs are known. The objective is always to minimize inequity. Remember that the problem being dynamic, with the decisions made at a period implemented before the next period is started, it is not possible to maintain a Pareto set along the decision process. Instead, at the end of each period, a routing plan and a driver assignment are definitely adopted. So, at the end of the horizon, the solution is a unique sequence of routing and assignment decisions.

In the first part of this section (Section 3.4.1), we present the strategy that we use to assign routes to drivers after each daily routing problem solution. This strategy is shown to be optimal as long as the objective of the assignment is to minimize unbalance. We call it BEA for Best Equity Assignment. Then, in Section 3.4.2, we describe the different frameworks that we introduced to deal with equity in the daily routing optimization. Finally, in Section 3.4.3, we complete the section with a solution scheme assuming

perfect information of customer demand since the beginning of the first period. This scheme is introduced to serve as a baseline.

Before describing the methods, we need to introduce new notation. The total painfulness of period t is denoted $\Pi^t = \sum_{i=1}^n \pi_i^t$. The total cumulative painfulness up to period t is denoted $\Pi^{1 \rightarrow t} = \sum_{t'=1}^t \Pi^{t'}$. The painfulness assigned to a driver m_k on a past period t is Π_k^t . The cumulative painfulness assigned to driver m_k up to period t is denoted $\Pi_k^{1 \rightarrow t} = \sum_{t'=1}^t \Pi_k^{t'}$.

3.4.1 Best Equity Assignment strategy

The Best Equity Assignment strategy (BEA) is applied once vehicle routes are known at any period t . We call \mathcal{R}^t these routes. As drivers are supposed identical and work every day, routes can indifferently be assigned to any driver. We assign the most painful route to the driver with the smallest cumulative painfulness up to period $t-1$, the second most painful route to the driver with second smallest cumulative painfulness, and so on, to balance the cumulative painfulness up to period T . Algorithm 3.1 describes this strategy. Note that this assignment strategy was used in some of the benchmark policies proposed by Wang et al. (2022).

Algorithm 3.1 BEA strategy

- 1: order routes in \mathcal{R}^t by decreasing order of painfulness
 - 2: order drivers in \mathcal{M} by increasing order of cumulative painfulness $\Pi_k^{1 \rightarrow t-1}$
 - 3: **for** $k = 1$ to K **do**
 - 4: assign the k^{th} route of \mathcal{R}^t to the k^{th} driver of \mathcal{M}
-

Proposition 2. *The BEA strategy finds an assignment that is optimal with regards to the cumulative painfulness up to period t (included), given the cumulative painfulness up to period $t-1$ and the set of routes \mathcal{R}^t .*

Proof. Without loss of generality, we assume that sets \mathcal{M} and \mathcal{R}^t are sorted according to the orders defined in the BEA algorithm: increasing value of cumulative painfulness for \mathcal{M} , decreasing value for \mathcal{R}^t . The assignment of a driver d to a route r is denoted (d, r) . The new cumulative painfulness of the driver after this assignment is denoted $\Pi(d, r)$.

We consider an optimal assignment \mathcal{A} . We assume that the first driver, driver m_1 , is not assigned to the first route, route R_1^t . Then, driver m_1 is assigned to a route R_r^t with $r > 1$ and route R_1^t is assigned to a driver m_d with $d > 1$. We show that a new optimal assignment can be obtained by reassigning driver m_1 to route R_1^t and driver m_d to route R_r^t . We call this new assignment \mathcal{A}' .

First, it is trivial that \mathcal{A}' is a valid assignment: All routes are assigned to a driver and all drivers to a route. We show that transforming \mathcal{A} in \mathcal{A}' does not increase the range. Because of the ordering of sets \mathcal{M} and \mathcal{R}^t , we know that $\Pi(1, 1) \geq \Pi(1, r)$ and $\Pi(d, r) \geq \Pi(1, r)$. Also, $\Pi(1, 1) \leq \Pi(d, 1)$ and $\Pi(d, r) \leq \Pi(d, 1)$.

We denote M and m the maximal and minimal driver painfulness after assignment \mathcal{A} . We denote M^- and m^- these values when drivers m_1 and m_d are removed from the driver set. We denote M' and m' these value after assignment \mathcal{A}' .

We have $M' = \max(M^-, \Pi(1, 1), \Pi(d, r)) \leq \max(M^-, \Pi(d, 1)) \leq M$. Similarly, we know that $m' = \min(m^-, \Pi(1, 1), \Pi(d, r)) \geq \min(m^-, \Pi(1, r)) \geq m$. Then, $m \leq m' \leq M' \leq M$, which shows that $M' - m' \leq M - m$.

At this stage, we known that there exists an optimal assignment with driver m_1 assigned to route R_1^t . Considering this assignment and starting from driver m_2 , we can assume that driver m_2 is not assigned to

route R_t^2 (nor route R_1^t that is linked to driver m_1) and repeat the proof mechanism to obtain an optimal assignment with driver m_1 assigned to route R_1^t , and driver m_2 assigned to route R_2^t . Progressively, we obtain an optimal assignment with all drivers assigned to the route of same index, which exactly corresponds to the assignment constructed using the BEA strategy. \square

3.4.2 Solution frameworks

We propose 5 solution frameworks. Remember that the objective function in the daily routing problem is always to minimize routing costs, and that the BEA strategy is always employed for the assignment of routes to drivers. So, the frameworks only differ in the way equity is considered when solving the daily routing problems. Below, we denote p_k^t the decision variable expressing the painfulness allocated to route k when solving the daily routing problem at period t .

Routing-First Equity-Second (RFES)

Vehicle routes are optimized without introducing any constraint on equity in the daily routing problem. The rationale behind this framework is that the BEA strategy might be enough to ensure a balanced share of painfulness at the end of the time horizon. This framework guarantees optimal routing costs.

Single-Period Constant Equity (SPCE)

When solving the daily routing problem at a period t , the average painfulness at that period is computed and routes are only allowed a limited upper deviation from this value. The limit is the same for all routes. The constraints imposed on variables p_k^t are:

$$p_k^t \leq \alpha \times \frac{\Pi^t}{K} \quad \forall m_k \in \mathcal{M} \quad (3.4)$$

where $\frac{\Pi^t}{K}$ is the average painfulness and α a parameter that controls the deviation ($\alpha > 1$). The larger α , the larger the allowed deviation. Note that, while the framework guarantees a limited upper deviation at the end of the time horizon, it does not give any guarantee on the range. However, it is quite clear that limiting the upper deviation also helps minimizing the range.

This framework is equivalent to what is very often used in the literature with “myopic” approaches. Facing the double objective of minimizing routing costs and balancing the workload, a limited deviation to the average workload is imposed. We call this framework the literature baseline.

Multi-Period Constant Equity (MPCE)

This framework extends SPCE by taking account of the painfulness assigned to the drivers at the previous periods. At period t , the constraints imposed on variables p_k^t are:

$$p_k^t \leq \alpha \times \frac{\Pi^{1 \rightarrow t}}{K} - \Pi_k^{1 \rightarrow t-1} \quad \forall m_k \in \mathcal{M} \quad (3.5)$$

$\Pi_k^{1 \rightarrow t-1} + p_k^t$ represents the cumulative painfulness received by the driver, $\frac{\Pi^{1 \rightarrow t}}{K}$ is the average cumulative painfulness. Again, this framework relies on a parameter α , with $\alpha > 1$.

Multi-Period Adaptive Equity (MPAE)

This framework generalizes MPCE. It allows changing the deviation limit at the different periods:

$$p_k^t \leq \alpha_t \times \frac{\Pi^{1 \rightarrow t}}{K} - \Pi_k^{1 \rightarrow t-1} \quad \forall m_k \in \mathcal{M} \quad (3.6)$$

with $(\alpha_1, \dots, \alpha_T)$ a sequence of parameters. It allows being more flexible on the first periods, relying on the next periods to recover equity.

In our implementation of MPAE, sequence $(\alpha_1, \dots, \alpha_T)$ is defined thanks to two parameters: α_s and α_e . α_t then decreases uniformly from α_s at period 1 to α_e at period T :

$$\alpha_t = \alpha_s - (t - 1) \times \frac{\alpha_s - \alpha_e}{T - 1} \quad (3.7)$$

Last-Period Adaptive Equity (LPAE)

This framework is a variant of MPAE with a different setting for sequence $(\alpha_1, \dots, \alpha_T)$. Here, parameters α_1 to α_{T-1} are set to large values, so that the routes are not constrained. Then, a single parameter α is used for the last period: $\alpha_T = \alpha$:

$$(\alpha_1, \dots, \alpha_T) = (+\infty, \dots, +\infty, \alpha) \quad (3.8)$$

In other words, framework RFES is applied for all periods except the last period, where the framework MPCE is applied. As a consequence, the framework guarantees that optimal routing costs are found for all periods except the last one.

3.4.3 Route planning and driver assignment with perfect information

For the purpose of evaluation, we propose another approach that we call Routing First Equity Second with Perfect Information (RFESPI). It consists in solving all daily routing problems first without considering equity, and then assigning routes to drivers for all periods minimizing the range.

In this approach, we assume that all requests of all days are known in advance, which is not compatible with our decision process. So, it is not an acceptable solution framework. However, it can serve as a good baseline for comparison. Indeed, the approach necessarily provides the minimal routing cost and, even if it does not mathematically guarantee a lower bound on the range, it should give a good approximation on the best range that could be achieved. In the Pareto front, it should thus provide a point that is not far from the ideal point.

Given the solutions of all daily routing problems $(\mathcal{R}^1, \dots, \mathcal{R}^T)$, the assignment phase of the RFESPI approach can be modeled and solved using a Mixed Integer Linear Program (MILP). The decision variables are reported in Table 3.1.

The assignment MILP is defined as follows:

$$\text{minimize } \Pi_{max} - \Pi_{min} \quad (3.9)$$

Table 3.1: RFESPI assignment decision variables

Variables	
$\gamma_{r,k}^t \in \{0, 1\}$	equals 1 if route $r \in \mathcal{R}^t$ is assigned to driver $m_k \in \mathcal{M}$ at period t , 0 otherwise
$\Pi_{min} \geq 0$	equals to the minimal cumulative painfulness among the drivers
$\Pi_{max} \geq 0$	equals to the maximal cumulative painfulness among the drivers

subject to

$$\sum_{r \in \mathcal{R}^t} \gamma_{r,k}^t = 1 \quad \forall t \in \mathcal{T}, \forall m_k \in \mathcal{M} \quad (3.10)$$

$$\sum_{m_k \in \mathcal{M}} \gamma_{r,k}^t = 1 \quad \forall t \in \mathcal{T}, \forall r \in \mathcal{R}^t \quad (3.11)$$

$$\sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}^t} \pi_r \times \gamma_{r,k}^t \geq \Pi_{min} \quad \forall m_k \in \mathcal{M} \quad (3.12)$$

$$\sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}^t} \pi_r \times \gamma_{r,k}^t \leq \Pi_{max} \quad \forall m_k \in \mathcal{M} \quad (3.13)$$

$$\gamma_{r,k}^t \in \{0, 1\} \quad \forall t \in \mathcal{T}, \forall m_k \in \mathcal{M}, \forall r \in \mathcal{R}^t \quad (3.14)$$

$$\Pi_{min} \geq 0 \quad (3.15)$$

$$\Pi_{max} \geq 0 \quad (3.16)$$

with π_r the painfulness of route r .

The objective function (3.9) minimizes the range. Constraints (3.10) and (3.11) ensure that for each period, a driver is assigned to exactly one route and a route is assigned to exactly one driver. Constraints (3.12) and (3.13) guarantee that the total cumulative painfulness assigned to a driver is bounded by the minimal and maximal amount of painfulness assigned to drivers. Constraints (3.14) to (3.16) define the variables.

3.5 Solution of the daily routing problem

As seen in Section 3.4, the daily routing problem can be solved with or without equity constraints. When solved without equity constraints, the problem is a MTMPD-RD and is solved using the branch-and-price algorithm provided by Agius et al. (2022). When adding equity constraints, the algorithm needs some adjustments. With the constraints introduced in our solution frameworks, we face two cases: equity constraints in which the equity upper bound (right-hand side of the constraint) is driver-independent and equity constraints in which it is driver-dependent. SPCE belongs to the first category, other frameworks (MPCE, MPAE and LPAE) to the second.

In Section 3.5.1, we briefly describe the branch-and-price algorithm without equity constraints. The purpose is not to give a complete description of the algorithm, readers are referred to Agius et al. (2022) for that. Instead, we briefly present the main components of the algorithm, that are the master problem, the labels defined in the pricing problem and the dominance rule. We do not detail the branching scheme, which is not impacted by equity constraints. Then, in Section 3.5.2 (resp., Section 3.5.3) we explain how the

algorithm is adapted to driver-independent equity constraints (resp., driver-dependent equity constraints). In the section, index t is omitted for the sake of a better readability.

3.5.1 Branch-and-price algorithm for the MTMPD-RD

The MTMPD-RD is formulated as a set partitioning problem (Agius et al., 2022) where Ω denotes the set of feasible routes, c_r is the cost of a route and a_i^r is a binary indicator of requests served in routes:

$$\min \sum_{r \in \Omega} c_r \theta_r \quad (3.17)$$

subject to

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (3.18)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (3.19)$$

$$\theta_r \in \mathbb{N} \quad \forall r \in \Omega \quad (3.20)$$

Decision variable θ_r indicates if a route is selected or not in the solution.

Column generation is used to solve the linear relaxation of this formulation and is embedded into a branch-and-price algorithm. The master problem is obtained from (3.17)-(3.20) by relaxing constraints (3.20). Let λ_i be the unsigned dual variable associated with Constraint (3.18) for $i \in \mathcal{N}$ and $\lambda_0 \leq 0$ be the dual variable associated with Constraint (3.19). The aim of the pricing problem is to find new routes with negative reduced costs. The reduced cost of a route $r \in \Omega$ is:

$$\bar{c}_r = c_r - \left(\sum_{i \in \mathcal{N}} a_i^r \lambda_i \right) - \lambda_0 = \sum_{(v_i, v_j) \in \mathcal{A}: v_j \neq v_0} b_{ij}^r (c_{ij} - \lambda_j) + \left(\sum_{(v_i, v_0) \in \mathcal{A}} b_{i0}^r c_{i0} \right) - \lambda_0 \quad (3.21)$$

with $b_{ij}^r = 1$ if arc (v_i, v_j) is traversed in route r .

The pricing problem is solved with a dynamic programming labeling algorithm. In the algorithm, a label L represents a path P starting at depot v_0 and combining several pickup and delivery trips. It is defined by a vector $L = (v, c, t, l, q, \tilde{q}, R = \{R_1, \dots, R_n\})$ of size $n + 6$.

The first three resources are classic, v is the last node visited, c is the current reduced cost of P , and t is the time at which node v is left. Next resources are specific to the MTMPD-RD. Resource l is the latest time allowed to be back to the depot to satisfy due dates of inbound requests served in the current trip. Resources q and \tilde{q} allow managing the capacity constraint in presence of inbound and outbound requests. The reachability vector R is defined with three possible states for each request: (i) $R_i = C$ if v_i is reachable in the current trip (ii) $R_i = N$ if v_i is unreachable in the current trip because of resource constraints but is reachable in the next trip (iii) $R_i = U$ if node v_i cannot be reached anymore in the route.

The labelling algorithm consists in extending labels starting from the depot. When a label is back to the depot, it corresponds to a complete route. Such a route can be added to the master problem (if its reduced cost is negative) or extended again as multi-trip is allowed. Label initialization and extension rules are not described here. We simply remind the dominance rule. Given two labels L^1 and L^2 , L^1 dominates L^2 if:

- $v^1 = v^2, c^1 \leq c^2, t^1 \leq t^2, l^1 \geq l^2, q^1 \leq q^2, \tilde{q}^1 \leq \tilde{q}^2,$
- for every $i \in \mathcal{N}$: if $R_i^2 = C$, then $R_i^1 = C$ and if $R_i^2 = N$, then $R_i^1 \in \{C, N\}$

3.5.2 Adaptation of the branch-and-price algorithm for driver-independent equity constraints

In the driver-independent case, solving the routing problem for a period t requires to solve a MTMPD-RD with route painfulness limited to a given threshold π_{max} .

The only impact in the master problem is that set Ω is more restricted. Routes with a painfulness larger than π_{max} are not eligible anymore. This limitation has to be managed by the pricing problem to avoid generating routes with an excessive painfulness. It is simply achieved with an additional resource.

We call this resource π . It stores the total painfulness of home pickups or deliveries that have been performed in the trajectory of the label. When creating a new label, resource π is initialized to 0. When a label is extended to a customer node v_i , the resource is updated: $\pi \leftarrow \pi + \pi_i^t$; otherwise it is not. If π exceeds π_{max} , the label is discarded. Finally, the dominance rule is modified. A label L^1 can only dominate a label L^2 if $\pi^1 \leq \pi^2$, in addition to the conditions of the initial dominance rule.

3.5.3 Adaptation of the branch-and-price algorithm for driver-dependent equity constraints

With driver-dependent constraints, the limit of painfulness varies between drivers within a same period. Hence, a route can be valid for a driver but not for another driver. Limiting set Ω is not enough. The intuitive way to tackle this difficulty would be to introduce a subset $\Omega_k \subset \Omega$ of eligible routes for each driver, which would result in the new master problem:

$$\min \sum_{m_k \in \mathcal{M}} \sum_{r \in \Omega_k} c_r \theta_{kr} \quad (3.22)$$

subject to

$$\sum_{m_k \in \mathcal{M}} \sum_{r \in \Omega_k} a_i^r \theta_{kr} = 1 \quad \forall i \in \mathcal{N} \quad (3.23)$$

$$\sum_{r \in \Omega_k} \theta_{kr} = 1 \quad \forall m_k \in \mathcal{M} \quad (3.24)$$

$$\theta_{kr} \geq 0 \quad \forall m_k \in \mathcal{M}, \forall r \in \Omega_k \quad (3.25)$$

Then, a pricing problem should be defined individually for each driver and solved exactly as explained in Section 3.5.2, with λ_0 being replaced by the appropriate dual variable from constraints (3.24) and the painfulness limit being set to the appropriate value. In order to avoid having to solve K pricing problems at each iteration of the column generation algorithm and to limit symmetries in the master problem, we propose an alternative formulation.

For that formulation, we first need to partition drivers into groups having equivalent cumulative painfulness. Hence, drivers are split into L groups, denoted g_l , with K_l drivers of equivalent painfulness limit π_{max}^l in group g_l . Without loss of generality, we assume that groups are indexed in the increasing order of painfulness limits. Hence, drivers of group g_1 are the more restrictive and only accept routes with small painfulness, while drivers of group g_L can be assigned the more painful routes, as long as this painfulness does not go beyond π_{max}^L . Then, the master problem can be written:

$$\min \sum_{r \in \Omega} c_r \theta_r \quad (3.26)$$

subject to

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (3.27)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (3.28)$$

$$\sum_{r \in \Omega: \pi_r > \pi_{max}^{l-1}} \theta_r \leq \sum_{l'=l}^L K_{l'} \quad \forall l \in \{2, \dots, L\} \quad (3.29)$$

$$\theta_r \geq 0 \quad \forall r \in \Omega \quad (3.30)$$

Constraint (3.29) can easily be explained for $l = L$. Only the drivers of group g_L are allowed to have a route with workload larger than the limit of group g_{L-1} . So, at most K_L routes exceeding π_{max}^{L-1} can be selected:

$$\sum_{r \in \Omega: \pi_r > \pi_{max}^{L-1}} \theta_r \leq K_L \quad (3.31)$$

Similarly, at most $K_{L-1} + K_L$ routes exceeding π_{max}^{L-2} can be selected:

$$\sum_{r \in \Omega: \pi_r > \pi_{max}^{L-2}} \theta_r \leq K_{L-1} + K_L \quad (3.32)$$

The generalization to any $l \in \{2, \dots, L\}$ follows. Other constraints are not changed. Set Ω is restricted to routes with a painfulness not larger than π_{max}^L .

Because constraints (3.27) and (3.28) remain unchanged, there is still one unsigned dual variable λ_i per patient and one non-positive dual variable λ_0 for the limit on the fleet size. The new set of constraints (3.29) implies $L - 1$ new non-positive dual variables, λ_0^l for $l \in \{2, \dots, L\}$.

The reduced cost of a route r becomes:

$$\bar{c}_r = c_r - \left(\sum_{i \in \mathcal{N}} a_i^r \lambda_i \right) - \lambda_0 - \sum_{l \in \{2, \dots, L\}: \pi_{max}^{l-1} < \pi_r} \lambda_0^l \quad (3.33)$$

Equivalently,

$$\bar{c}_r = \sum_{(v_i, v_j) \in \mathcal{A}: v_j \neq v_0} b_{ij}^r (c_{ij} - \lambda_j) + \left(\sum_{(v_i, v_0) \in \mathcal{A}} b_{i0}^r c_{i0} \right) - \left(\sum_{l \in \{2, \dots, L\}: \pi_{max}^{l-1} < \pi_r} \lambda_0^l \right) - \lambda_0 \quad (3.34)$$

Finding new routes with negative reduced cost can still be achieved solving the pricing problem as defined in Section 3.5.1 with an additional resource π for painfulness. The way π is managed in the dynamic programming algorithm is however a little bit more tricky than in Section 3.5.2.

When creating a new label, resource π is initialized to 0. When a label is extended to a customer node v_i , the resource is updated: $\pi \leftarrow \pi + \pi_i^t$. Furthermore, if this update implies that π crosses threshold π_{max}^{l-1} for a given l , the dual variable λ_0^l is subtracted from resource c (the resource that computes the reduced cost). Note that this dual variable being non-positive, it remains desirable to have a small value for resource π and the dominance rule remains valid. A label is discarded if π exceeds π_{max}^L .

When a new route with negative reduced cost is found, it is only added to Constraints (3.29) for values of l allowed by its painfulness: $\pi_r > \pi_{max}^{l-1}$.

3.6 Computational experiments

In this section, we describe our computational experiments and discuss the results. This section is organized in three parts. First, we describe the 30 benchmark instances that we introduced. Second, we report the computational results and compare the different frameworks through several measures: (i) feasibility, (ii) trade off between equity and cost, (iii) daily equity and evolution of equity along the time horizon, (iv) computing times. Finally, we summarize the main findings.

All algorithms are implemented in the C++ programming language, and linear programs are solved using the commercial solver IBM CPLEX 12.9. Experiments are executed on a computer with a 3.6 GHz CPU and 32 GB of RAM computing on Windows 10. All computational times are given in seconds. Unless explicitly indicated, the computational time is limited to 600 seconds for each daily routing problem and for the RFESPI assignment MILP. However, if neither a feasible solution is found nor infeasibility is proven within 600 seconds, the algorithms keep computing until one of those occurs. For all instances, we apply the 6 solution frameworks presented in Section 3.4 with the following parameter values:

- RFESPI: no parameter
- RFES: no parameter
- SPCE: $\alpha \in \{1.1, 1.08, 1.06, 1.04, 1.02, 1.01\}$
- MPCE: $\alpha \in \{1.1, 1.08, 1.06, 1.04, 1.02, 1.01\}$
- MPAE: $\alpha_s = 1.1$ and $\alpha_e \in \{1.08, 1.06, 1.04, 1.02, 1.01\}$
- LPAE: $\alpha \in \{1.1, 1.08, 1.06, 1.04, 1.02, 1.01\}$

This gives us a total of 25 runs per instance, that is, a total of 750 runs. As for MPAE, coefficient α_s is set to 1.1 for all runs, from now we use indifferently terms α_e and α to simplify the notation. Also, from now we call a pair $p = (A, \alpha)$, the combination of an approach A with a parameter value α . For RFESPI and RFES that do not require a parameter α , we denote $p = (RFESPI, -)$ and $p = (RFES, -)$ respectively.

3.6.1 Generation of instances

Experiments were conducted on a benchmark of realistic instances extracted from the city of Aix-en-Provence, France. The road network of the city is extracted from OpenStreetMap. We first generated 10 instances. Each instance is composed of 20 periods of 50 requests and corresponds to a time horizon of a month (assuming that drivers work 5 days a week). Generating these instances implied generating 200 MTMPD-RD instances. We actually generated 200 sets of customer requests. All other parameters are the same in all daily routing problems of all instances, except the fleet size, which was set as explained below.

Given any instance and any period t , the depot is the Public Hospital of the city. Customer locations are selected randomly and uniformly. However, they are more likely to be selected in urban areas than in the countryside as we use a real road network and the denser an area is, the more nodes it contains. The probability that a request is inbound is fixed to 50%. The demands are unitary: they correspond to the transportation of single patients from or to the hospital. The daily time horizon is set to $[0 \text{ min.}, 240 \text{ min.}]$ corresponding to a working period of 4 hours. Pick-up and drop-off times at the depot (T_P and T_D) are fixed to 3 minutes, as well as service times at customer locations. Vehicles are minivans of capacity

4. Release and due dates are generated to ensure an acceptable ride time for customers that is, the time needed for a direct ride between pickup and drop-off locations plus 20 minutes. For a complete description of the generation of MTMPD-RD instances readers are referred to [Agius et al. \(2022\)](#).

The painfulness of a request π_i^t is a value selected uniformly in set $\{0, 1, 2, 4, 8\}$. Value 0 corresponds to an easy request and 8 to a hard one. A logarithmic scale is used to better differentiate requests.

To determine the fleet size K of an instance, we first compute the minimum number of vehicles K^t needed at each period t . To do so, we repeatedly solve each daily routing problem with the branch-and-price algorithm, without equity constraints and with decreasing limits on the number of vehicles, as long as the daily routing problem is feasible. It gives K^t . Then, the fleet size is set to $K = \max_{t \in \{1, \dots, T\}} K^t$. It guarantees that the daily vehicle routing problems are all feasible, without overestimating too much the number of available vehicles.

From those 10 instances, we deduced 10 instances of 10 days (two working weeks) and 10 instances of 5 days (one working week) by selecting the 10 (resp., 5) first days of the 20-days instances. Fleet sizes are set to the maximal value of K^t on the appropriate subset of periods.

It finally gives us 3 sets of instances and a total of 30 instances. Instance sets are named S_T and instances $I_{T/i}$ where, $T \in \{5, 10, 20\}$ corresponds to the number of periods in the instance and $i \in \{1, \dots, 10\}$ corresponds to the instance ID.

3.6.2 Feasibility

Enforcing the equity too strictly with equity constraints sometimes prevent from finding a feasible solution (in at least one period of the horizon). Tables 3.2, 3.3 and 3.4 show the number of feasible solutions found for each set of instances by RFESPI and RFES, SPCE and MPCE, MPAE and LPAE, respectively.

Table 3.2: Feasibility of RFESPI and RFES frameworks

Approach	RFESPI	RFES
#feas inst for $T = 5$	10	10
#feas inst for $T = 10$	10	10
#feas inst for $T = 20$	10	10
#feas inst total	30	30

Table 3.3: Feasibility of SPCE and MPCE frameworks

Approach α	SPCE						MPCE					
	1.1	1.08	1.06	1.04	1.02	1.01	1.1	1.08	1.06	1.04	1.02	1.01
#feas inst for $T = 5$	10	10	10	5	0	0	10	10	10	8	1	0
#feas inst for $T = 10$	10	10	9	2	0	0	10	10	10	8	0	0
#feas inst for $T = 20$	10	10	8	1	0	0	10	10	10	8	0	0
#feas inst total	30	30	27	8	0	0	30	30	30	24	1	0
	95/180						115/180					

Table 3.4: Feasibility of MPAE and LPAE frameworks

Approach α/α_e	MPAE					LPAE					
	1.08	1.06	1.04	1.02	1.01	1.1	1.08	1.06	1.04	1.02	1.01
#feas inst for $T = 5$	10	10	10	10	9	10	10	10	10	10	9
#feas inst for $T = 10$	10	10	10	10	10	10	10	10	10	9	9
#feas inst for $T = 20$	10	10	10	10	10	10	10	10	10	10	10
#feas inst total	30	30	30	30	29	30	30	30	30	29	28
	149/150					177/180					

Instances were generated in a way that ensures feasibility when equity is ignored. Table 3.2 confirms that solutions are found for all instances for frameworks RFESPI and RFES, in which the daily routing problem is not constrained by equity.

Table 3.3 shows that feasibility with SPCE and MPCE frameworks is widely impacted by α . SPCE is driver-independent that is, for a given period all drivers have the same equity constraint. Results show that when α is small, the constraint can be too restrictive and makes the problem infeasible. Indeed, only 8 out of 30 feasible solutions are found when $\alpha = 1.04$, and 0 when $\alpha \leq 1.02$. Without surprise, the larger the number of periods, the larger the risk that a period does not admit a feasible solution. MPCE is driver-dependent, hence equity constraints are adjusted everyday and for every driver. This approach is more flexible: at a given period, solutions with only a few maximally-constrained routes could be feasible. Indeed, routes that are more painful can be assigned to drivers that received less than the average of painfulness in previous periods. When $\alpha = 1.04$, feasible solutions are found for 24 instances. MPCE however still fails for most instances when $\alpha \leq 1.02$.

Table 3.4 shows the effectiveness of MPAE and LPAE frameworks to ensure feasibility. For MPAE even when α_e is set to a very constraining value (e.g. 1.02 or 1.01), daily routing problems are not very constrained for most periods, and solutions are found for most instances: only one instance out of 30 has no feasible solution when $\alpha_e = 1.01$, 0 for larger values of α . Note that with $\alpha = 1.01$, MPAE finds solutions for 9 instances out of 10 when $T = 5$, against 10 when $T = 10$ and $T = 20$. This results might look counter intuitive because with more periods, instances look more likely to contain an infeasible period. The fact is that the risk only applies to a few periods; so, for MPAE, the chance to get a feasible solution does not necessarily depend on the number of periods. As MPAE, LPAE allows finding feasible solutions for most instances (177 out of 180). Furthermore, the larger the number of periods, the larger the absolute gap allowed for a given value of α , because the total painfulness increases. In LPAE, unfeasibility can only occur at the last period. Results show that the risk of unfeasibility is slightly increased compared to MPAE. Probably, drivers reach the last periods with more contrasted levels of painfulness and it is harder to reach a very tight balance.

3.6.3 Trade-off between equity and cost

Figures 3.2, 3.3 and 3.4 show the trade-off between equity and cost using the different frameworks, for instances with 5, 10 and 20 periods respectively. Each point corresponds to a pair (A, α) and is located as explained below.

The x-axis corresponds to the average increase (in %) of travel cost against the optimal cost without equity (obtained by RFES and RFESPI). Precisely, let $C(I, A, \alpha)$ denote the total travel cost obtained with framework (A, α) on instance I . The cost increase is $\frac{C(I, A, \alpha) - C(I, RFES, -)}{C(I, RFES, -)} \times 100$. The abscissa for framework (A, α) is set to the average cost increase on the instances for which a feasible solution is found (out of the 10 instances) using this framework.

The y-axis corresponds to the average value of the range (in %) against the average total painfulness. Precisely, let $R(I, A, \alpha)$ be the range obtained with (A, α) on instance I , and $K(I)$ be the fleet size. The ordinate is the average value of $\frac{R(I, A, \alpha)}{\Pi^{1 \rightarrow T} / K(I)} \times 100$ on the instances for which a feasible solution is found.

Note that these figures hide the fact that some frameworks hardly find feasible solutions. This issue was discussed in the previous section. The figures are also somehow biased because points are not always averaged on the same subset of instances. Many interesting comments can however be made from these figures. Below, we review the frameworks one by one.

The RFES framework preserves optimal routing costs. However, results show that applying the BEA strategy for route assignment is not enough to guarantee equity. A range of 14% on average can be

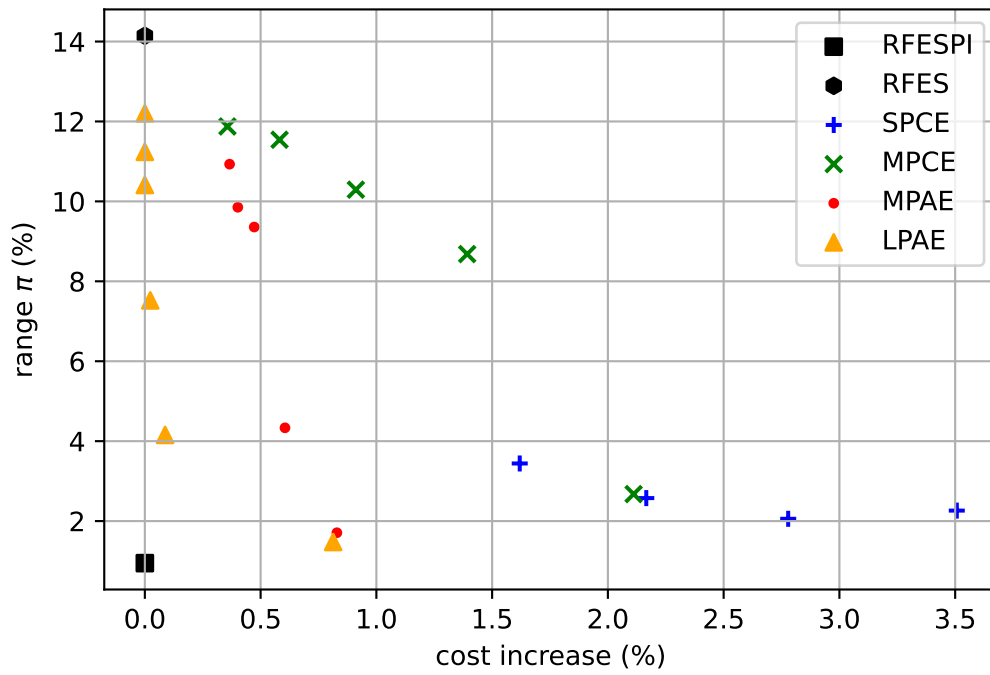


Figure 3.2: Trade-off between equity and cost on 5-days instances

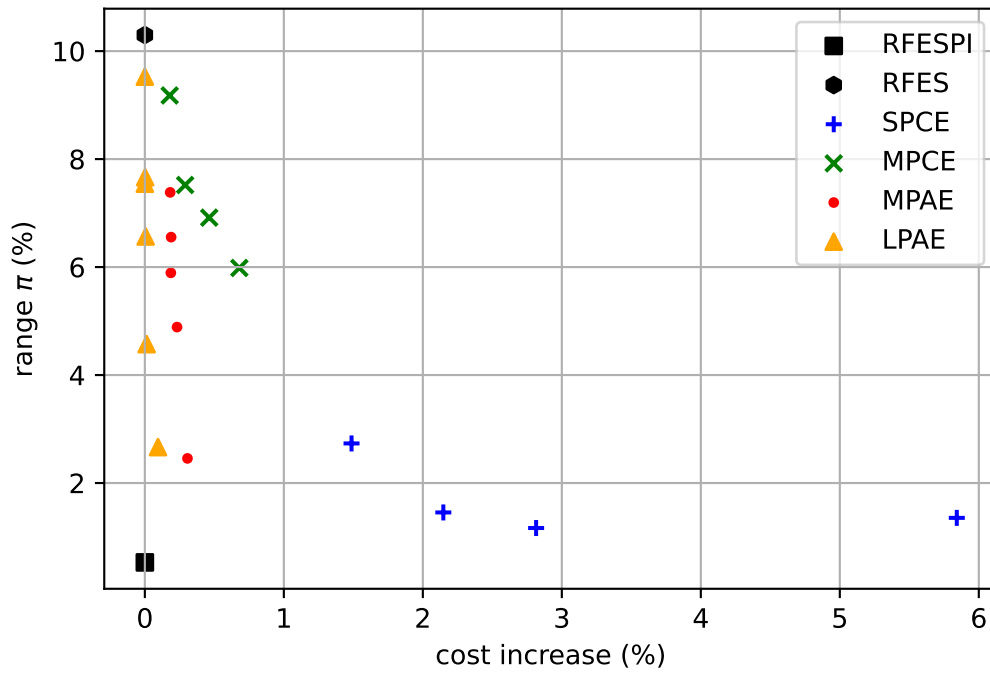


Figure 3.3: Trade-off between equity and cost on 10-days instances

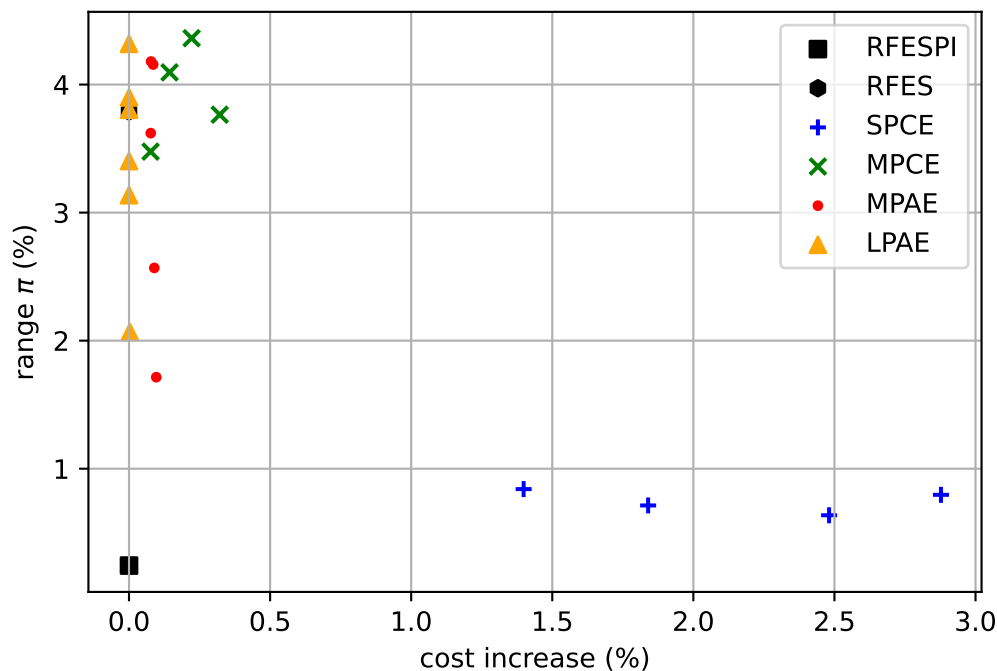


Figure 3.4: Trade-off between equity and cost on 20-days instances

observed for 5-days instances. It is however interesting to see that this value quickly decreases when the horizon length increases. It can easily be explained when looking at the ordinate value $\frac{R(I, A, \alpha)}{\Pi^{1 \rightarrow T} / K(I)} \times 100$. Indeed, $R(I, A, \alpha)$ should not really be impacted by the size of the horizon, $K(I)$ should only be slightly increased when the horizon increases, while $\Pi^{1 \rightarrow T}$ is typically proportional to the size of the horizon.

RFESPI shows on the contrary very small ranges, not far from 0. And, similarly to RFES, it guarantees optimal travel costs. It proves that it is theoretically possible to obtain very well-balanced driver's planings at no cost. The gap in equity between RFES and RFESPI justifies the search for other frameworks, both efficient on cost and equity (recall that RFESPI is not applicable in practice).

The literature baseline SPCE is not completely satisfactory with this regards. It allows reaching excellent values for the range, not far from zero nor RFESPI, but with a clear impact on travel costs. Before entering into details, the figures need some clarifications. For this framework, only four points can be found on each figure. They correspond to the four values of α that allow finding solutions (see Section 3.6.3): 10%, 8%, 6%, 4%. The smaller α , the larger the cost increase, *i.e.*, the further to the right on the figure. The figures show that small values α result in important cost increases, in addition to the risks of infeasibility already observed. However, a very interesting finding is that relatively large values (α set to 10% or 8%) still allow reaching very good ranges (less than 1% on average for the longer horizon), while with these values the cost increases are limited (around 1 to 2%). It shows that the SPCE framework set with an overestimated limit α enables finding feasible, cost-efficient and well-balanced solutions.

MPCE has the same feasibility issues as SPCE, and, so, points are also missing on the figures for small values α . MPCE shows an excellent behavior with regards to travel costs, which are very slightly increased. However, it is not very successful regarding equity. Range are much larger that those obtained with SPCE, and not clearly improved against RFES. This discredits the use of this framework.

With MPAE, the feasibility issue is fixed and points are reported on the figure even for very small values of α ($\alpha = 1\%$). Whatever the value of α , travel costs are very close to the optimal reference value. Regarding equity, the behavior is relatively similar to that of MPCE. However, the fact that very small values can be chosen for α allows achieving very small ranges, *i.e.*, ranges that are almost as good as those obtained with the literature baseline SPCE.

LPAE exhibits a behavior that is very close to MPAE, with a cost increase that is even more limited. Both frameworks are excellent when α is small. This observation might look surprising, seeing that LPAE is basically the same as RFES, except at the last period. It is another very interesting finding of this work that the BEA assignment strategy is not enough to achieve equitable routes, but it suffices to set a strong equity constraint at the last period to reach this goal.

3.6.4 Daily equity and evolution along the time horizon

Results reported in Section 3.6.3 analyze how the different frameworks behave with regards to the range along the whole horizon. Though considering equity with a medium-term perspective is the key topic of this work, it is still relevant to check how they preserve equity on the short term. In this section, we analyze daily equity and the evolution of equity along the time horizon, for the different frameworks. For the sake of conciseness, we limit this study to the 5-days instances.

Daily equity is measured as the range on a given day, *i.e.*, the difference between the most painful and the least painful routes. The evolution of equity along the horizon is measured seeing how the range of accumulated painfulness evolves as periods go. This measure, up to a given period, is called accumulated range. Given a framework (A, α) , an instance I and a day t , we denote $R(I, t, A, \alpha)$ the daily range and $R(I, 1 \rightarrow t, A, \alpha)$ the accumulated range up to day t . The reported daily range (resp., accumulated range) is then the average value of $\frac{R(I, t, A, \alpha)}{\Pi^t / K(I)} \times 100$ (resp., the average value of $\frac{R(I, 1 \rightarrow t, A, \alpha)}{\Pi^{1 \rightarrow t} / K(I)} \times 100$) on the instances for which a feasible solution is found.

Figure 3.5 shows the average daily equity measure on every days of the 5-days horizon. A single value of parameter α is kept for each framework. We arbitrarily choose the smallest value of α for which at least 5 out of 10 instances are feasible. Hence, the selected frameworks are $(RFES, -)$, $(SPCE, 1.04)$, $(MPCE, 1.04)$, $(MPAE, 1.01)$ and $(LPAE, 1.01)$. Figure 3.6 shows the evolution of the accumulated equity period per period, on the same instances and with the same frameworks. Note that the equity measures reported on the first periods (resp., last periods) of Figure 3.6 exactly correspond to the equity measures reported on Figure 3.5 (resp., Figure 3.2).

The figures show that frameworks RFES and LPAE induce very unbalanced routes at each period. The range is consistently larger than the average painfulness, and goes up to 150% of this value. This strong inequity can be detrimental to framework LPAE, even though this framework allows an excellent equity among drivers at the end of the horizon.

Imbalances can also be observed for framework MPAE, but at a much lower level. The graph reported in Figure 3.5 shows a behavior that might look unexpected at first sight, but that can actually be explained. The graph is the result of two opposite effects. First, the daily range is limited according to the accumulated painfulness. At a given period, the accumulated painfulness is approximately the period number multiplied by the average painfulness per period. So the daily range is less and less constrained. Consequently, the graph is prone to increase period by period. This effect also exists and can very clearly be observed for framework MPCE. Second, the value of parameter α_t decreases period per period, which compensates the previous effect. The combination of these two effects make the daily equity first increasing and then decreasing.

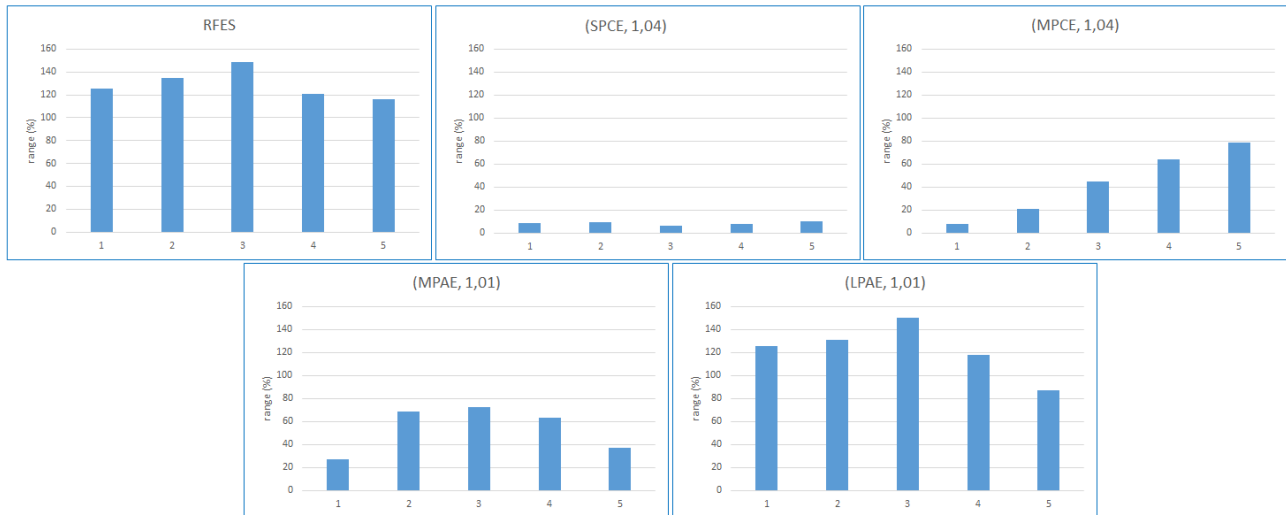


Figure 3.5: Daily equity on 5-days instances

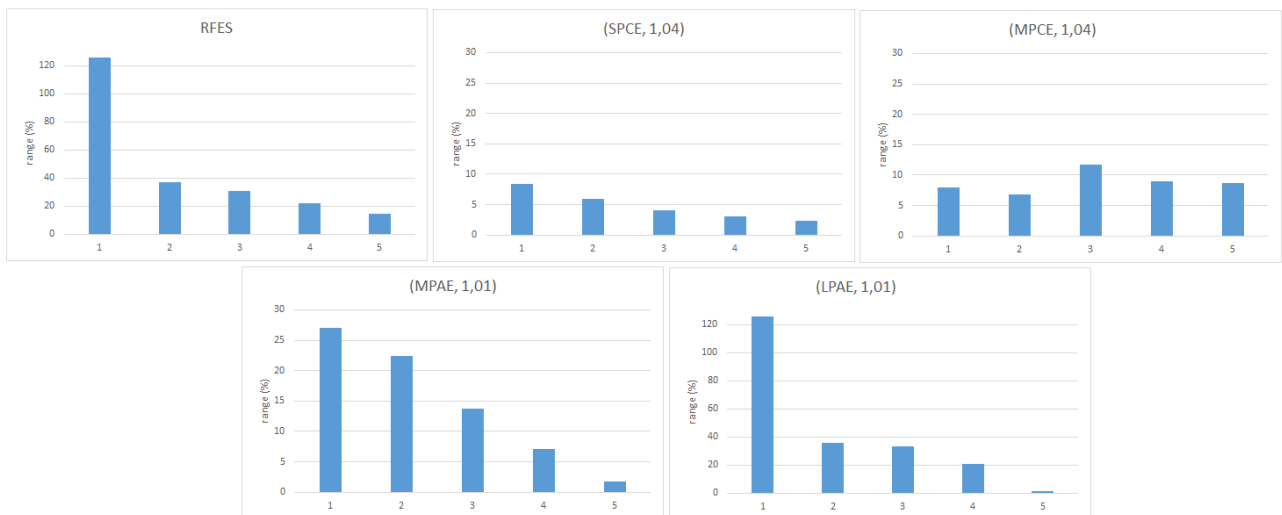


Figure 3.6: Long term equity evolution on 5-days instances

As expected, the literature baseline, SPEC, is the most efficient framework to preserve equity at each period. Very limited inequities are observed on every period. It gives a clear advantage to this framework if limited daily unbalances are preferred.

3.6.5 Computing time

Frameworks SPCE, MPCE, MPAE and LPAE (at the last period) imply managing a resource constraint in the daily routing problem. It results in a new resource in the pricing problem, which is known to weaken dominance rules and could thus increase the combinatorial explosion. Furthermore, frameworks MPCE, MPAE and LPAE (last period) embed a more complex master problem, required to deal with driver-dependent inequity thresholds. In this section, we measure the impact of these changes on computing times. Tables 3.5, 3.6 and 3.7 show the average computing time for each set of instances for frameworks RFESPI and RFES, SPCE and MPCE, MPAE and LPAE respectively. Times are expressed in seconds and only take into account feasible instances.

Table 3.5: Computing times for RFESPI and RFES frameworks

Approach	RFESPI	RFES
avg computing time for $T = 5$	185.6	18.6
avg computing time for $T = 10$	403.0	40.7
avg computing time for $T = 20$	449.0	118.4

Table 3.6: Computing times for SPCE and MPCE frameworks

Approach α	SPCE						MPCE					
	1.1	1.08	1.06	1.04	1.02	1.01	1.1	1.08	1.06	1.04	1.02	1.01
avg computing time for $T = 5$	185.8	221.7	220.6	179.1	-	-	99.8	102.1	60.9	88.4	176.3	-
avg computing time for $T = 10$	395.0	435.3	411.7	361.9	-	-	140.3	136.2	100.4	131.4	-	-
avg computing time for $T = 20$	837.9	923.1	1153.5	917.0	-	-	269.3	259.5	218.4	295.4	-	-

Table 3.7: Computing times for MPAE and LPAE frameworks

Approach α/α_e	MPAE					LPAE					
	1.08	1.06	1.04	1.02	1.01	1.1	1.08	1.06	1.04	1.02	1.01
avg computing time for $T = 5$	101.1	102.0	101.8	146.7	187.2	21.6	21.7	22.1	22.5	29.3	93.5
avg computing time for $T = 10$	135.7	139.3	136.4	143.9	159.8	44.8	44.4	45.7	45.4	48.6	67.9
avg computing time for $T = 20$	268.9	261.5	266.7	266.3	294.1	121.0	120.9	121.1	121.0	120.8	122.5

From Table 3.5, one can see that the RFESPI approach requires much more time than RFES, although they solve the same routing problems everyday. The additional time is due to the solution of the assignment MILP. This formulation permits to find fast feasible solution but proving optimality is hard, hence the solving often reaches the 6 minutes of time limit.

As expected, Tables 3.6 and 3.7 show that the constrained frameworks (SPCE, MPCE and MPAE) require much more computing time than RFES. However, the average computing time with SPCE is roughly 4 times bigger than with the driver-dependant approaches, for the largest instances ($T = 20$). This can be explained by the fact that bounds are more constraining in SPCE. For LPAE, the computing time is less impacted as only the last period involves equity constraints.

Note that the value of α does not seem to have a clear impact on computing times. One can find a few discrepancies in the tables, but only for cases where few instances are solved (e.g., MPCE with $\alpha = 1.02$).

3.6.6 Synthesis: main findings

Table 3.8 summarizes the pros and cons of the five solution frameworks proposed in this chapter. The quality of these frameworks is assessed with regards to the six KPIs (Key Performance Indicators) studied in our experiments:

- Feasibility: Is the framework always successful in returning a solution?
- Travel cost: Does the framework finds solutions with minimal travel costs?
- Medium-term equity: Does the framework return solutions with a strong equity among drivers in the medium-term?
- Daily equity: Does the framework return solutions with a strong equity among drivers on each day?
- Computing times: Does the framework cause a computing times increase?
- Coding burden: Which additional components need to be introduced in the daily routing optimization?

Table 3.8 reports qualitative information for the six KPIs. The terms used in the table are somehow arbitrary but aim at reflecting the conclusions drawn through the experiments and developed in the previous sections.

Table 3.8: Pros and cons of the different frameworks

Framework	Feasibility	Travel cost	Medium-term eq.	Daily eq.	Computing time	Coding burden
RFES	always feasible	optimal	weak	weak	not increased	no impact
SPCE	good for large α	good for large α	excellent for all α	excellent	large increase	equity constraint
MPCE	good for large α	good for large α	weak	medium	moderate increase	driver-dependent equity constraint
MPAE	excellent	excellent	excellent for small α	medium	moderate increase	driver-dependent equity constraint
LPAE	excellent	excellent	excellent for small α	weak	small increase	driver-dependent equity constraint

The first clear conclusion with this synthesis is that applying the BEA strategy is not enough to achieve equity among drivers (RFES framework). Also, framework MPCE does not show any clear advantage against other frameworks. On the contrary, frameworks SPCE, MPAE and LPAE all have strengths and weaknesses, and could be considered as the best framework depending on the context.

3.7 Conclusion

Equity in vehicle routing has shown a growing interest in the last two decades. While the majority of existing studies consider equity on a daily basis, in this chapter we deal with workload equity in vehicle routing over a planning horizon of several periods. The studied problem is dynamic, in the sense that requests are revealed period per period. The equity metric used in our paper is said constant-sum which means that for any feasible solution, the sum of workload assigned to drivers remains constant. The studied routing problem is motivated by healthcare logistics.

In order to balance equity between drivers over the whole planning horizon while minimizing the total travel cost, we propose five frameworks. Namely, Routing-First Equity-Second (RFES), Single-Period Constant Equity (SPCE), Multi-Period Constant Equity (MPCE), Multi-Period Adaptive Equity (MPAE), and Last-Period Adaptive Equity (LPAE). These frameworks are compared to a reference framework, called Routing First Equity Second with Perfect Information (RFESPI). It tries to ensure a perfect equity over the

whole time horizon where all requests of all days are supposed to be known in advance. A dedicated branch-and-price algorithm is used to solve the daily routing problem and is adapted for the different solution frameworks.

From our experiments, we can derive several conclusions. We show that the RFES framework does not necessarily allow to balance the workload over the time horizon. The unbalance reduces when the time horizon is longer. We show that frameworks SPCE, MPAE and LPAE have different strengths and weaknesses. The MPCE framework does not show any clear advantage against other frameworks. Table 3.8 summarizes the advantages and the drawbacks of the different frameworks.

This chapter is one of the first works investigating equity in vehicle routing over a time horizon of several periods. We believe it opens up different research directions. The first straightforward perspective is to consider that the workload depends on the driver. In fact, in some applications, the painfulness of a request can depend on the driver skills, characteristics or preferences. This will have an important impact on the characteristics of the metric and the solution approach of the daily problem. A second perspective would be to address the studied problem with a variable-sum metric (such as route cost or duration). This will have an important impact on the way the work balance is calculated and the way the daily problem is solved. A last perspective of this work would be to address the stochastic version of this problem by considering a stochastic demand. It falls within the continuity of the work of [Wang et al. \(2022\)](#). This problem is very challenging. One of the main challenges is to adapt the proposed frameworks and to develop efficient exact solution methods for the daily routing problem.

Workload equity for a routing problem with a variable-sum equity metric

Chapter 2 introduces the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD), a routing problem motivated by the non-emergency transportation of patients and solved by branch-and-price solution method. Chapter 3 provides the multi-period version of the problem in which we consider equity for drivers regarding the painfulness of work, a constant-sum metric. In this chapter, we study the MTMPD-RD with equity constraints on the routing cost, which is a variable-sum metric. This study is relevant because a variable-sum metric covers several classical objective functions related to the route design (travel time, working time, fuel consumption...). Moreover, the variable-sum metrics are challenging objectives as their modeling requires non-linear formulations.

Equity is ensured with constraints limiting the deviation of each route cost from the average route cost. This constraint is the variable-sum version of the equity constraint considered in the daily routing problem of SPCE approach of Chapter 3. We show that the straightforward set partitioning formulation can not be solved efficiently by the standard branch-and-price approach. Instead, we propose two solution methods.

First, we propose a heuristic algorithm based on a dichotomic search where at each step, a simplified version of the problem is solved and the structure of the solution guides the search for the next step. Our algorithm is heuristic, since the wrong search space alternative may be selected in some cases. If such cases are not met, our approach can prove optimality.

Our second approach is exact. It is a branch-and-price algorithm based on a new model where the variables (columns) are indexed by a route and a driver.

Both the heuristic and the branch-and-price algorithm are tested on instances derived from Chapter 2 involving 10 and 25 customers, also the heuristic is tested on the 50-customers instances. The cost of equity is evaluated. The branch-and-price algorithm shows on small instances that there exist equitable solutions with a very low impact on the cost. The heuristic provides very good solutions; in many instances, it proves the optimality of the solutions.

Contents

4.1 Introduction	83
--	----

4.2 Literature review	84
4.2.1 Equity in VRP	84
4.2.2 Constraint-based approaches for workload equity	84
4.2.3 Equity in medical transportation	85
4.3 Problem description	85
4.3.1 The MTMPD-RD: a reminder	85
4.3.2 Measure of equity	88
4.4 Solution methods	89
4.4.1 Branch-and-price algorithm for the MTMPD-RD: a reminder	90
4.4.2 Weakness of the initial model for the MTMPD-RD-WE	91
4.4.3 A Dichotomy-based Heuristic	92
4.4.4 A branch-and-price approach based on a driver-indexed model	99
4.5 Computational experiments	103
4.5.1 Benchmark instances	103
4.5.2 Results and analysis	105
4.6 Conclusion	108

4.1 Introduction

The vehicle routing literature is rich, and many variants are derived from real-life applications. Among the most emerging variants, equity has gained interest in the last two decades (Vidal et al., 2020). Efficient solutions are usually not equitable. Hence, the challenge is to dispatch fairly workload among drivers limiting the impact on the total cost.

Considering equity in a routing problem requires making some important modeling choices. First, one needs to decide on the workload to balance fairly. We call it the equity metric and we distinguish two families. An equity metric is said constant-sum if, for any feasible solution, the sum of workload assigned to drivers remains constant (*e.g.*, number of customers served, total customer load served, service time). Otherwise, a metric is said variable-sum if the sum of workload assigned to drivers differs between solutions (*e.g.*, routing cost/distance/time). In this chapter, we study the routing cost.

The second important modeling choice is the way equity is measured. In the literature, equity is usually measured with an inequality function to minimize (called equity function or equity measure) and is combined with the min-cost objective. Equity can also be handled with constraints while minimizing only the routing cost. This approach has rarely been investigated in the literature despite its relevance. Hence, we opt for constraint-based modeling.

The proposed equity constraints consist in limiting the deviation of each route cost from the average route cost. This constraint is the variable-sum version of the equity constraint considered in the daily routing problem of SPCE approach of Chapter 3. These constraints are more difficult to manage in the variable-sum version as the average route cost is unknown and depends directly on the solution (it is not mean-preserving). We use a test-bed problem selected in the field of healthcare logistics, the Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD) presented in Chapter 2 (and in Agius et al. (2022)). We call the derived problem, the MTMPD-RD with workload equity and denote it as MTMPD-RD-WE.

We propose two solution methods to solve the MTMPD-RD-WE, a heuristic algorithm and an exact solution. The former is based on a dichotomic search where at each step, a simplified version of the problem is solved; the structure of the solution guides the search for the next step. Although heuristic, the algorithm proves optimality in some specific cases. The latter is a branch-and-price algorithm. The particularity of the method is that columns in the column generation are indexed by route and driver which require adaptations at each level of the algorithm: the master problem, the column generation, the pricing problem and the branching rules.

The contributions of this chapter are threefold. First, we model workload equity with a constraint-based approach that is original regarding the literature. Second, we propose two original solution methods to manage the equity constraints, a dichotomy-based heuristic and an exact method. Computational experiments on realistic instances are conducted to evaluate the impact of the proposed approach to deal with equity on routing costs.

The chapter is organized as follows. Section 4.2 reviews the related literature. Section 4.3 describes the problem, first it provides a reminder of the MTMPD-RD, second it describes how equity is measured. Section 4.4 describes the proposed solution methods. First, it shows that the straightforward set partitioning formulation cannot be solved efficiently by the standard branch-and-price approach. Then, the two solution methods are explained. Section 4.5 details the experiments and discusses obtained results, particularly we study the impact of equity on the routing costs. Finally, Section 4.6 concludes this chapter, summarizing our findings and suggesting possible extensions to this work.

4.2 Literature review

In this section, we review the relevant literature in three parts. First, we review the main contributions to workload equity in VRP. Then, we review constraint-based approaches for workload equity in VRP, particularly we focus on the solution methods. Finally, as our test-bed problem, the MTMPD-RD-WE arises from the context of medical transportation, we review the literature on equity for medical transportation literature.

4.2.1 Equity in VRP

In their review of equity in the operations research, field [Karsu and Morton \(2015\)](#) point out the growth of interest in equity for routing problems.

The recent paper of [Matl et al. \(2018\)](#) reviews the literature on workload equity in routing problems. Also, they formalize some important definitions and provide an axiomatic analysis of the main equity functions to determine which desirable properties are respected. They also determine properties guiding the choice of equity functions. The authors present some important properties and validate them with computational experiments on small instances. We review and use [Matl et al. \(2018\)](#) more in detail in Section 4.3.2. The same authors extend their work [Matl et al. \(2019\)](#) studying different equity metrics and conclude that the metric has more impact on the solutions than the equity function.

The most studied variant of the routing problem with equity is the bi-objective version of the Capacitated Vehicle Routing Problem (CVRP) where the min-cost objective is combined with the range equity function, computed as the difference between the longest and the shortest route. This problem is called CVRP with Route Balancing (CVRPRB) and was introduced by [Jozefowiez et al. \(2002\)](#). It has been largely studied ([Jozefowiez et al., 2007](#); [Borgulya, 2008](#); [Oyola and Løkketangen, 2014](#); [Lacomme et al., 2015](#)). A weakness of this formulation is that some routes can be artificially worsened to improve the workload balance ([Jozefowiez et al., 2002](#)). This issue actually occurs for any combination of a variable-sum metric with a non-monotonic equity function (any equity function different from the min-max or lexicographic min-max objectives ([Matl et al., 2018](#))).

4.2.2 Constraint-based approaches for workload equity

[Blakeley et al. \(2003\)](#) conducted research to optimize periodic maintenance operations for elevators, considering workload equity. They included an equity objective to minimize the difference between the longest and shortest routes, alongside objectives for cost and overtime minimization. They also established lower and upper bounds on the workload of each route to ensure workload distribution within acceptable limits.

In the context of the delivery operations of the San Francisco Public Library, balancing the utilization of trucks is essential. Hence, [Apte and Mason \(2006\)](#) proposed equity constraints to bound the number of stops and the load of vehicles. The problem is solved with a Modified Generalized Assignment heuristic.

[Kim et al. \(2006\)](#) study a waste collection vehicle routing problem with time windows. While the primary objective is to minimize total costs, equity concerns related to route compactness and workload balance are indirectly addressed within a cluster-based solution procedure. This solution method is compared with a classical insertion procedure. Experiments highlight the outperforming of the cluster-based solution procedure to generate better-balanced routes.

[Goodson \(2014\)](#) investigates the election day routing of rapid response attorneys to poll observers in the United States, emphasizing the importance of ensuring a fair allocation of workload. The study

explores two alternative models: one model incorporates equity by considering the number of assigned polling stations per attorney, while the other model focuses on route duration as a measure of equity. Both models incorporate constraints to manage equity, specifically by enforcing a maximum bound on the range between the highest and lowest workload routes.

Within the existing literature, several papers have introduced exact solution methods for routing problems involving equity. However, in these papers, equity is commonly addressed by incorporating an explicit objective, either within a single-objective (Valle et al., 2011) or a bi-objective model (Halvorsen-Weare and Savelsbergh, 2016). Notably, all papers that address equity through constraints utilize heuristic methods. A key contribution of our work is the introduction of an exact solution method, setting it apart from previous approaches in the field.

4.2.3 Equity in medical transportation

Liu et al. (2013) study a weekly home health care logistics problem where the objective is to minimize the longest route (min-max) in the week to ensure reasonable working hours.

Triki et al. (2014) define a periodic home health care planning problem in which both the total cost and equity are considered and combined within a weighted-sum. In this work, the concern on equity is not for drivers but on the daily planning, the goal is to plan well-balanced days. Each day a target working time is defined and the equity objective is to minimize the gap between the actual working time and the target working time for each day.

Liu et al. (2021) consider the satisfaction of caregivers both on a daily basis and in the long term. Some constraints limit the daily working time of caregivers and on the whole horizon, the range of the accumulated working time assigned to caregivers is minimized. This objective is combined within a weighted-sum objective involving routing cost minimization and satisfaction of patients measured by penalizing reassignment to multiple caregivers. A sensitivity analysis shows that choosing an objective function involving only one sub-objective causes a high level of dissatisfaction for caregivers or patients.

Cappanera and Scutellà (2015) study several extensions of a home care services problem. Particularly they study two equity objectives max-min and min-max. The former maximizes the minimum operator utilization factor while the latter minimizes the maximum operator utilization factor. The results on computational experiments highlight the pros and cons of each objective. The authors conclude that the max-min is preferable when the focus is on equity and the min-max should be preferred when the focus is on cost.

4.3 Problem description

The studied problem, the MTMPD-RD-WE is an extension of the MTMPD-RD. Hence, we first remind the definition of the MTMPD-RD and provide a Mixed Integer Linear Programming (MILP) formulation (Subsection 4.3.1). Then, we justify and explain how equity is measured and show the adaptations of the model of the MTMPD-RD to integrate equity (Subsection 4.3.2).

4.3.1 The MTMPD-RD: a reminder

Description

The Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD) is a routing problem motivated by the non-emergency transportation of patients. It con-

sists in transporting patients between their home and hospital for appointments. We consider a single hospital and a single depot located at the hospital (v_0). We distinguish two types of requests: inbound and outbound requests. Inbound requests consist in transporting patients from their home to the hospital, while outbound requests consist in transporting patients from the hospital to their home. We denote by \mathcal{N}^+ the set of inbound requests and by \mathcal{N}^- the set of outbound requests. $\mathcal{N} = \mathcal{N}^+ \cup \mathcal{N}^-$ is the union of these two sets. We also equivalently denote $\mathcal{N} = \{1, \dots, n\}$, where the customer location of request i is v_i . The transportation network can then be represented by a complete graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{v_0, v_1, \dots, v_n\}$ and $\mathcal{A} = \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}, i \neq j\}$. A travel time t_{ij} and a travel cost c_{ij} are defined on each arc $(v_i, v_j) \in \mathcal{A}$. In what follows, we set $c_{ij} = t_{ij}$. A set $\mathcal{M} = \{m_1, \dots, m_K\}$ of K drivers is available to ensure the service. Driver vehicles are identical with capacity Q .

With each request, $i \in \mathcal{N}$ is associated with a demand $q_i \geq 0$, a release date r_i , a due date d_i , and a service time s_i . The demand indicates the number of people involved in the request. The release date is the time at which these people are available at the pickup node. The due date is the latest allowed arrival time at the delivery node. The service time is the time needed at the customer location for pickup (inbound request) or drop off (outbound request).

Due to release and due date constraints and vehicle capacity, vehicles are enforced to come back regularly to the depot; hence multiple trips are allowed. Each trip is allowed to mix inbound and outbound requests. When leaving the depot (starting a new trip), a vehicle can either be empty (the trip will not fulfill any outbound request) or not. In the second case, a constant pickup time T_P is counted, before the vehicle can leave and after all release dates associated with the outbound requests served in the trip are passed. When returning to the depot (ending a trip), a vehicle can again either be empty (the trip did not fulfill any inbound request) or not. In the second case, a constant time T_D is counted for drop-off. The daily time horizon is $[0, T_{max}]$. Vehicle routes consist of one or several successive trips, starting and coming back to the depot. The objective of the MTMPD-RD is to minimize the routing cost while servicing all requests with K routes. The reader are referred to Chapter 2 (Section 2.3.1) or Agius et al. (2022) for a more detailed description of the problem.

MILP formulation

In order to represent the trips, we introduce the set $\mathcal{H} = \{0, \dots, n-1\}$ to number trips within a vehicle route. In the worst case, a vehicle can use up to n trips. The decision variables are reported in Table 4.1. Note that for time variables τ_i^{kt} and for load variables u_i^{kt} , we introduce a duplicate of the depot, v_{n+1} , to differentiate between the start and the end of the trip.

Table 4.1: Decision variables

Variables	
$x_{ij}^{kt} \in \{0, 1\}$	1 if trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ uses arc $(v_i, v_j) \in \mathcal{A}$, 0 otherwise
$y_i^{kt} \in \{0, 1\}$	1 if trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ serves request $i \in \mathcal{N}$, 0 otherwise
$i^{kt} \in \{0, 1\}$	1 if at least one inbound request is served in trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$, 0 otherwise
$o^{kt} \in \{0, 1\}$	1 if at least one outbound request is served in trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$, 0 otherwise
$\tau_i^{kt} \geq 0$	time at which trip $t \in \mathcal{H}$ of vehicle $m_k \in \mathcal{M}$ starts service at node $v_i \in \mathcal{V} \cup \{v_{n+1}\}$
$u_i^{kt} \geq 0$	load of vehicle $m_k \in \mathcal{M}$ on trip $t \in \mathcal{H}$ leaving node $v_i \in \mathcal{V} \cup \{v_{n+1}\}$

Using variables of Table 4.1, the MTMPD-RD is formulated as a Mixed Integer Linear Program (MILP)

as follows:

$$\min \sum_{(v_i, v_j) \in \mathcal{A}} c_{ij} \sum_{m_k \in \mathcal{M}} \sum_{t \in \mathcal{H}} x_{ij}^{kt} \quad (4.1)$$

subject to

$$\sum_{m_k \in \mathcal{M}} \sum_{t \in \mathcal{H}} y_i^{kt} = 1 \quad \forall i \in \mathcal{N} \quad (4.2)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_i\}} x_{ji}^{kt} = y_i^{kt} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.3)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_i\}} x_{ij}^{kt} = y_i^{kt} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.4)$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_0\}} x_{0j}^{kt} \leq 1 \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.5)$$

$$\sum_{v_i \in \mathcal{V} \setminus \{v_0\}} x_{i0}^{kt} \leq 1 \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.6)$$

$$i^{kt} \leq \sum_{i \in \mathcal{N}^+} y_i^{kt} \leq |\mathcal{N}^+| i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.7)$$

$$o^{kt} \leq \sum_{i \in \mathcal{N}^-} y_i^{kt} \leq |\mathcal{N}^-| o^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.8)$$

$$u_0^{kt} = \sum_{i \in \mathcal{N}^-} q_i \times y_i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.9)$$

$$u_j^{kt} \geq u_i^{kt} + q_j - Q(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, j \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.10)$$

$$u_j^{kt} \geq u_i^{kt} - q_j - Q(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, j \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.11)$$

$$u_{n+1}^{kt} = \sum_{i \in \mathcal{N}^+} q_i \times y_i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.12)$$

$$u_i^{kt} \leq Q \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.13)$$

$$\tau_j^{kt} \geq \tau_0^{kt} + T_P \times o^{kt} + t_{0j} - T_{max}(1 - x_{0j}^{kt}) \quad \forall (v_0, v_j) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.14)$$

$$\tau_j^{kt} \geq \tau_i^{kt} + s_i + t_{ij} - T_{max}(1 - x_{ij}^{kt}) \quad \forall (v_i, v_j) \in \mathcal{A}, i \neq 0, j \neq 0, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.15)$$

$$\tau_{n+1}^{kt} \geq \tau_i^{kt} + s_i + t_{i0} - T_{max}(1 - x_{i0}^{kt}) \quad \forall (v_i, v_0) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.16)$$

$$\tau_{n+1}^{kt} \geq \tau_0^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.17)$$

$$\tau_0^{k,t+1} \geq \tau_{n+1}^{kt} + T_D \times i^{kt} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \setminus \{n-1\} \quad (4.18)$$

$$T_{max} \geq \tau_{n+1}^{k,n-1} + T_D \times i^{k,n-1} \quad \forall m_k \in \mathcal{M} \quad (4.19)$$

$$\tau_i^{kt} \geq r_i \times y_i^{kt} \quad \forall i \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.20)$$

$$\tau_{n+1}^{kt} \leq d_i + T_{max}(1 - y_i^{kt}) \quad \forall i \in \mathcal{N}^+, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.21)$$

$$\tau_0^{kt} \geq r_i \times y_i^{kt} \quad \forall i \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.22)$$

$$\tau_i^{kt} \leq d_i + T_{max}(1 - y_i^{kt}) \quad \forall i \in \mathcal{N}^-, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.23)$$

$$x_{ij}^{kt} \in \{0, 1\} \quad \forall (v_i, v_j) \in \mathcal{A}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.24)$$

$$y_i^{kt} \in \{0, 1\} \quad \forall i \in \mathcal{N}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.25)$$

$$i^{kt} \in \{0, 1\} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.26)$$

$$o^{kt} \in \{0, 1\} \quad \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.27)$$

$$0 \leq \tau_i^{kt} \leq T_{max} \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.28)$$

$$u_i^{kt} \geq 0 \quad \forall v_i \in \mathcal{V} \cup \{v_{n+1}\}, \forall m_k \in \mathcal{M}, \forall t \in \mathcal{H} \quad (4.29)$$

The objective function (4.1) minimizes the total cost. Constraints (4.2) ensure that each request is assigned to exactly one vehicle and one trip. Constraints (4.3) and (4.4) are the flow conservation constraints. Constraints (4.5) and (4.6) ensure that trips start and end at the depot. Constraints (4.7) (resp., constraints (4.8)) ensure i^{kt} (resp., o^{kt}) equals to 1 when at least one inbound (resp., outbound) request is served in the trip t of driver m_k , and equals to 0 otherwise. Constraints (4.9) to (4.13) are flow capacity constraints, where constraints (4.9) ensure that the load of a vehicle starting a trip is equal to the sum of loads of the outbound requests served in the trip, constraints (4.10) and (4.11) ensure vehicle load flow conservation when visiting an inbound (resp., outbound) customer, constraints (4.12) ensure that the load of a vehicle ending a trip is equal to the sum of loads of the inbound requests served in the trip, and constraints (4.13) check that vehicle capacity is never exceeded. Time precedence relations are ensured with constraints (4.14) to (4.19). Constraints (4.14) ensure that the pickup service time T_P is added when leaving the depot if the trip contains at least one outbound request. Constraints (4.15) model travel and service times between two customers while constraints (4.16) model the return to the depot. Constraints (4.17) ensure that the ending time of a trip is later than its starting time. Constraints (4.18) are used to add the delivery service time T_D when returning to the depot if the trip contained at least one inbound request. Constraints (4.19) ensure that the service of a route ends before the end of the time horizon. Release and due date constraints are respected with constraints (4.20) to (4.23). Constraints (4.20) and (4.21) respectively check that an inbound customer is picked up at home after their release date and dropped off at the depot before their due date, while constraints (4.22) and (4.23) check that an outbound customer is picked up at the depot after their release date and dropped off at home before their due date. Finally, constraints (4.24) to (4.29) define the variables.

4.3.2 Measure of equity

To integrate equity within a routing problem, one needs first to determine the equity metric that is the workload to balance fairly between drivers. The commonly used equity metrics can be classified as either constant-sum or variable-sum. Examples of constant-sum equity metrics include the number of customers served, the total customer load served, and the service time. On the other hand, variable-sum metrics focus on the routing cost, which can be measured in terms of travel distance or time. The existing literature primarily emphasizes the consideration of routing cost as the equity metric.

As shown in Matl et al. (2018), to integrate equity within a routing problem, several modeling choices are possible: i) a bi-objective model where two explicit objectives (min-cost and an equity objective) are simultaneously optimized and the Pareto front is generated. ii) weighted-sum: the two objectives are combined within a single objective function with weights $\alpha \times \text{obj}_1 + \beta \times \text{obj}_2$. iii) constraint-based: cost minimization is the single objective and equity is handled with constraints. iv) primary objective: the equity objective is the single objective and cost efficiency is not considered.

Equity is rarely considered as the primary objective because it provides very costly solutions and it mainly concerns service and client-oriented logistics applications where equity is more important than cost. For example, in applications to disaster relief, minimizing the latest arrival of aid (min-max objective) as the single objective is relevant to provide the best possible service (Golden et al., 2014).

In contrast, in provider-oriented logistics applications, both equity and cost are important and must be considered. Generating the complete Pareto front is not trivial and among all Pareto optimal solutions, only a restricted subset of solutions is relevant because as for most of the solutions, one of the objectives is very deteriorated. A weighted-sum model provides solutions with both an attractive cost and equity when coefficients are well chosen but the choice is not trivial and varies depending on the instance.

A common weakness of all these approaches is that the choice of the equity objective is not trivial and there exist no "perfect" equity measures. Indeed, as studied in some recent papers (Halvorsen-Weare and Savelsbergh, 2016; Matl et al., 2018, 2019), each equity function has pros and cons and none satisfies every desirable property for equity.

We believe that the constraint-based model is relevant because minimizing the routing cost remains the main objective of a provider-oriented application and this approach guarantees a good control of equity with constraints.

We recall that $c_{ij} = t_{ij}$ i.e., the routing cost is the driving time (service times are not counted). Let c^k be the routing cost of driver m_k and C the routing cost of the solution ($C = \sum_{m_k \in \mathcal{M}} c^k$). To impose equitable routes, we propose to control for each route the gap between its cost and the average routing cost of the solution. Hence, we impose for each driver a limit deviation above the average routing cost as follows:

$$c^k \leq \frac{\alpha}{K} \times C \quad \forall m_k \in \mathcal{M} \quad (4.30)$$

To add these constraints to the model (4.1)-(4.29), we introduce one new variable and one new constraint per driver. For any driver m_k , the non-negative variable w^k represents the workload of the route of m_k .

$$w^k = \sum_{(v_i, v_j) \in \mathcal{A}} \sum_{t \in \mathcal{H}} c_{ij} \times x_{ij}^{kt} \quad \forall m_k \in \mathcal{M} \quad (4.31)$$

$$w^k \leq \frac{\alpha}{K} \times \sum_{m_{k'} \in \mathcal{M}} w^{k'} \quad \forall m_k \in \mathcal{M} \quad (4.32)$$

$$w^k \geq 0 \quad \forall m_k \in \mathcal{M} \quad (4.33)$$

Constraints (4.31) compute the workload for each driver route while Constraints (4.32) are the equity constraints and Constraints (4.33) define positive variables.

A constraint-based approach is a specific case of the bi-objective model so, the same undesirable consequence can occur: a solution can artificially worsen the cost to satisfy the equity measures (Jozefowicz et al., 2002). A key question about our modeling is to measure the impact of equity constraints on the total cost.

4.4 Solution methods

As seen in Chapter 2, solving a straightforward formulation of a vehicle routing problem to optimally use on-the-shelf MILP solvers is inefficient. The main issue is the weakness of the dual bound provided by

linear relaxation. Hence, we proposed in Chapter 2 a branch-and-price algorithm to solve the MTMPD-RD. In Section 4.4.1, we briefly describe this branch-and-price algorithm (for the MTMPD-RD without equity constraints). The purpose is not to give a complete description of the algorithm, readers are referred to Agius et al. (2022) or Chapter 2 (Section 2.4) for that. Instead, we briefly present the main components of the algorithm, which are the master problem, the labels defined in the pricing problem, the dominance rule, and the branching rules.

This algorithm is based on a set partitioning formulation. Intuitively, one might think that it could be sufficient to integrate the equity constraints (4.30) into the model and solve the new problem with a slight adaptation of the algorithm but it is not. We show in Section 4.4.2 that with this formulation the pricing problem cannot be solved efficiently. Then, in Section 4.4.3, we propose a heuristic that relies on solving successfully a simplified version of the problem by calibrating a parameter. This solution method is based on a dichotomy search principle. Finally, in Section 4.4.4 we propose a new formulation and an adapted branch-and-price algorithm. In this model, columns are indexed both by the route and by the driver.

4.4.1 Branch-and-price algorithm for the MTMPD-RD: a reminder

The MTMPD-RD is formulated as a set partitioning problem (Agius et al., 2022) where Ω denotes the set of feasible routes, c_r is the cost of a route and a_i^r is a binary indicator of requests served in routes:

$$\min C = \sum_{r \in \Omega} c_r \theta_r \quad (4.34)$$

subject to:

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (4.35)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (4.36)$$

$$\theta_r \in \{0, 1\} \quad \forall r \in \Omega \quad (4.37)$$

Set Ω denotes the set of feasible routes. The objective function (4.34) minimizes the total cost. Constraints (4.35) are the set partitioning constraints, they ensure that each request is served. Constraint (4.36) imposes to build at most K routes.

Column generation is used to solve the linear relaxation of this formulation. The master problem is obtained from (4.34)-(4.37) by relaxing constraints (4.37). We denote λ_i the unsigned dual variable associated with Constraint (4.35) for $i \in \mathcal{N}$ and $\lambda_0 \leq 0$ be the dual variable associated with Constraint (4.36). The reduced cost of a route $r \in \Omega$ is:

$$\bar{c}_r = c_r - \left(\sum_{i \in \mathcal{N}} a_i^r \lambda_i \right) - \lambda_0 = \sum_{(v_i, v_j) \in \mathcal{A}: v_j \neq v_0} b_{ij}^r (c_{ij} - \lambda_j) + \left(\sum_{(v_i, v_0) \in \mathcal{A}} b_{i0}^r c_{i0} \right) - \lambda_0 \quad (4.38)$$

with $b_{ij}^r = 1$ if arc (v_i, v_j) is traversed in route r .

The pricing problem finds new routes with negative reduced costs and is solved with a dynamic programming labeling algorithm, where a label L represents a path P starting at depot v_0 and combining several pickup and delivery trips. It is defined by a vector $L = (v, c, t, l, q, \tilde{q}, R = \{R_1, \dots, R_n\})$ of size $n + 6$.

The first three resources are classic, v is the last node visited, c is the current reduced cost of P , and t is the time at which node v is left. The next resources are specific to the MTMPD-RD. Resource l is the

latest time allowed to be back to the depot to satisfy the due dates of inbound requests served in the last trip. Resources q and \tilde{q} manage the capacity constraint in the presence of inbound and outbound requests. Finally, resource R is the reachability vector and is defined with three possible states for each request: (i) $R_i = C$ if v_i is reachable in the current trip (ii) $R_i = N$ if v_i is unreachable in the current trip because of resource constraints but is reachable in the next trip (iii) $R_i = U$ if node v_i cannot be reached anymore in the route.

The labeling algorithm consists in extending labels starting from the depot. When a label is back to the depot, it corresponds to a complete route. Such a route can be added to the master problem (if its reduced cost is negative) or extended again as multi-trip is allowed. Label initialization and extension rules are not described here (see Section 2.5). We simply remind the dominance rule. Given two labels L^1 and L^2 , L^1 dominates L^2 if:

- $v^1 = v^2, c^1 \leq c^2, t^1 \leq t^2, l^1 \geq l^2, q^1 \leq q^2, \tilde{q}^1 \leq \tilde{q}^2$,
- for every $i \in \mathcal{N}$: if $R_i^2 = C$, then $R_i^1 = C$ and if $R_i^2 = N$, then $R_i^1 \in \{C, N\}$

The dominance rule does not apply on labels L^1 and L^2 if they are attached to the depot v_0 such that the previously visited customer is v_i and a branching constraint on a triplet (v_i, v_0, v_j) holds. Column generation is embedded into a branch-and-price algorithm to solve the MTMPD-RD. Branching rules lie in the definition of flows. Given a solution S of $MP(\Omega_t)$, we call *flow* on arc $(v_i, v_j) \in \mathcal{A}$, the sum of variables using this arc in S , we denote it as $f_{ij} = \sum_{r \in \Omega} b_{ij}^r \theta_r$. Similarly, we call *flow* on triplet (v_i, v_0, v_j) , the sum of variables using this triplet in S , we denote it as $f_{i0j} = \sum_{r \in \Omega} b_{i0j}^r \theta_r$.

Then, an optimal solution S of MP is integer if and only if flows on all arcs and triplets are integers. This property is used for branching: if S is not integer then, we branch on an arc or triplet with a fractional flow.

4.4.2 Weakness of the initial model for the MTMPD-RD-WE

Let R denote the initial routing problem without equity concerns (MTMPD-RD) and P the problem that considers equity constraints (MTMPD-RD-WE). As explained in Chapter 3, R can easily be extended to the case of a constant-sum equity constraint by restricting the set of feasible routes Ω , in formulation (4.34)-(4.37), to routes that respect the limit defined on the workload. For a variable-sum equity constraint however, the feasibility of a route depends on the routes that are selected in the solution: Ω cannot be restricted the same way. A natural approach would then be to explicitly introduce in P a constraint limiting route workload:

$$P : \min C(\Theta) = \sum_{r \in \Omega} c_r \theta_r \quad (4.39)$$

subject to:

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (4.40)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (4.41)$$

$$c_r \theta_r \leq \frac{\alpha}{K} \times \sum_{r' \in \Omega} c_{r'} \theta_{r'} \quad \forall r \in \Omega \quad (4.42)$$

$$\theta_r \in \{0, 1\} \quad \forall r \in \Omega \quad (4.43)$$

Let λ_i unsigned, $\lambda_0 \leq 0$ and $\lambda_r \geq 0$ be the dual variables associated respectively with Constraints (4.40), (4.41) and (4.42). The reduced cost of a route r is defined as:

$$\bar{c}_r = c_r - \sum_{i \in \mathcal{N}} a_i^r \lambda_i - \lambda_0 - \frac{\alpha}{K} \sum_{r' \in \Omega} c_{r'} \lambda_{r'} + c_r \lambda_r \quad (4.44)$$

$$= \sum_{(v_i, v_j) \in r} c_{ij} - \sum_{v_i \in r, i \neq 0} \lambda_i - (\lambda_0 + \frac{\alpha}{K} \sum_{r' \in \Omega} c_{r'} \lambda_{r'}) + \sum_{(v_i, v_j) \in r} c_{ij} \lambda_r \quad (4.45)$$

At a given step t of the column generation, when the restricted master problem has been solved, dual variables λ_0 , λ_i and λ_r for $r \in \Omega_t$ are known. Thanks to the complementary slackness theorem, dual variables λ_r for $r \in \Omega \setminus \Omega_t$ are also implicitly equal to 0. As a consequence, the reduced cost of a route in $\Omega \setminus \Omega_t$ is equal to:

$$\sum_{(v_i, v_j) \in r} c_{ij} - \sum_{v_i \in r, i \neq 0} \lambda_i - (\lambda_0 + \frac{\alpha}{K} \sum_{r' \in \Omega} c_{r'} \lambda_{r'}) \quad (4.46)$$

with term $\lambda_0 + \frac{\alpha}{K} \sum_{r' \in \Omega} c_{r'} \lambda_{r'}$ known and independent of route r . As this reduced cost is cumulative on arcs and vertices, it can be used as the objective function of the pricing problem to generate new routes. Resources and resource extension functions would be the same as those presented in Section 4.4.1.

However, equation (4.46) is wrong for routes $r \in \Omega_t$ as it does not take account of term $\sum_{(v_i, v_j) \in r} c_{ij} \lambda_r$. Hence, some routes in Ω_t may wrongly be considered as having a negative reduced cost. These routes may dominate targeted routes and therefore prevent the column generation method to converge. Removing the dominance procedure from the label setting algorithm that solves the pricing problem is possible but would make it very slow. The only alternative approach we are aware of is using efficient completion bounds, which does not seem easy in our setting. As we did not find any other way to overcome this problem, we did not investigate further this formulation.

4.4.3 A Dichotomy-based Heuristic

As explained in Section 4.4.2, the difficulty of solving the model (4.39)-(4.43) lies on the equity constraints (4.42) because the total cost of the solution is unknown while solving the problem. In this section, we propose a simplified version of the problem where the total cost is fixed in equity constraints. Hence, equity constraints can be written as follows:

$$c^k \leq \frac{\alpha}{K} \times Z \quad \forall m_k \in \mathcal{M} \quad (4.47)$$

with Z a parametric fixed cost. We call the original problem with this new constraint $Q(Z)$. In the following, we present some properties linking problems P and $Q(Z)$. Then, based on these properties we propose a heuristic algorithm. The algorithm is a kind of dichotomic search algorithm, at each step, we solve $Q(Z)$ with a fixed value Z , then, according to the results and using the following properties, we guide the search for the next step.

First, let us present the formulation of problem $Q(Z)$. This formulation is derived from the model (4.39)-(4.43) of P with equity constraints (4.42) replaced by their fixed-cost version (4.47):

$$Q(Z) : \min C(\Theta) = \sum_{r \in \Omega} c_r \theta_r \quad (4.48)$$

subject to:

$$\sum_{r \in \Omega} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (4.49)$$

$$\sum_{r \in \Omega} \theta_r \leq K \quad (4.50)$$

$$c_r \theta_r \leq \frac{\alpha}{K} \times Z \quad \forall r \in \Omega \quad (4.51)$$

$$\theta_r \in \{0, 1\} \quad \forall r \in \Omega \quad (4.52)$$

where Ω is the set of all feasible routes for the MTMPD-RD (without equity constraints). Constraints (4.51) can be decomposed by the routes and therefore, integrated into the pricing problem. We denote $\Omega(Z) \subseteq \Omega$, the set of routes r for which constraint (4.51) is respected. Thus, $Q(Z)$ can be reformulated as follows:

$$Q(Z) : \min C(\Theta) = \sum_{r \in \Omega(Z)} c_r \theta_r \quad (4.53)$$

subject to:

$$\sum_{r \in \Omega(Z)} a_i^r \theta_r = 1 \quad \forall i \in \mathcal{N} \quad (4.54)$$

$$\sum_{r \in \Omega(Z)} \theta_r \leq K \quad (4.55)$$

$$\theta_r \in \{0, 1\} \quad \forall r \in \Omega(Z) \quad (4.56)$$

Note that the obtained model (4.53)-(4.56) is more suitable for a branch-and-price approach. Given a value Z , solving problem $Q(Z)$ provides some properties on the feasibility and on optimal solutions of P . In the following, we present some properties that will help us derive an efficient dichotomy-based heuristic.

Properties of $Q(Z)$

Before presenting the needed properties, let us introduce some important notations that will be used in the following:

- Given a problem X , we denote respectively by $\mathcal{F}(X)$ and $\mathcal{O}(X)$ the set of feasible solutions and optimal solutions of X .
- The cost of a solution $\Theta = \{r \in \Omega : \theta_r = 1\}$ is denoted $C(\Theta) = \sum_{r \in \Theta} c_r = \sum_{r \in \Omega} c_r \theta_r$.
- The cost of the longest route of a solution Θ is denoted $\bar{c}(\Theta) = \max_{r \in \Theta} c_r = \max_{r \in \Omega} c_r \theta_r$.
- We denote by Θ^* with a cost $C^* = C(\Theta^*)$ an optimal solution of P and by Θ_Z with a cost $C(Z) = C(\Theta_Z)$ an optimal solution of $Q(Z)$.

- $Z(\Theta^*) = \frac{K}{\alpha} \times \bar{c}(\Theta^*)$.

Figure 4.1 shows an example of curve $C(Z)$ associated with a problem P . The x-axis corresponds to the parameter value Z and the y-axis corresponds to $C(Z)$, the optimal cost of $Q(Z)$. The curve is non-increasing and it can be proven for any problem P . For any $Z_1 \leq Z_2$, equity constraints (4.51) are stronger in $Q(Z_1)$ than in $Q(Z_2)$ so, $C(Z_1) \geq C(Z_2)$. In Figure 4.1, the orange zone (for $Z \leq Z_{inf}$) represents the space for which $Q(Z)$ is infeasible while the green zone (for $Z \geq Z_{+\infty}$) represents the space for which equity constraints are irrelevant because Z is large enough to have $C(Z) = C(+\infty)$ (when equity constraints are relaxed).

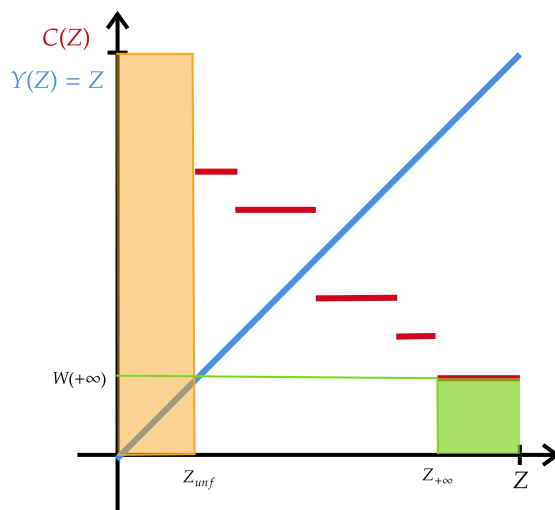


Figure 4.1: Example of curve $C(Z)$

From the definition of $Q(Z)$, we can deduce some properties linking solutions of P and $Q(Z)$. First, it is important to clarify that if P is feasible then, there does not necessarily exist a Z such that an optimal solution of P is optimal for $Q(Z)$. In other words, the optimal cost of P is not necessarily a point of the curve of $C(Z)$.

Indeed, let $\Theta^* \in \mathcal{O}(P)$ then, by definition $\Theta^* \in \mathcal{F}(Q(C^*))$ and for any $\Theta_{C^*} \in \mathcal{O}(Q(C^*))$ of cost $C(C^*)$ we have: $C(C^*) \leq C^*$. Then, two cases can occur either $C(C^*) = C^*$ or $C(C^*) < C^*$.

- Case 1: if $C(C^*) = C^*$ then, Θ_{C^*} is optimal for both $Q(Z)$ and P . In this case, the optimal cost of P is a point of the curve of $C(Z)$ and is found in $Z = C^*$.
- Case 2: if $C(C^*) < C^*$ then, Θ_{C^*} is not feasible for P and the optimal cost of P might not be a point of the curve of $C(Z)$, at least not found for $Z = C^*$.

If P is feasible then, we can state:

$$C^* \geq Z(\Theta^*) = \frac{K}{\alpha} \times \bar{c}(\Theta^*) \quad \forall \Theta^* \in \mathcal{O}(P) \quad (4.57)$$

When solving $Q(Z)$, one might distinguish two cases, either $Q(Z)$ is feasible or is unfeasible. In the following, we distinguish all cases.

1. If $Q(Z)$ is unfeasible then:

$$\bar{c}(\Theta^*) > \frac{\alpha}{K} \times Z, \quad \forall \Theta^* \in \mathcal{O}(P) \quad (4.58)$$

Combining (4.58) with (4.57) gives

$$C^* > Z \quad (4.59)$$

Note that $Z = +\infty$ means P is unfeasible and allows to conclude. Otherwise, all the values lower than Z can be discarded because for any $Z' < Z$ if $Q(Z)$ is unfeasible then, $Q(Z')$ is necessarily unfeasible.

2. If $Q(Z)$ is feasible with $\Theta_Z \in \mathcal{O}(Q(Z))$ then, either Θ_Z is feasible or not for P .

- (a) If $\Theta_Z \in \mathcal{F}(P)$ then, we can state:

$$\bar{c}(\Theta_Z) \leq \frac{\alpha}{K} \times \min(Z; C(Z)) \quad (4.60)$$

and we distinguish two cases according to values Z and $C(Z)$:

- i. If $C(Z) \leq Z$ then:

- A. $C^* \geq C(Z)$.

Proof by contradiction: Assume $C^* < C(Z) \Rightarrow$ (4.61) where the first inequality comes from the absurd hypothesis.

$$\exists \Theta^* \in \mathcal{O}(P) : \frac{\alpha}{K} \times Z < \bar{c}(\Theta^*) \leq \frac{\alpha}{K} \times C(\Theta^*) < \frac{\alpha}{K} \times C(Z) \quad (4.61)$$

This contradicts $C(Z) \leq Z$. \square

- B. Also since $\Theta_Z \in \mathcal{F}(P)$, $C^* \leq C(Z)$.

- C. From A and B we deduce that $C^* = C(Z)$ and we can state that $\Theta_Z \in \mathcal{O}(P)$

- ii. If $C(Z) > Z$ then:

- A. $Z < C^* \leq C(Z)$

proof: First, assume $C^* \leq Z$ then, $\forall r \in \Theta^*$, $c_r \leq \frac{\alpha}{K} \times C^* \leq \frac{\alpha}{K} \times Z$. Then, Θ^* is feasible for $Q(Z)$ and $C^* \leq Z < C(Z)$, this contradicts $Z < C^* \leq C(Z)$.

Second, since $Z < C(Z)$, we have $\forall r \in \Theta'(Z)$, $c_r \leq \frac{\alpha}{K} \times Z < \frac{\alpha}{K} \times C(Z)$. So, $\Theta'(Z)$ is feasible for $Q(C(Z))$ and consequently for P . Hence, $C^* \leq C(Z)$. \square

- B. Since $\Theta_Z \in \mathcal{F}(P)$ and $C(Z)$ is not increasing then, for any $Z' < Z$, either $Q(Z')$ is unfeasible or $C(Z') > C(Z)$. Therefore, all the values lower than Z can be discarded.

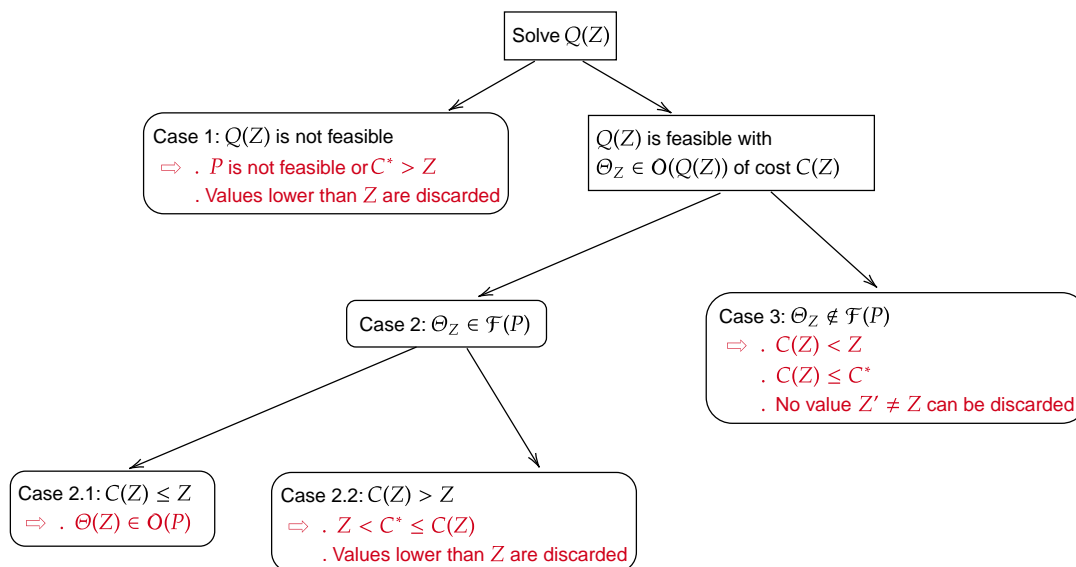
- (b) If $\Theta_Z \notin \mathcal{F}(P)$, we can state that:

$$\frac{\alpha}{K} \times C(Z) < \bar{c}(Z) \leq \frac{\alpha}{K} \times Z \quad (4.62)$$

- i. This implies that $C(Z) < Z$.

- ii. $C^* \geq C(Z)$. Similar proof as case 2.a.i.A.

- iii. We distinguish two cases:

Figure 4.2: $Q(Z)$: Decision tree

A. For $Z' < \frac{K}{\alpha} \times \bar{c}(\Theta_Z)$, we have:

$$\frac{\alpha}{K} \times \frac{K}{\alpha} \times \bar{c}(\Theta_Z) = \bar{c}(\Theta_Z) > \frac{\alpha}{K} \times Z' \quad (4.63)$$

hence, Θ_Z is excluded from $Q(Z')$ so possibly, a $\Theta^* \in \mathcal{O}(P)$ such that $\bar{c}(\Theta^*) < \bar{c}(\Theta_Z)$ might be feasible for $Q(Z')$.

B. For $Z' \geq \frac{K}{\alpha} \times \bar{c}(\Theta_Z)$, all optimal solutions may be such that $\bar{c}(\Theta^*) \geq \bar{c}(\Theta_Z)$, or $\mathcal{O}(P)$ may be empty. Another problem than Q has to be defined, in order to discard Θ_Z .

iv. In this case, we cannot discard any value $Z' \neq Z$.

Figure 4.2 depicts a decision tree summarizing the properties between P and $Q(Z)$.

Branch-and-price for $Q(Z)$

Let us recall that the problem without equity constraints (MTMPD-RD) can be solved using the branch-and-price algorithm described in Chapter 2, Section 2.4 (see 4.4.1 for a brief reminder). In the following, we present the adaptation required to solve problem $Q(Z)$, that is the MTMPD-RD integrating equity constraints with a fixed cost Z (4.47). Model (4.53)-(4.56) is solved for the master problem instead of the model (4.34)-(4.37). The single difference is that the set of feasible routes Ω is reduced to $\Omega(Z)$.

Neither the branching rules nor the column generation scheme are impacted, only the pricing problem requires a slight adaptation. To be feasible, a route cannot exceed the limit cost $c_{\max} = \frac{\alpha}{K} \times Z$. To manage this constraint, we add a new resource w to the label definition corresponding to the workload of the route in terms of route cost. Resource w is different from resource c , the former corresponds to the real route cost while the latter corresponds to the reduced cost of the route.

Given a label L , resource w is initialized to 0 as initially, the route is empty. Then, when L is extended from a node v_i to a node v_j , resource w is updated: $w \leftarrow w + c_{ij}$. To ensure the feasibility of a route, one

needs to check that returning to the depot directly from the current node does not exceed the cost limit: $w + c_{j0} \leq \frac{\alpha}{K} \times Z$. Finally, one condition must be added to the dominance rules: a label L^1 dominates a label L^2 only if $w^1 \leq w^2$.

Heuristic algorithm

Our heuristic relies on solving successively $Q(Z)$ with different values for Z . We start with an initial value Z , solve $Q(Z)$ and according to the obtained result, we determine a new value Z' to evaluate. We use the monotonicity of $Q(Z)$ (non-increasing function of Z) and the properties linking $Q(Z)$ and P to guide the search. The approach is based on a dichotomy search, a search space for Z is bounded in $[L_Z, U_Z]$ and updated at each iteration by updating either L_Z or U_Z . Two simplifications are made to derive a heuristic algorithm:

- Only optimal solutions of $Q(Z)$ are explored. We remind the reader that these solutions are not necessarily feasible for P .
- A single interval $[L_Z, U_Z]$ is considered.

The decision tree of $Q(Z)$ (Figure 4.2) is used to guide the search. Given a value Z , we solve $Q(Z)$ and follow the following rules. These rules are based on the previously announced properties.

- Case 1: $Q(Z)$ is unfeasible then, $[L_Z, U_Z] \leftarrow [Z, U_Z]$
- Case 2: $Q(Z)$ is feasible and the optimal solution found $\Theta_Z \in \mathcal{F}(P)$
 - Case 2.1: If $C(Z) \leq Z$ then, Θ_Z is optimal for P
 - Case 2.2: If $C(Z) > Z$ then, $[L_Z, U_Z] \leftarrow [Z, U_Z]$
- Case 3: $Q(Z)$ is feasible and the optimal solution found $\Theta_Z \notin \mathcal{F}(P)$ then, $[L_Z, U_Z] \leftarrow [L_Z, Z]$

In case 1, if $Q(Z)$ is unfeasible, then it is useless to explore $Z' < Z$ since $Q(Z')$ is necessary unfeasible; hence $L_Z \leftarrow Z$.

In case 2.1, the optimal solution Θ_Z of $Q(Z)$ is also optimal for P which concludes the algorithm.

In case 2.2, the solution Θ_Z is feasible for P , and exploring $Z' < Z$ provides necessary solutions with a higher cost because of the monotonicity of $C(Z)$; hence $L_Z \leftarrow Z$.

Finally, in case 3, the optimal solution found Θ_Z is infeasible for P and $C(Z) < Z$ so, looking for a $Z' > Z$ provides a solution $\Theta_{Z'}$ such that $C(Z') \leq C(Z) < Z < Z'$. Larger values than Z cannot provide feasible solutions for P ; hence $U_Z \leftarrow Z$.

These rules guide the search. We keep solving $Q(Z)$ until we reach a stopping criterion. The possible stopping criterion are i) we prove that an optimal solution Θ_Z for $Q(Z)$ is optimal for P ($C^* = C(\Theta_Z)$) and ii) we close the space search of Z , i.e., $L_Z > U_Z$.

$[L_Z, U_Z]$ is initialized to $[0, F_B]$ with F_B a feasibility bound. A simple feasible bound is $F_B = K \times T_{max}$ corresponding to each driver driving the whole horizon. The interval is updated after each solving of $Q(Z)$. Also, we keep the best current feasible solution of P found into B . Initially, $B \leftarrow +\infty$ meaning that no feasible solution is known. At each step a new Z value is selected following the simple rule to be the middle of the interval $[L_Z, U_Z]$: $Z \leftarrow \frac{L_Z + U_Z}{2}$. The algorithm ends when a stopping criterion is reached: i) we are in case 2.1 which proves optimality or ii) the search space is closed ($U_Z - L_Z < \epsilon$). If the algorithm ends without finding any feasible solution for P , it does not guarantee that P is infeasible as only optimal solutions of $Q(Z)$ are explored. Algorithm 4.1 describes the algorithm.

Algorithm 4.1 Heuristic

```

1:  $[L_Z, U_Z] \leftarrow [0, F_B]$ 
2:  $B \leftarrow +\infty$ 
3:  $opt \leftarrow \text{false}$ 
4: repeat
5:    $Z \leftarrow \frac{L_Z + U_Z}{2}$ 
6:    $\Theta_Z \leftarrow \text{solve } Q(Z)$ 
7:   if  $Q(Z)$  is not feasible then
8:     case 1
9:      $[L_Z, U_Z] \leftarrow [Z, U_Z]$ 
10:  else
11:    if  $\Theta_Z \in \mathcal{F}(P)$  then
12:      if  $C(\Theta_Z) \leq Z$  then
13:        case 2.1
14:         $\Theta'(Z)$  is optimal for  $P$ 
15:         $B \leftarrow C(\Theta_Z)$ 
16:         $opt \leftarrow \text{true}$ 
17:      else
18:        case 2.2 ( $C(\Theta_Z) > Z$ )
19:         $B \leftarrow \min(B, C(\Theta_Z))$ 
20:         $[L_Z, U_Z] \leftarrow [Z, U_Z]$ 
21:      else
22:        case 3 ( $\Theta_Z \notin \mathcal{F}(P)$ )
23:         $[L_Z, U_Z] \leftarrow [L_Z, Z]$ 
24:  until  $opt = \text{true}$  or  $U_Z - L_Z < \epsilon$ 
25:  if  $opt = \text{true}$  then
26:    optimal of  $P$  found ( $C^* = B$ )
27:  else if  $B \neq +\infty$  then
28:    feasible solution of  $P$  found of cost  $B \geq C^*$ 
29:  else
30:    no feasible solution found for  $P$ 

```

4.4.4 A branch-and-price approach based on a driver-indexed model

In the following, we present an exact solution method based on a branch-and-price approach. In this solution method, the problem's initial formulation is updated with new constraints and variables. Columns are indexed by route ID and driver ID. We distinguish the sets of routes between drivers and denote $\Omega_{(m_k)}$ the set of feasible routes of driver m_k . So, given a driver m_k and a route $r \in \Omega_{(m_k)}$, the decision variable θ_r^k equals 1 if the route is selected for this driver, 0 otherwise. The equity is managed in the master problem by adding Constraint (4.67) for each driver. So, the pricing problem looks for feasible routes for the MTMPD-RD (*i.e.*, without regard to the equity constraint). This solution method requires adaptations at each level of the branch-and-price algorithm: i) the master problem with new constraints and variables, ii) the column generation algorithm with different pricing problems solved for drivers iii) the pricing problem with a new reduced cost formula and iv) new branching rules. These adaptations are presented in the following.

Master problem

The integer master problem is defined as follows:

$$\min \sum_{m_k \in \mathcal{M}} \sum_{r \in \Omega_{(m_k)}} c_r \theta_r^k \quad (4.64)$$

subject to

$$\sum_{m_k \in \mathcal{M}} \sum_{r \in \Omega_{(m_k)}} a_i^r \theta_r^k = 1 \quad \forall i \in \mathcal{N} \quad (4.65)$$

$$\sum_{r \in \Omega_{(m_k)}} \theta_r^k \leq 1 \quad \forall m_k \in \mathcal{M} \quad (4.66)$$

$$\sum_{r \in \Omega_{(m_k)}} c_r \theta_r^k \leq \frac{\alpha}{K} \times \sum_{m_l \in \mathcal{M}} \sum_{r \in \Omega_{(m_l)}} c_r \theta_r^l \quad \forall m_k \in \mathcal{M} \quad (4.67)$$

$$\theta_r^k \in \{0, 1\} \quad \forall m_k \in \mathcal{M}, \forall r \in \Omega_{(m_k)} \quad (4.68)$$

In this model, Constraints (4.65) are similar to (4.40) and ensure that each customer is served exactly once. Constraints (4.66) replace (4.41) and ensure that at most one route is assigned to each driver. Constraints (4.67) are the new equity constraints.

The master problem (MP) is obtained by the continuous relaxation of (4.64)-(4.68).

Column generation

In formulation (4.64)-(4.68), let $\lambda_i, \forall i \in \mathcal{N}$ the unsigned dual variables associated with constraints (4.65), $\mu^k \leq 0, \forall m_k \in \mathcal{M}$ the dual variables associated with Constraints (4.66) and $\gamma^k \leq 0, \forall m_k \in \mathcal{M}$ the dual variables associated with Constraints (4.67).

Because of the dual variables associated with the equity constraints (4.67) a route can have different reduced costs according to the associated driver. Hence columns are indexed by routes and drivers. A consequence is that reduced costs must be computed differently between drivers and at each iteration of the column generation, one subproblem per driver is run. The routes computed for driver m_k are added as new variables in the restricted master problem (θ_r^k) and are effective only for the associated driver.

Pricing problem

In order to support the writing of the dual of MP, we rewrite Constraints (4.67) into Constraints (4.69)

$$\left(1 - \frac{\alpha}{K}\right) \times \sum_{r \in \Omega(m_k)} c_r \theta_r^k - \frac{\alpha}{K} \times \sum_{m_l \in \mathcal{M} \setminus \{m_k\}} \sum_{r \in \Omega(m_l)} c_r \theta_r^l \leq 0 \quad \forall m_k \in \mathcal{M} \quad (4.69)$$

The dual of MP is expressed as follows:

$$\max \sum_{i \in \mathcal{N}} \lambda_i + \sum_{m_k \in \mathcal{M}} \mu^k \quad (4.70)$$

subject to

$$\sum_{i \in \mathcal{N}} a_i^r \lambda_i + \mu^k + \left(1 - \frac{\alpha}{K}\right) c_r \gamma^k - \frac{\alpha}{K} c_r \sum_{m_l \in \mathcal{M} \setminus \{m_k\}} \gamma^l \leq c_r \quad \forall m_k \in \mathcal{M}, \forall r \in \Omega(m_k) \quad (4.71)$$

$$\lambda_i \text{ unsigned} \quad \forall i \in \mathcal{N} \quad (4.72)$$

$$\mu^k \leq 0 \quad \forall m_k \in \mathcal{M} \quad (4.73)$$

$$\gamma^k \leq 0 \quad \forall m_k \in \mathcal{M} \quad (4.74)$$

From (4.70)-(4.74) we deduce the reduced cost of a route $r \in \Omega(m_k)$ for each driver $m_k \in \mathcal{M}$:

$$\bar{c}_r^k = c_r - \sum_{i \in \mathcal{N}} a_i^r \lambda_i - \mu^k - \left(1 - \frac{\alpha}{K}\right) c_r \gamma^k + \frac{\alpha}{K} c_r \sum_{m_l \in \mathcal{M} \setminus \{m_k\}} \gamma^l \quad (4.75)$$

$$= - \sum_{i \in \mathcal{N}} a_i^r \lambda_i - \mu^k + c_r \left[1 - \left(1 - \frac{\alpha}{K}\right) \gamma^k + \frac{\alpha}{K} \sum_{m_l \in \mathcal{M} \setminus \{m_k\}} \gamma^l\right] \quad (4.76)$$

We denote

$$\delta^k = 1 - \left(1 - \frac{\alpha}{K}\right) \gamma^k + \frac{\alpha}{K} \sum_{m_l \in \mathcal{M} \setminus \{m_k\}} \gamma^l. \quad (4.77)$$

The binary indicator b_{ij}^r is set to one if the arc (v_i, v_j) is traversed by the route r , and to zero otherwise.

The reduced cost is spread over arcs as follows:

$$\bar{c}_r^k = - \sum_{i \in \mathcal{N}} a_i^r \lambda_i - \mu^k + c_r \delta^k \quad (4.78)$$

$$= -\mu^k + \sum_{(v_i, v_j) \in \mathcal{A}, v_j \neq v_0} b_{ij}^r (c_{ij} \delta^k - \lambda_j) + \sum_{(v_i, v_0) \in \mathcal{A}} b_{i0}^r c_{i0} \delta^k \quad (4.79)$$

When solving the pricing problem of driver m_k , the reduced cost of a route r , \bar{c}_r^k is initialized to $-\mu^k$. Then, while extending a label through arc (v_i, v_j) with $v_j \neq v_0$, value $c_{ij} \delta^k - \lambda_j$ is added to the reduced cost and through arc (v_i, v_0) , value $c_{i0} \delta^k$ is added.

Branching rules

For the initial problem without equity constraints, the MTMPD-RD, branching rules of the branch-and-price algorithm are presented in Section 2.4.3. Proposition 1 states that an optimal solution S of the master problem is integer if and only if flows on all arcs and triplets are integer. For the MTMPD-RD-WE, this proposition is necessary but not sufficient because drivers are different and any arc, triplet, or node cannot be shared by several drivers.

To deal with this issue, we propose to compute flows separately for each driver and update the branching rules. Thus, we denote by $f_{ij}^k = \sum_{r^k \in \Omega(m_k)} b_{ij}^{r^k} \theta_r^k$ the flow on arc (v_i, v_j) for driver m_k . Similarly, we denote by $f_{i0j}^k = \sum_{r^k \in \Omega(m_k)} b_{i0j}^{r^k} \theta_r^k$ the flow on triplet (v_i, v_0, v_j) for driver m_k .

Based on these new definitions, the Proposition 1 on branching rules becomes: an optimal solution S of MP is integer if and only if flows on all arcs and triplets are integer for all drivers.

Based on this property, we propose a new branching scheme. If a solution S is not integer and there exists at least one arc (v_i, v_j) and one driver m_k with a fractional flow f_{ij}^k then, we derive two branches: in the first branch, arc (v_i, v_j) is forbidden for driver m_k in the solution (i.e., $f_{ij}^k = 0$) and allowed for other drivers; in the second branch, arc (v_i, v_j) is enforced for driver m_k (i.e., $f_{ij}^k = 1$) while servicing both v_i and v_j is forbidden for other drivers (i.e., $f_{il}^p = f_{li}^p = f_{jl}^p = f_{lj}^p = 0$ for all $m_p \neq m_k$ and all $v_l \in \mathcal{V}$). Note that any route that covers v_i and v_j through any arc has to be forbidden for any driver; except (v_i, v_j) for m_k of course. Otherwise, Constraints (4.65) in MP for i and j could be satisfied without using (v_i, v_j) with m_k .

If S is not integer but all arcs have an integer flow then, there exists a fractional flow on a triplet for a driver. Let (v_i, v_0, v_j) be such a triplet for driver m_k . We derive two branches: in the first branch, the triplet (v_i, v_0, v_j) is forbidden for m_k in the solution (i.e., $f_{i0j}^k = 0$) and allowed for other drivers; in the second branch, the triplet (v_i, v_0, v_j) is enforced for m_k (i.e., $f_{i0j}^k = 1$) while servicing both v_i and v_j is forbidden for other drivers (i.e., $f_{il}^p = f_{li}^p = f_{jl}^p = f_{lj}^p = 0$ for all $m_p \neq m_k$ and all $v_l \in \mathcal{V}$). Again, in this case, it is not sufficient to forbid only the use of triplet (v_i, v_0, v_j) for drivers $m_p \neq m_k$.

Let's enumerate the different cases and the implications they have both on the master problem and the pricing problem:

1. To forbid the use of an arc (v_i, v_j) for a driver m_k :
 - (a) In the master problem: routes r of driver m_k using arc (v_i, v_j) are removed from the master problem (i.e., $\theta_r^k = 0$).
 - (b) In the pricing problem of driver m_k : arc (v_i, v_j) is removed from the graph.
2. To enforce the use of arc (v_i, v_j) for driver m_k and therefore forbid the service of v_i and v_j for other drivers:
 - (a) Enforce the use of (v_i, v_j) for driver m_k :
 - i. In the master problem: routes r of driver m_k not using arc (v_i, v_j) (either they serve v_i , v_j , both or none of them) are removed from the master problem (i.e., $\theta_r^k = 0$).
 - ii. In the pricing problem of driver m_k : arcs (v_i, v_l) with $v_l \neq v_j$ and (v_l, v_j) with $v_l \neq v_i$ are removed from the graph.
 - (b) Forbid the service of v_i (iff $i \neq 0$) and v_j (iff $j \neq 0$) for any driver $m_p \neq m_k$:
 - i. In the master problem: for all drivers $m_p \neq m_k$, routes r servicing either v_i (if $i \neq 0$) or v_j (if $j \neq 0$) are removed from the master problem ($\theta_r^p = 0$).

-
- ii. In the pricing problem of driver m_p : arcs (v_i, v_l) , (v_l, v_i) (if $i \neq 0$); (v_j, v_l) , (v_l, v_j) (if $j \neq 0$); for any $v_l \in V$ are removed from the graph.
3. To forbid the use of triplet (v_i, v_0, v_j) for driver m_k :
 - (a) In the master problem: routes r of driver m_k using this triplet are removed from the master problem ($\theta_r^k = 0$).
 - (b) In the pricing problem of driver m_k : a label that finishes with (v_i, v_0) is never extended to v_j . This constraint has some consequences on the dominance as explained bellow.
 4. To enforce the use of triplet (v_i, v_0, v_j) for driver m_k and forbid the service of v_i and v_j for other drivers:
 - (a) Enforce the use of (v_i, v_0, v_j) for driver m_k :
 - i. In the master problem: routes r of driver m_k not using triplet (v_i, v_0, v_j) (either they serve v_i, v_j , both or none of them) are removed from the master problem ($\theta_r^k = 0$).
 - ii. In the pricing problem of driver m_k : arcs (v_i, v_l) with $v_l \neq v_0$ and (v_l, v_j) with $v_l \neq v_0$ are removed. Also, during the execution of the labeling algorithm, all triplets (v_i, v_0, v_l) with $v_l \neq v_j$ and (v_l, v_0, v_j) with $v_l \neq v_i$ are forbidden. To forbid a triplet, a label that finishes with the first arc of the triplet cannot be extended using the second arc. Again, it impacts the dominance rule. Also, a route finishing with arc (v_i, v_0) cannot be added to the master problem even if it has a negative reduced cost, the corresponding label must be extended to v_j to enforce using the triplet.
 - (b) Forbid the service of v_i and v_j for any driver $m_p \neq m_k$:
 - i. In the master problem: for all drivers $m_p \neq m_k$, routes r serving either v_i or v_j are removed from the master problem ($\theta_r^p = 0$).
 - ii. In the pricing problem of driver m_p : arcs (v_i, v_l) , (v_l, v_i) , (v_j, v_l) , (v_l, v_j) for any v_l are removed from the graph.

Enforcing the service of v_i and v_j either through arc (v_i, v_j) or triplet (v_i, v_0, v_j) for a given driver m_k weakens the dominance rules in the pricing problem of driver m_k . Let us illustrate this through a simple example. Assume a branching rule enforcing the use arc (v_i, v_j) with driver m_k and let L^1 and L^2 be two labels attached to the same node v_l such that L^1 does not use (v_i, v_j) yet while L^2 does. Then, assume that L^1 dominates L^2 in the sense defined in the MTMPD-RD (see section 2.5.4). In this case, it is not sufficient to conclude that L^1 dominates L^2 because L^2 already served the required arc (v_i, v_j) while L^1 did not.

To handle this issue, we add a new resource T to the label definition that corresponds to the set of vertices that must be serviced in the route. Note that if either arc (v_i, v_j) or triplet (v_i, v_0, v_j) must be served then, only vertex v_i needs to be added to T as the branching constraint ensures that v_j (resp. v_0 and v_j) will be visited directly after v_i . Resource T is initialized with all v_i of a branching constraint (either on (v_i, v_j) or (v_i, v_0, v_j)) enforcing to use it for driver m_k . Then, when a vertex $v_i \in T$ is visited, T is updated: $T \leftarrow T \setminus \{v_i\}$. To be valid, a route must be associated with a label located at the depot with $T = \emptyset$. To dominate L^2 , L^1 must serve all imposed vertices in T that are already served by L^2 . Hence, the dominance rules are updated as follows. L^1 dominates L^2 if:

- $v^1 = v^2, c^1 \leq c^2, t^1 \leq t^2, l^1 \geq l^2, q^1 \leq q^2, \tilde{q}^1 \leq \tilde{q}^2,$

- for every $i \in \mathcal{N}$: if $R_i^2 = C$, then $R_i^1 = C$ and if $R_i^2 = N$ then, $R_i^1 \in \{C, N\}$
- $T^1 \subseteq T^2$

The dominance rule is not applied in a very specific situation, because branching constraints on triplets are non-robust about the pricing problem. Let us consider that a constraint on a triplet (v_i, v_0, v_j) is active at the current branch-and-price node. Let us also consider two labels L^1 and L^2 , both attached to the depot and such that L^1 originates from v_i . Then, L^1 is subject to a constraint that does not apply to L^2 and is thus more restricted in its extensions: L^1 cannot dominate L^2 .

4.5 Computational experiments

In this section, we describe the conducted computational experiments to evaluate our solution methods. Section 4.5.1 describes the benchmark of instances used for the experiments while Section 4.5.2 presents and discusses the obtained results.

All algorithms are implemented in the C++ programming language and Linear Programs are solved using the commercial solver IBM CPLEX 12.9. Experiments are executed on a computer with a 3.6 GHz CPU and 32 GB of RAM running on Windows 10. All computational times are given in seconds. From now, let's call Dicho-Heu the heuristic algorithm and DI-BP the driver-indexed branch-and-price algorithm.

4.5.1 Benchmark instances

As the MTMPD-RD-WE is an extension of the MTMPD-RD (adding equity constraints), benchmark instances of the MTMPD-RD are used to evaluate the MTMPD-RD-WE. In Chapter 2, we built a set of instances for the MTMPD-RD. These instances are reused in this chapter. The following briefly recalls how the instances were generated for the MTMPD-RD (see Section 2.6.2 for a more detailed description). Then, we explain which instances were kept and how new instances are derived from these instances (reducing their size).

Instances for the MTMPD-RD: a reminder

Benchmark instances for the MTMPD-RD were generated using actual data from the city of Aix-en-Provence, France. The depot is the Public Hospital of the city and the demands are unitary: they correspond to the transportation of single patients from or to this hospital. The road network of the city is extracted from OpenStreetMap and represented by a graph. The travel times are expressed in minutes and rounded up to integer values.

To generate an instance of n customers, n nodes are selected randomly in the road network graph. The time horizon is set to $[0min., 240min.]$ corresponding to a working period of 4 hours. $T_P = T_D = 3$ minutes. The release and due dates of customer i (r_i and d_i) are generated within the time horizon ensuring feasibility and such that the maximal ride time allowed is the time of a direct ride between pickup and delivery locations plus a detour of 20 minutes.

An instance is characterized by five values: n , t , p , i , and K which correspond respectively to the number of customers, the type of vehicle (s for sedan vehicles of capacity 2 or m for minivans of capacity 4), the probability that a request is inbound (in percentage), the instance ID and the number of vehicles. So, an instance is named $I_{n/t/p/i(K)}$.

First, we generated 15 instances of $n = 50$ customers: 5 with $p = 25\%$, 5 with $p = 50\%$ and 5 with $p = 75\%$. These instances are duplicated for sedan vehicles and minivans and for both, the fleet size K is

fixed to the minimum number of needed vehicles to ensure feasibility. This gives us a total of 30 instances with 50 customers. Then, 50-customers instances are combined to build 30 instances with 100 customers.

Instances and experiment design for the MTMPD-RD-WE

To evaluate the MTMPD-RD-WE, both algorithms are not efficient on 100-customers instances so, we only keep 50-customers instances. Also, the BP algorithm showed its limits. Hence, we create new instances composed of 10 and 25 customers.

The 10 (resp. 25) -customers instances are obtained from the 50-customers instances by keeping only the 10 (resp. 25) first customers. As for the 50-customers instances, the fleet size K is fixed to the minimum number of needed vehicles to ensure feasibility (without equity constraints) for the 25-customers instances. For the 10-customers, one vehicle is added to this setting otherwise most instances become infeasible when adding equity constraints.

For the 10- and 25-customers instances, we only keep instances with $p = 50\%$ with the sedan version. This gives a total of 5 instances with 10 customers and 5 instances with 25 customers. For the 50-customers instances, we only keep instances with $p = 50\%$ (both the sedan and minivan versions).

To sum-up, we have a total of 20 instances divided into 3 sets:

- S_{10} : 5 instances of 10 customers with sedan vehicles and a probability of 50% for requests to be inbound.
- S_{25} : 5 instances of 25 customers with sedan vehicles and a probability of 50% for requests to be inbound.
- S_{50} : 10 instances of 50 customers divided into 2 subsets:
 - $S_{50/s}$: 5 instances of sedan vehicles and a probability of 50% for requests to be inbound.
 - $S_{50/m}$: 5 instances of minivans and a probability of 50% for requests to be inbound.

The i^{th} instance (composed of K vehicles) of subset S_n for vehicle type t and probability p of inbound requests is named $I_{n/t/p/i(K)}$. So, the instances tested are named:

- Instances of set S_{10} : $I_{10/s/50/i(K_i)}$ for $i \in \{1, \dots, 5\}$
- Instances of set S_{25} : $I_{25/s/50/i(K_i)}$ for $i \in \{1, \dots, 5\}$
- Instances of set $S_{50/s}$: $I_{50/s/50/i(K_i)}$ for $i \in \{1, \dots, 5\}$
- Instances of set $S_{50/m}$: $I_{50/m/50/i(K_i)}$ for $i \in \{1, \dots, 5\}$

For all 10- and 25-customers instances, both Dicho-Heu and DI-BP are run with the following parameter values: $\alpha \in \{1.1, 1.08, 1.06, 1.04, 1.02, 1.01\}$. For the 50-customers instances, only Dicho-Heu is run using the same parameter values. Also, each instance is solved without equity constraints (MTMPD-RD) and is used as a reference for comparison.

The DI-BP algorithm is an exact solution method so if it is proven that an instance is infeasible for a given α then, this instance is necessarily infeasible and not tested for $\alpha' < \alpha$. The Dicho-Heu algorithm being heuristic, might not find a feasible solution for a given α but it does not guarantee that the instance is infeasible. Thus, the tests with Dicho-Heu are conducted for all α values.

For the DI-BP algorithm, the total running time is limited to 7200 seconds. For the Dicho-Heu algorithm, neither the total running time nor the number of steps is limited but at each step solving the

parameterized problem is limited to 7200 seconds and if solving this problem reaches the time limit then, the algorithm is stopped to avoid too long computing time.

4.5.2 Results and analysis

Results on the 10- and 25-customers instances

Tables 4.2 and 4.3 provide the results on the small-size instances (10 and 25 customers). Column “Instance” states the instance name; in column “No equity”, sub-columns “cost” and “time” state respectively, the solution cost, and the running time to solve optimally the instance without equity constraints; “ α ” states the value of the parameter α in the equity constraints; in column “Dicho-Heu”, sub-columns “status”, “cost”, “cost incr.” and “time” state respectively, the solution status, the solution cost, the cost increase compared to the cost of the non-constrained problem (in subcolumn “cost” of column “No equity”), and the running time to solve the instance with Dicho-Heu; similarly, in column “DI-BP”, sub-columns “status”, “cost”, “cost incr.” and “time” states respectively, the solution status, the solution cost, the cost increase compared to the cost of the non-constrained problem, and the running time to solve the instance with Dicho-Heu. Cost increases are expressed in percentage and running times in seconds. Possible statuses are “opt.” when optimality is proven, “feas.” when a feasible solution is found without proof of optimality, “not found” when no feasible solution is found, and “infes.” when infeasibility is proven. To simplify the lecture, we call instance, the combination of an instance with a parameter value α .

On the 10-customers instances (Table 4.2), the Dicho-Heu finds feasible solutions for only 6 instances out of 30 and proves optimality for 5 of them, no feasible solution is found for the remaining 24 instances. The running times are very short (less than a second). The DI-BP closes all instances: optimal solutions are found for 20 instances and infeasibility is proven for the remaining 10 instances. The running times don’t exceed a few hundred seconds. The cost increases can be very important, up to 32.4% for instance $I_{10/s/50/2(5)}$ with $\alpha = 1.04$, even for a large α (e.g., 14.97% for $I_{10/s/50/1(4)}$ with $\alpha = 1.1$ or 21.60% for $I_{10/s/50/2(5)}$ with $\alpha = 1.1$).

On the 25-customers instances (Table 4.3), the Dicho-Heu finds feasible solutions for 23 instances out of 30, and proves optimality for 16 of them, no feasible solution is found for the remaining 7 instances. The running times on these instances remain low (a few seconds). The DI-BP finds 21 feasible solutions and 8 of them are proven to be optimal. The algorithm cannot conclude on the feasibility of the remaining 9 instances within two hours of running time. Also, we can see that for a given instance, the tighter the α , the longer the running time. This highlights the performance limits of DI-BP to solve middle-sized instances of the MTMPD-RD-WE. For instances with a feasible solution, we can see that the cost increases are low, always under 5.00% with the best feasible solution found.

Results on the 50-customers instances

Tables 4.4 and 4.5 provide results of the Dicho-Heu algorithm on the 50-customers instances.

Column “Instance” states the instance name; in column “No equity”, sub-columns “cost” and “time” state respectively, the solution cost, and the running time to solve the instance to optimality without equity constraints; “ α ” states the value of parameter α for equity constraints; in column “Dicho-Heu”, sub-columns “status”, “cost”, “cost incr.”, “time” and “#steps” state respectively, the solution status, the solution cost, the cost increase compared to the cost of the non-constrained problem, the running time and the number of steps (number of parameterized problems $Q(Z)$ solved) to solve the instance with Dicho-Heu.

Table 4.2: Instances of set S_{10}

Instance	No equity		α	Dicho-Heu				DI-BP			
	cost	time		status	cost	cost incr.	time	status	cost	cost incr.	time
$I_{10/s/50/1(4)}$	250	<0.1	1.1	not found	-	-	0.2	opt.	294	14.97	29.5
			1.08	not found	-	-	0.1	opt.	294	14.97	22.7
			1.06	not found	-	-	0.1	opt.	302	17.22	22.1
			1.04	not found	-	-	0.1	opt.	308	18.83	22.0
			1.02	not found	-	-	0.1	opt.	315	20.63	34.4
			1.01	not found	-	-	0.1	opt.	321	22.12	53.3
$I_{10/s/50/2(5)}$	225	<0.1	1.1	not found	-	-	0.1	opt.	287	21.60	83.5
			1.08	not found	-	-	0.1	opt.	288	21.88	69.4
			1.06	not found	-	-	0.1	opt.	297	24.24	106.1
			1.04	not found	-	-	0.1	opt.	335	32.84	361.2
			1.02	not found	-	-	0.1	infeas.	-	-	340.5
			1.01	not found	-	-	0.1	infeas.	-	-	-
$I_{10/s/50/3(4)}$	245	<0.1	1.1	opt.	245	0.00	0.1	opt.	245	0.00	0.1
			1.08	opt.	245	0.00	0.1	opt.	245	0.00	0.1
			1.06	opt.	245	0.00	0.1	opt.	245	0.00	0.1
			1.04	not found	-	-	0.2	opt.	254	3.54	3.1
			1.02	not found	-	-	0.2	opt.	254	3.54	2.6
			1.01	not found	-	-	0.2	opt.	254	3.54	2.5
$I_{10/s/50/4(4)}$	309	<0.1	1.1	feas.	314	1.59	0.2	opt.	314	1.59	4.9
			1.08	opt.	314	1.59	<0.1	opt.	314	1.59	4.2
			1.06	opt.	314	1.59	<0.1	opt.	314	1.59	4.0
			1.04	not found	-	-	<0.1	infeas.	-	-	143.8
			1.02	not found	-	-	<0.1	infeas.	-	-	-
			1.01	not found	-	-	<0.1	infeas.	-	-	-
$I_{10/s/50/5(5)}$	320	<0.1	1.1	not found	-	-	<0.1	opt.	355	9.86	26.2
			1.08	not found	-	-	<0.1	infeas.	-	-	106.5
			1.06	not found	-	-	<0.1	infeas.	-	-	-
			1.04	not found	-	-	<0.1	infeas.	-	-	-
			1.02	not found	-	-	<0.1	infeas.	-	-	-
			1.01	not found	-	-	<0.1	infeas.	-	-	-

Table 4.3: Instances of set S_{25}

Instance	No equity		α	Dicho-Heu				DI-BP			
	cost	time		status	cost	cost incr.	time	status	cost	cost incr.	time
$I_{25/s/50/1(5)}$	553	<0.1	1.1	opt.	557	0.72	0.9	opt.	557	0.72	87.3
			1.08	opt.	560	1.25	0.5	feas.	560	1.25	-
			1.06	opt.	560	1.25	0.5	feas.	560	1.25	-
			1.04	opt.	566	2.30	0.9	feas.	566	2.30	-
			1.02	opt.	566	2.30	0.8	feas.	566	2.30	-
			1.01	opt.	566	2.30	1.0	feas.	566	2.30	-
$I_{25/s/50/2(5)}$	511	<0.1	1.1	opt.	511	0.00	0.1	opt.	511	0.00	38.5
			1.08	feas.	523	2.29	1.8	opt.	514	0.58	315.5
			1.06	opt.	523	2.29	1.4	feas.	523	2.29	-
			1.04	opt.	527	3.04	1.3	feas.	558	8.42	-
			1.02	not found	-	-	3.0	not found	-	-	-
			1.01	not found	-	-	2.5	not found	-	-	-
$I_{25/s/50/3(5)}$	531	0.1	1.1	opt.	531	0.00	0.2	opt.	531	0.00	185.2
			1.08	opt.	531	0.00	1.6	opt.	531	0.00	107.6
			1.06	feas.	543	2.21	3.1	opt.	536	0.93	2702.9
			1.04	opt.	543	2.21	2.5	feas.	543	2.21	-
			1.02	not found	-	-	3.8	not found	-	-	-
			1.01	not found	-	-	3.5	not found	-	-	-
$I_{25/s/50/4(5)}$	623	<0.1	1.1	feas.	649	4.01	2.1	opt.	636	2.04	1305.1
			1.08	feas.	649	4.01	1.9	opt.	644	3.26	4673.2
			1.06	feas.	652	4.45	2.3	feas.	652	4.45	-
			1.04	opt.	652	4.45	0.7	feas.	652	4.45	-
			1.02	not found	-	-	2.3	feas.	706	4.45	-
			1.01	not found	-	-	2.1	not found	-	-	-
$I_{25/s/50/5(5)}$	527	<0.1	1.1	opt.	534	1.31	1.0	feas.	534	1.31	-
			1.08	opt.	534	1.31	1.1	feas.	534	1.31	-
			1.06	feas.	541	2.58	2.5	not found	-	-	-
			1.04	feas.	549	4.01	2.7	not found	-	-	-
			1.02	opt.	549	4.01	1.1	not found	-	-	-
			1.01	not found	-	-	2.5	not found	-	-	-

For the sedan instances (Table 4.4), optimality is proven for 16 instances out of 30, and 20 feasible solutions are found. Hence, no feasible solution was found for the remaining 10 instances. When feasible solutions are found, we can see that the impact on cost is low (between 0.00% and 3.75%). However, the running time is drastically increased: all instances are solved within a few seconds without equity against hundreds or thousands of seconds with equity. This is due to the fact that solving an instance with Dicho-Heu relies on solving several branch-and-price (BP) and at each BP, a new resource is managed which weakens the dominance in the pricing problem. In some instances, there might be a step that is not solved to optimality within two hours. This also explains the long running time of some instances (for which the algorithm is stopped before the end).

Table 4.5 shows similar results for the minivan instances, optimality is proven for 16 instances out of 30 and 20 feasible solutions are found. Hence, no feasible solution was found for 10 instances. Again, the costs are not drastically deteriorated (between 0.00% and 2.93%).

Discussion of the results

Results of Tables 4.2, 4.3, 4.4 and 4.5 highlight the pros and cons of each solution method:

- Dicho-Heu: very fast to solve small instances (10 customers) but doesn't find feasible solutions for most of the instances. This method is efficient on 25-customers instances finding a feasible solution for most instances within a short computing time. It is also efficient on the 50-customers instances finding a feasible solution for most instances. However, the running times are largely impacted which shows the limits of the algorithm.
- DI-BP: is fast to solve the 10-customers instances and allows to conclude on feasibility for any instance. It is drastically impacted in terms of running time to solve the 25-customers instances and unable to solve the 50-customers instances. The efficiency is also largely impacted by the parameter α . Indeed, given an instance whatever the value of α , the root lower bound remains usually the same because equity constraints are relaxed. However, the optimal cost is increased when α increases which increases the gap between the root lower bound and the optimal cost.

The obtained results allow us to conclude on the quality of the solutions. We can notice that the impact of equity on the cost directly depends on the size of the instance. On small instances (10 customers) cost is drastically increased to ensure the satisfaction of equity constraints (up to 32.84% on instance $I_{10/s/50/2(5)}$). However, on the 25 (resp., 50)-customers instances, the cost increase never exceeds 5% (resp., 4%).

4.6 Conclusion

Interest in equity in the routing literature has been recently growing considerably. Most studies consider bi-objective models with two explicit objectives, one for cost minimization and one for equity. In this chapter, we rather prefer to keep minimizing the cost as the single objective and manage equity with constraints. These constraints aim at limiting the deviation of route costs from the average route cost. The equity metric is the route cost, a variable-sum metric. We use the MTMPD-RD (the non-emergency transportation problem studied in chapter 2) as a case study problem. We call MTMPD-RD with workload equity (MTMPD-RD-WE) this extension.

We show that the problem cannot be solved efficiently by the standard branch-and-price approach with the straightforward set partitioning formulation and propose two original solution methods: i) a

Table 4.4: Instances of set $S_{50/s}$

Instance	No equity		α	Dicho-Heu				
	cost	time		status	cost	cost incr.	time	#steps
$I_{50/s/50/1(7)}$	910	10.3	1.1	opt.	911	0.11	242.6	8
			1.08	opt.	911	0.11	141.8	8
			1.06	feas.	912	0.22	318.2	15
			1.04	opt.	913	0.33	652.1	10
			1.02	opt.	924	1.52	1326.5	8
			1.01	not found	-	-	4944.6	15
$I_{50/s/50/2(8)}$	771	0.9	1.1	feas.	786	1.91	169.7	15
			1.08	opt.	786	1.91	88.3	8
			1.06	opt.	791	2.53	343.2	7
			1.04	feas.	801	3.75	1647.4	15
			1.02	opt.	801	3.75	319.1	6
			1.01	not found	-	-	872.2	15
$I_{50/s/50/3(9)}$	1047	2.6	1.1	opt.	1061	1.32	54.5	7
			1.08	opt.	1062	1.41	67.3	7
			1.06	feas.	1072	2.33	223.5	15
			1.04	not found	-	-	7200.7	1
			1.02	not found	-	-	7254.9	7
			1.01	not found	-	-	7245.5	6
$I_{50/s/50/4(7)}$	976	5.6	1.1	opt.	976	0.00	119.5	7
			1.08	not found	-	-	7237.9	4
			1.06	opt.	982	0.61	177.1	10
			1.04	not found	-	-	539.9	15
			1.02	not found	-	-	7252.3	5
			1.01	not found	-	-	7295.7	7
$I_{50/s/50/5(7)}$	884	2.7	1.1	opt.	884	0.00	88.5	10
			1.08	opt.	884	0.00	84.3	8
			1.06	opt.	884	0.00	86.2	8
			1.04	opt.	884	0.00	111.6	8
			1.02	opt.	898	1.56	200.4	11
			1.01	not found	-	-	315.1	15

Table 4.5: Instances of set $S_{50/m}$

Instance	No equity		α	Dicho-Heu				
	cost	time		status	cost	cost incr.	time	#steps
$I_{50/m/50/1(7)}$	877	1.8	1.1	opt.	878	0.11	98.0	12
			1.08	opt.	878	0.11	69.1	7
			1.06	opt.	879	0.23	65.7	7
			1.04	opt.	885	0.90	89.0	8
			1.02	opt.	888	1.24	117.7	9
			1.01	not found	-	-	235.1	15
$I_{50/m/50/2(7)}$	745	2.8	1.1	opt.	762	2.23	417.0	10
			1.08	not found	-	-	7223.7	4
			1.06	feas.	767	2.87	1233.8	15
			1.04	not found	-	-	7316.4	6
			1.02	not found	-	-	7506.6	8
			1.01	not found	-	-	7255.9	5
$I_{50/m/50/3(9)}$	1026	1.7	1.1	opt.	1028	0.19	21.7	6
			1.08	opt.	1037	1.06	28.5	8
			1.06	opt.	1037	1.06	26.2	6
			1.04	not found	-	-	90.8	15
			1.02	not found	-	-	109.7	15
			1.01	not found	-	-	105.9	15
$I_{50/m/50/4(7)}$	962	6.1	1.1	opt.	966	0.41	121.4	6
			1.08	opt.	966	0.41	95.0	5
			1.06	opt.	966	0.41	137.7	6
			1.04	feas.	991	2.93	409.5	15
			1.02	feas.	991	2.93	404.8	15
			1.01	not found	-	-	485.8	15
$I_{50/m/50/5(7)}$	809	1.6	1.1	opt.	809	0.00	44.1	6
			1.08	opt.	809	0.00	43.8	6
			1.06	feas.	819	1.22	143.4	15
			1.04	opt.	820	1.34	70.0	8
			1.02	opt.	821	1.46	96.6	10
			1.01	not found	-	-	160.7	15

heuristic based on a dichotomic search where at each step, a simplified version of the problem is solved and ii) a branch-and-price algorithm where variables (columns) are indexed by the route and the driver.

Solution methods are evaluated on instances of 10 and 25 customers, plus instances with 50 customers for the heuristic. Results show that the heuristic is able to solve optimally in most instances. Its main weakness is that when an instance is unfeasible, the heuristic cannot prove it and consequently, the feasibility of many instances is not proved (or disproved). The branch-and-price algorithm permits to prove unfeasibility and is efficient to solve small instances but it shows its limits on mid-sized instances. Results show that the cost of equity on small instances (10 customers) is high but small on larger instances (25 and 50 customers).

A first interesting perspective to this work would be to improve the exact method in order to evaluate larger instances especially to evaluate feasibility when the parameter α is small. To limit the number of variables and pricing problems to solve, we could propose a model where equity is managed with constraints on customers which would add n new constraints into the master problem. It would considerably reduce the number of variables and a single pricing problem would be sufficient. Another perspective is to study the emergency version of the problem, where requests can be revealed during the execution of the routes. This problem is dynamic and new solution methods need to be proposed. Also, the stochastic version of the problem would require new solution methods. A last perspective for which some research has already been investigated is to consider several transportation companies and consider equity in terms of benefits between them. Collaborations exchanging requests between companies can improve the equity and benefits of companies.

Conclusion and perspectives

5.1 Conclusion

In this thesis, we have introduced novel models and solution methods to incorporate equity into routing problems. As a benchmark problem, we have utilized non-emergency transportation of patients and defined a fundamental problem called Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD) in Chapter 2. To address this problem, we have developed a branch-and-price algorithm based on a multi-trip model. Experiments show that this model outperforms a standard pickup-and-delivery model. Chapters 3 and 4 define extensions of the MTMPD-RD integrating workload equity, and the different solution methods proposed are all based on or adapted from the branch-and-price algorithm proposed to solve the MTMPD-RD.

In Chapter 3, we investigate a multi-period variant of the MTMPD-RD with workload equity, where requests are dynamically revealed on a daily basis making the problem dynamic. This unique aspect distinguishes it from the majority of existing literature, which usually focuses on extensions of the Periodic VRP (PVRP). In this chapter, equity is measured on painfulness of requests, using a constant-sum metric. To address this novel problem, we propose five distinct solution frameworks aimed at effectively managing equity. These frameworks are rigorously evaluated through extensive computational experiments, involving a wide range of Key Performance Indicators for comprehensive comparison and analysis. Results highlight the strengths and weaknesses of each proposed framework. Specifically, the findings indicate that an extended time horizon leads to a reduction in both workload imbalances and cost increases. Overall, the introduced frameworks demonstrate their effectiveness in achieving high-quality solutions with regards to both cost optimization and equity considerations.

Chapter 4 extends the MTMPD-RD by introducing workload equity considerations aimed at achieving balanced route costs. It presents a distinctive approach that sets it apart from the prevailing literature. Instead of defining two explicit objectives, namely cost minimization and equity, the focus here is on a single objective: cost minimization. Equity is ensured through the implementation of constraints that restrict route costs. Two solution methods, a heuristic and an exact approach, are proposed to address the problem. Computational experiments highlight the limitations of these algorithms in solving large instances, particularly the exact method. However, the experiments also demonstrate that a high level of equity can be achieved with only a minor impact on the overall solution cost.

In conclusion, this thesis has made contributions to the field of workload equity in routing problems. It has introduced innovative algorithms based on branch-and-price for handling equity considerations and

analyzed the impact of equity on routing solutions. These contributions have enhanced our understanding of the complex relationship between equity and routing optimization, providing valuable insights for future research in this area.

5.2 Perspectives

Workload equity is an emerging topic in the literature of Vehicle Routing Problem, and there are various perspectives to explore. Some of these perspectives are direct extensions of the work presented in this thesis. Specifically, a direct extension of Chapter 3 is to consider a varying perception of painfulness depending on individual drivers. This can provide further insights into incorporating individual factors in workload equity analysis. Another direct perspective for Chapter 4 is to propose a new exact solution method to solve larger instances. This would address the limitations observed in the computational experiments.

Another relevant perspective involves applying our solution methods to other problem domains to assess their effectiveness. This would allow for a broader understanding of the applicability and generalizability of the proposed approaches. Additionally, it is important to investigate the emergency version of the MTMPD-RD, wherein requests can emerge during the execution of routes. This dynamic problem necessitates the development of new solution methods capable to real time changes. Moreover, addressing the stochastic variant of the problem would require the introduction of novel solution approaches that can effectively handle uncertainty. Lastly, considering multiple transportation companies and examining equity in terms of benefits between them presents an additional avenue for exploration. Collaborative efforts, involving the exchange of requests between companies, have the potential to enhance equity and the overall benefits accrued by these companies. In the following sections, we detail these perspectives.

5.2.1 Different perception of painfulness or preferences

In Chapter 3, each request i is assigned an integer value π_i , representing its level of painfulness. Painfulness is determined based on various factors, such as the age of the patients, the number of patients, their mobility, and the type of residential building they reside in. The higher the value of π_i , the more painful the request is considered to be. A relevant extension is to consider different perception of painfulness for drivers. For instance, a driver m_1 may find a particular request i more painful than another driver m_2 does. In such cases, painfulness values must be indexed by drivers, denoted as $\pi_i^1 > \pi_i^2$. As a result, painfulness is no longer a constant-sum metric, necessitating the adaptation of solution methods. Another related aspect to explore is assigning preferences to drivers to achieve a balance. In this scenario, equity depends on individual preferences, and the assumption of anonymity, which is typically made in the literature, no longer holds.

5.2.2 New exact solution method for the MTMPD-RD-WE

In Chapter 4, the proposed branch-and-price algorithm for the MTMPD-RD-WE showed its limits to solve large instances. This is primarily due to the fact that routes are generated separately between drivers, resulting in a large number of variables and the need to solve distinct pricing problems for each driver. In addition, when α is small, the relaxation becomes weaker, further impacting the algorithm efficiency. To improve efficiency of the algorithm and solve larger instances, a potential approach could involve proposing a new model where equity constraints are based on nodes instead of drivers. This means that previous equity constraints:

$$\sum_{r \in \Omega(m_k)} c_r \theta_r^k \leq \frac{\alpha}{K} \times \sum_{m_l \in \mathcal{M}} \sum_{r \in \Omega(m_l)} c_r \theta_r^l \quad \forall m_k \in \mathcal{M} \quad (5.1)$$

can be expressed as follows:

$$\sum_{r \in \Omega} a_i^r \theta_r c_r \leq \frac{\alpha}{K} \times \sum_{r \in \Omega} \theta_r c_r \quad \forall i \in \mathcal{N} \quad (5.2)$$

These constraints ensure that each customer is visited by a route with a routing time respecting the imposed limit for equity. By using these equity constraints, routes can be generated for all drivers which reduces the number of variables and a single pricing problem is sufficient for all drivers. This consolidation of pricing problems may contribute to the overall efficiency of the algorithm. This will help allowing for improved computational performance to handle larger instances.

5.2.3 Application of the solution methods to other problems

To validate the proposed solution methods of Chapters 3 and 4 it would be relevant to apply them to other problems both theoretical and practical. From a theoretical point of view, it is valuable to test these methods on well-known generic problems and compare the obtained solutions with those already published in the literature. For example, instead of using the MTMPD-RD as the underlying routing problem, we can use the Capacitated Vehicle Routing Problem (CVRP) and compare our solutions with solutions of the literature for the VRPRB (bi-objective extension of the CVRP with the range as a second objective). From a practical perspective, it is beneficial to apply our algorithms to real-world applications such as last-mile delivery or home-to-work transportation scenarios. This would allow us to assess the effectiveness of the proposed algorithms in addressing the challenges and complexities of these specific problems. By conducting such validation on both theoretical and practical fronts, we can establish the robustness and applicability of our solution methods beyond their initial contexts.

5.2.4 Dynamic version/emergency

This thesis focuses on the MTMPD-RD as basic routing problem, specifically addressing the non-emergency transportation of patients problem. It would be interesting to consider the emergency version of this problem. In the emergency version, the problem becomes dynamic, where requests are revealed during the route execution, necessitating real time (online) updates to the routes. Consequently, novel solution approaches are required for the MTMPD-RD and its related problems, while incorporating equity constraints. Furthermore, for enhanced realism, it is pertinent to integrate stochastic demands. By considering uncertain and varying demands, the proposed models and solution approaches can be further adapted in order to handle the stochastic nature of real-world routing problems.

5.2.5 Equity in collaborative vehicle routing

Another promising avenue to explore is the integration of equity considerations in collaborative vehicle routing. In the real world, multiple transportation companies compete to offer their services. The primary goal is to maximize their profits while simultaneously ensuring a fair distribution of requests among them. Collaboration among transportation companies can be achieved through the exchange of

requests. By sharing requests, companies can collectively optimize their operations while maintaining equity. Although there is existing research in the literature on this subject, there are still open questions and opportunities for further exploration. Dealing with the integration of equity in collaborative vehicle routing can contribute to a deeper understanding of how equity and efficiency can be balanced in a competitive environment. This research direction could lead to the development of innovative models, solution approaches and strategies.

Appendix

This appendix presents the detailed results of the MT-BP algorithm on instances of sets E_1 , E_2 and E_3 studied in Section 2.6.6 of Chapter 2. Similarly to Tables 2.7 and 2.8 which present the results on instances of set E_0 with 50 and 100 customers, Tables A.1 and A.2 present the results on instances of set E_1 with 50 and 100 customers and the same for Tables A.3 and A.4 on instances of set E_2 and for Tables A.5 and A.6 on instances of set E_3 . These tables show for each instance, the value of the lower bound (“RLB”) at the root node, the computing time (“Time^{RLB}”) in seconds to reach “RLB”, the objective function value of the best integer solution found (“UB”) and the total computing time (“Time”) in seconds of MT-BP. The symbol “-” for “Time^{RLB}” and “Time” means that the process did not finish within the time limit of two hours. For the other parameters, the symbol “-” means that no value was found within the time limit of two hours. Value “UB” is highlighted in bold if optimality is proven. We also provide the gap at the root node (“root gap”) between RLB and the optimal solution and, the final gap (“final gap”) when MT-BP terminates with a non-proven optimum. The value “root gap” is only provided if optimality is proven. Columns “#trips”, “min”, and “max” respectively provide the total number of trips, the minimum number of trips per vehicle, and the maximum number of trips per vehicle in the obtained solution.

Table A.1: Results of the MT-BP algorithm on the 50-customers instances of set E_1

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{50/s/25/1(7)}$	1003.60	6.0	1008	22.2	0.44	0.00	20	2	4
$I_{50/s/25/2(6)}$	898.36	14.8	899	20.3	0.07	0.00	21	2	6
$I_{50/s/25/3(6)}$	903.20	22.0	916	463.8	1.40	0.00	20	2	5
$I_{50/s/25/4(6)}$	839.30	16.1	847	2256.9	0.91	0.00	19	2	4
$I_{50/s/25/5(6)}$	775.59	15.2	784	326.8	1.07	0.00	20	2	5
$I_{50/s/50/1(6)}$	758.50	7.1	775	1133.4	2.13	0.00	17	2	4
$I_{50/s/50/2(6)}$	682.13	13.5	689	423.6	1.00	0.00	16	2	3
$I_{50/s/50/3(7)}$	898.46	6.9	903	22.6	0.50	0.00	19	2	4
$I_{50/s/50/4(6)}$	834.89	10.3	848	257.3	1.55	0.00	17	2	3
$I_{50/s/50/5(6)}$	743.50	10.4	746	18.1	0.34	0.00	19	2	4
$I_{50/s/75/1(6)}$	861.78	7.9	873	202.7	1.28	0.00	21	3	4
$I_{50/s/75/2(6)}$	800.67	7.5	809	35.9	1.03	0.00	19	2	4
$I_{50/s/75/3(6)}$	809.00	12.5	815	54.5	0.74	0.00	20	2	5
$I_{50/s/75/4(6)}$	847.21	4.9	862	-	-	0.23	19	2	4
$I_{50/s/75/5(6)}$	948.31	6.5	993	-	-	2.95	24	3	6
$I_{50/m/25/1(6)}$	799.13	7.9	801	11.7	0.23	0.00	12	1	3
$I_{50/m/25/2(5)}$	705.84	19.3	719	154.3	1.83	0.00	13	2	3
$I_{50/m/25/3(6)}$	750.35	14.9	751	23.7	0.09	0.00	14	1	3
$I_{50/m/25/4(5)}$	692.00	25.2	692	25.3	0.00	0.00	14	2	3
$I_{50/m/25/5(5)}$	708.40	18.5	724	881.9	2.15	0.00	14	2	4
$I_{50/m/50/1(5)}$	680.00	12.3	681	13.5	0.15	0.00	15	2	4
$I_{50/m/50/2(5)}$	619.00	21.1	623	45.5	0.64	0.00	12	2	3
$I_{50/m/50/3(6)}$	806.65	7.5	833	597.7	3.16	0.00	15	2	4
$I_{50/m/50/4(5)}$	749.20	12.6	781	3601.7	4.07	0.00	14	2	4
$I_{50/m/50/5(5)}$	675.70	12.0	678	44.6	0.34	0.00	13	2	3
$I_{50/m/75/1(5)}$	650.26	9.9	662	54.3	1.77	0.00	15	2	4
$I_{50/m/75/2(5)}$	693.09	9.9	708	135.8	2.11	0.00	14	2	4
$I_{50/m/75/3(5)}$	723.95	14.1	734	-	-	0.56	13	2	3
$I_{50/m/75/4(5)}$	703.80	7.0	716	970.6	1.70	0.00	13	2	3
$I_{50/m/75/5(5)}$	759.49	8.8	769	64.7	1.24	0.00	15	2	4

Table A.2: Results of the MT-BP algorithm on the 100-customers instances of set E_1

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{100/s/25/1(13)}$	1751.20	530.2	-	-	-	-	-	-	-
$I_{100/s/25/2(12)}$	1633.59	983.5	-	-	-	-	-	-	-
$I_{100/s/25/3(12)}$	1555.06	961.5	-	-	-	-	-	-	-
$I_{100/s/25/4(12)}$	1413.25	922.5	-	-	-	-	-	-	-
$I_{100/s/25/5(13)}$	1587.33	536.4	-	-	-	-	-	-	-
$I_{100/s/50/1(12)}$	1268.13	392.5	-	-	-	-	-	-	-
$I_{100/s/50/2(13)}$	1366.79	436.1	-	-	-	-	-	-	-
$I_{100/s/50/3(13)}$	1517.63	342.0	-	-	-	-	-	-	-
$I_{100/s/50/4(12)}$	1386.84	454.8	-	-	-	-	-	-	-
$I_{100/s/50/5(12)}$	1323.56	350.8	-	-	-	-	-	-	-
$I_{100/s/75/1(12)}$	1533.37	315.9	-	-	-	-	-	-	-
$I_{100/s/75/2(12)}$	1451.83	423.7	-	-	-	-	-	-	-
$I_{100/s/75/3(12)}$	1479.75	293.7	-	-	-	-	-	-	-
$I_{100/s/75/4(12)}$	1618.16	216.7	-	-	-	-	-	-	-
$I_{100/s/75/5(12)}$	1618.26	320.9	-	-	-	-	-	-	-
$I_{100/m/25/1(11)}$	1247.53	370.4	-	-	-	-	-	-	-
$I_{100/m/25/2(11)}$	1191.23	417.5	-	-	-	-	-	-	-
$I_{100/m/25/3(11)}$	1177.67	525.4	1192	-	-	0.34	23	1	3
$I_{100/m/25/4(10)}$	1127.86	655.3	1140	-	-	0.49	25	2	3
$I_{100/m/25/5(11)}$	1204.74	356.8	-	-	-	-	-	-	-
$I_{100/m/50/1(10)}$	1047.94	474.9	-	-	-	-	-	-	-
$I_{100/m/50/2(11)}$	1137.18	353.5	-	-	-	-	-	-	-
$I_{100/m/50/3(11)}$	1220.10	332.7	1227	4031.1	0.56	0.00	21	1	2
$I_{100/m/50/4(10)}$	1133.05	370.2	-	-	-	-	-	-	-
$I_{100/m/50/5(10)}$	1131.16	309.7	1153	-	-	0.93	20	1	3
$I_{100/m/75/1(10)}$	1118.44	288.3	1120	569.5	0.14	0.00	22	2	3
$I_{100/m/75/2(10)}$	1157.95	344.9	-	-	-	-	-	-	-
$I_{100/m/75/3(10)}$	1148.98	216.1	-	-	-	-	-	-	-
$I_{100/m/75/4(10)}$	1163.09	223.4	-	-	-	-	-	-	-
$I_{100/m/75/5(10)}$	1139.19	228.0	-	-	-	-	-	-	-

Table A.3: Results of the MT-BP algorithm on the 50-customers instances of set E_2

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{50/s/25/1(4)}$	1283.50	14.0	1298	59.4	1.12	0.00	24	5	7
$I_{50/s/25/2(5)}$	1062.00	13.6	1068	37.5	0.56	0.00	26	4	6
$I_{50/s/25/3(5)}$	1111.50	11.4	1112	15.2	0.04	0.00	24	3	7
$I_{50/s/25/4(5)}$	1040.00	16.1	1040	34.9	0.00	0.00	27	4	7
$I_{50/s/25/5(4)}$	1009.50	13.5	1026	29.3	1.61	0.00	24	5	7
$I_{50/s/50/1(5)}$	1017.00	4.9	1017	7.2	0.00	0.00	22	3	8
$I_{50/s/50/2(4)}$	878.00	8.0	884	20.8	0.68	0.00	19	3	6
$I_{50/s/50/3(4)}$	1115.00	10.3	1118	15.1	0.27	0.00	22	5	7
$I_{50/s/50/4(4)}$	1100.00	16.9	1100	16.9	0.00	0.00	23	5	7
$I_{50/s/50/5(5)}$	1005.50	7.2	1006	9.0	0.05	0.00	22	4	5
$I_{50/s/75/1(5)}$	987.00	2.8	991	11.3	0.40	0.00	22	3	5
$I_{50/s/75/2(4)}$	951.50	9.1	959	13.2	0.78	0.00	21	5	6
$I_{50/s/75/3(4)}$	1028.50	4.2	1039	21.4	1.01	0.00	24	5	7
$I_{50/s/75/4(4)}$	1101.00	6.7	1101	6.7	0.00	0.00	24	5	7
$I_{50/s/75/5(4)}$	1074.00	4.6	1074	12.1	0.00	0.00	25	4	7
$I_{50/m/25/1(5)}$	1254.00	6.2	1269	149.2	1.18	0.00	24	3	6
$I_{50/m/25/2(4)}$	1103.00	21.7	1103	24.5	0.00	0.00	28	5	9
$I_{50/m/25/3(5)}$	1096.42	8.4	1101	16.7	0.42	0.00	22	4	5
$I_{50/m/25/4(5)}$	1006.00	9.1	1014	35.8	0.79	0.00	25	3	7
$I_{50/m/25/5(5)}$	980.00	6.9	980	6.9	0.00	0.00	21	3	6
$I_{50/m/50/1(4)}$	996.00	7.9	996	7.9	0.00	0.00	21	3	6
$I_{50/m/50/2(6)}$	884.00	1.4	884	1.4	0.00	0.00	20	2	5
$I_{50/m/50/3(4)}$	1072.00	13.4	1072	13.9	0.00	0.00	22	4	7
$I_{50/m/50/4(4)}$	1003.00	6.6	1003	6.6	0.00	0.00	19	3	6
$I_{50/m/50/5(5)}$	956.00	9.6	956	11.2	0.00	0.00	20	2	6
$I_{50/m/75/1(4)}$	968.33	8.4	979	33.4	1.09	0.00	24	4	7
$I_{50/m/75/2(5)}$	927.50	2.9	960	27.6	3.39	0.00	20	3	5
$I_{50/m/75/3(5)}$	932.00	3.6	933	7.7	0.11	0.00	23	3	6
$I_{50/m/75/4(5)}$	1024.50	1.7	1025	3.5	0.05	0.00	21	4	5
$I_{50/m/75/5(5)}$	1052.00	1.9	1052	3.2	0.00	0.00	23	3	6

Table A.4: Results of the MT-BP algorithm on the 100-customers instances of set E_2

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{100/s/25/1(9)}$	2076.00	1350.8	-	-	-	-	-	-	-
$I_{100/s/25/2(10)}$	1863.50	815.6	1871	-	-	0.27	43	3	6
$I_{100/s/25/3(10)}$	1868.88	1243.4	1883	-	-	0.48	44	2	8
$I_{100/s/25/4(9)}$	1739.50	955.3	-	-	-	-	-	-	-
$I_{100/s/25/5(8)}$	1899.25	802.0	1929	-	-	0.23	39	3	7
$I_{100/s/50/1(9)}$	1621.00	144.9	1629	-	-	0.06	36	3	6
$I_{100/s/50/2(8)}$	1716.31	243.3	1731	-	-	0.29	34	3	5
$I_{100/s/50/3(8)}$	1845.50	770.6	-	-	-	-	-	-	-
$I_{100/s/50/4(9)}$	1698.00	1105.0	1717	3426.3	1.11	0.00	38	2	6
$I_{100/s/50/5(10)}$	1635.25	228.9	1641	496.6	0.35	0.00	35	2	5
$I_{100/s/75/1(9)}$	1710.75	289.8	-	-	-	-	-	-	-
$I_{100/s/75/2(8)}$	1688.50	264.9	1695	5851.1	0.38	0.00	40	3	6
$I_{100/s/75/3(8)}$	1724.50	111.8	-	-	-	-	-	-	-
$I_{100/s/75/4(8)}$	2006.17	131.9	2011	1460.2	0.24	0.00	43	5	7
$I_{100/s/75/5(9)}$	1879.50	111.4	1899	2791.9	1.03	0.00	43	4	7
$I_{100/m/25/1(9)}$	1948.83	1186.5	-	-	-	-	-	-	-
$I_{100/m/25/2(9)}$	1773.50	1063.6	1776	2438.7	0.14	0.00	37	3	6
$I_{100/m/25/3(10)}$	1769.25	939.6	1785	2044.9	0.88	0.00	38	3	5
$I_{100/m/25/4(10)}$	1674.67	870.2	1686	3751.2	0.67	0.00	38	2	5
$I_{100/m/25/5(10)}$	1878.75	468.2	1891	-	-	0.32	39	3	5
$I_{100/m/50/1(10)}$	1516.54	101.0	-	-	-	-	-	-	-
$I_{100/m/50/2(10)}$	1647.50	227.2	1656	3896.5	0.51	0.00	33	2	4
$I_{100/m/50/3(8)}$	1710.00	314.1	1738	6386.8	1.61	0.00	33	2	6
$I_{100/m/50/4(9)}$	1704.83	546.8	1719	4247.2	0.82	0.00	32	3	5
$I_{100/m/50/5(9)}$	1538.50	245.3	1540	393.6	0.10	0.00	31	3	4
$I_{100/m/75/1(9)}$	1624.29	161.6	1651	-	-	1.09	37	2	6
$I_{100/m/75/2(10)}$	1603.00	109.6	1613	156.0	0.62	0.00	33	2	4
$I_{100/m/75/3(10)}$	1582.83	75.9	1600	4432.7	1.07	0.00	34	2	6
$I_{100/m/75/4(10)}$	1738.92	48.6	1754	-	-	0.26	38	2	6
$I_{100/m/75/5(9)}$	1777.00	188.6	1788	588.9	0.62	0.00	41	4	6

Table A.5: Results of the MT-BP algorithm on the 50-customers instances of set E_3

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{50/s/25/1(4)}$	1079.00	27.5	1086	2126.9	0.64	0.00	18	3	6
$I_{50/s/25/2(4)}$	940.88	48.6	943	215.1	0.23	0.00	22	5	6
$I_{50/s/25/3(4)}$	951.14	33.9	954	135.1	0.30	0.00	21	5	6
$I_{50/s/25/4(4)}$	910.75	45.9	913	143.7	0.25	0.00	22	5	6
$I_{50/s/25/5(4)}$	883.00	28.5	885	60.1	0.23	0.00	19	3	8
$I_{50/s/50/1(4)}$	893.69	22.8	894	39.2	0.03	0.00	18	2	6
$I_{50/s/50/2(3)}$	789.61	36.2	792	448.1	0.30	0.00	18	5	7
$I_{50/s/50/3(3)}$	934.42	48.7	942	181.6	0.81	0.00	18	5	7
$I_{50/s/50/4(4)}$	913.00	26.7	923	771.7	1.08	0.00	19	3	7
$I_{50/s/50/5(3)}$	954.00	63.0	954	63.3	0.00	0.00	21	6	8
$I_{50/s/75/1(4)}$	907.50	12.8	911	332.5	0.38	0.00	22	4	8
$I_{50/s/75/2(3)}$	882.00	37.6	908	-	-	0.88	19	4	8
$I_{50/s/75/3(3)}$	874.29	31.6	884	542.1	1.10	0.00	20	6	7
$I_{50/s/75/4(4)}$	928.50	18.7	935	1334.3	0.70	0.00	21	5	6
$I_{50/s/75/5(4)}$	974.50	11.5	980	38.0	0.56	0.00	22	4	8
$I_{50/m/25/1(4)}$	934.33	26.9	948	99.7	1.44	0.00	16	2	5
$I_{50/m/25/2(3)}$	903.29	69.1	905	110.6	0.19	0.00	19	6	7
$I_{50/m/25/3(4)}$	841.25	41.7	845	88.3	0.44	0.00	15	3	5
$I_{50/m/25/4(4)}$	800.00	41.3	826	-	-	2.06	16	3	5
$I_{50/m/25/5(4)}$	803.60	25.6	810	45.7	0.79	0.00	14	2	5
$I_{50/m/50/1(4)}$	788.00	14.0	788	14.0	0.00	0.00	14	3	4
$I_{50/m/50/2(4)}$	759.50	11.0	763	39.8	0.46	0.00	15	3	5
$I_{50/m/50/3(3)}$	881.50	38.2	888	113.9	0.73	0.00	17	5	7
$I_{50/m/50/4(3)}$	851.79	34.9	871	1899.2	2.21	0.00	14	4	5
$I_{50/m/50/5(4)}$	808.00	39.8	808	39.9	0.00	0.00	15	2	5
$I_{50/m/75/1(4)}$	763.75	19.9	783	-	-	0.26	14	2	5
$I_{50/m/75/2(4)}$	775.00	14.3	778	40.3	0.39	0.00	14	3	4
$I_{50/m/75/3(4)}$	748.00	17.8	756	1589.3	1.06	0.00	14	2	5
$I_{50/m/75/4(4)}$	795.50	6.5	797	10.0	0.19	0.00	15	2	6
$I_{50/m/75/5(4)}$	830.33	10.2	836	126.1	0.68	0.00	14	2	5

Table A.6: Results of the MT-BP algorithm on the 100-customers instances of set E_3

Instance	RLB		UB		Gaps (%)		Trips		
	RLB	Time ^{RLB}	UB	Time	root gap	final gap	nb trips	min	max
$I_{100/s/25/1(8)}$	1856.70	3897.5	-	-	-	-	-	-	-
$I_{100/s/25/2(8)}$	1713.50	6841.0	-	-	-	-	-	-	-
$I_{100/s/25/3(8)}$	-	-	-	-	-	-	-	-	-
$I_{100/s/25/4(8)}$	1564.83	6114.4	-	-	-	-	-	-	-
$I_{100/s/25/5(8)}$	1719.00	2861.6	-	-	-	-	-	-	-
$I_{100/s/50/1(7)}$	1401.31	1022.8	-	-	-	-	-	-	-
$I_{100/s/50/2(6)}$	1473.75	1794.3	1486	-	-	0.32	30	4	6
$I_{100/s/50/3(7)}$	1618.71	2463.9	-	-	-	-	-	-	-
$I_{100/s/50/4(7)}$	1492.00	3060.2	-	-	-	-	-	-	-
$I_{100/s/50/5(7)}$	1454.44	2681.5	1461	6365.7	0.45	0.00	32	4	6
$I_{100/s/75/1(7)}$	1606.92	3125.0	-	-	-	-	-	-	-
$I_{100/s/75/2(6)}$	1521.61	2131.3	-	-	-	-	-	-	-
$I_{100/s/75/3(7)}$	1583.21	994.0	-	-	-	-	-	-	-
$I_{100/s/75/4(8)}$	1783.20	1495.5	-	-	-	-	-	-	-
$I_{100/s/75/5(8)}$	1750.50	1794.6	-	-	-	-	-	-	-
$I_{100/m/25/1(7)}$	1483.58	2937.5	-	-	-	-	-	-	-
$I_{100/m/25/2(7)}$	1399.01	2273.5	-	-	-	-	-	-	-
$I_{100/m/25/3(8)}$	1393.37	1895.7	-	-	-	-	-	-	-
$I_{100/m/25/4(8)}$	1350.13	2644.9	1357	-	-	0.33	26	2	4
$I_{100/m/25/5(8)}$	1477.33	1517.9	-	-	-	-	-	-	-
$I_{100/m/50/1(8)}$	1238.17	557.9	-	-	-	-	-	-	-
$I_{100/m/50/2(7)}$	1360.22	1490.5	-	-	-	-	-	-	-
$I_{100/m/50/3(6)}$	1415.34	1397.8	-	-	-	-	-	-	-
$I_{100/m/50/4(7)}$	1385.97	1480.3	-	-	-	-	-	-	-
$I_{100/m/50/5(8)}$	1294.15	924.3	-	-	-	-	-	-	-
$I_{100/m/75/1(8)}$	1276.12	1034.2	-	-	-	-	-	-	-
$I_{100/m/75/2(8)}$	1280.50	460.8	-	-	-	-	-	-	-
$I_{100/m/75/3(8)}$	1263.92	269.7	-	-	-	-	-	-	-
$I_{100/m/75/4(8)}$	1340.50	248.9	1345	1886.5	0.33	0.00	25	2	4
$I_{100/m/75/5(8)}$	1386.32	460.0	-	-	-	-	-	-	-

Extended summary in French

Sommaire

Introduction générale

Chapitre 1 : Contexte de la thèse et état de l'art

- 1.1 Problème de tournées de véhicule et variantes pertinentes
- 1.2 Branch-and-price et génération de colonnes pour le VRP
- 1.3 Équité de la charge de travail dans le VRP

Chapitre 2 : Un algorithme de branch-and-price pour un problème de tournées avec des requêtes entrantes et sortantes

- 2.1 Introduction
- 2.2 Etat de l'art
- 2.3 Description du problème et formulation mathématique
- 2.4 Methode de résolution
- 2.5 Problème de plus court chemin avec contraintes de ressources et multi tours
- 2.6 Expériences numériques
- 2.7 Conclusion

Chapitre 3 : Équité sur la charge de travail dans les tournées de véhicules avec une perspective à moyen terme

- 3.1 Introduction
- 3.2 Etat de l'art
- 3.3 Description du problème
- 3.4 Méthodes de résolution
- 3.5 Solution du problème de tournées journalier
- 3.6 Expériences numériques
- 3.7 Conclusion

Chapitre 4 : Équité sur la charge de travail pour un problème de tournées de véhicules avec une métrique d'équité à somme variable

- 4.1 Introduction
- 4.2 Etat de l'art
- 4.3 Description du problème
- 4.4 Méthodes de résolution

4.5 Expériences numériques

4.6 Conclusion

Chapitre 5 : Conclusion et perspectives

5.1 Conclusion

5.2 Perspectives

Annexe A

Résumé détaillé en Français

Liste des Figures

Liste des Tables

Liste des Algorithmes

Bibliographie

Introduction

Le problème de tournées de véhicules (VRP) est l'un des types de problèmes les plus étudiés en Recherche Opérationnelle, à la fois pour des raisons pratiques et théoriques. Parmi la grande variété d'applications du VRP, l'équité a gagné en intérêt dans la littérature récente. L'équité peut concerner soit le service et le client, soit le fournisseur. Dans le premier cas, l'équité est souvent la principale préoccupation, tandis que dans le second, l'équité et le coût sont optimisés simultanément. Dans les applications concernant le fournisseur, l'équité consiste à répartir équitablement la charge de travail entre les conducteurs. Le principal défi consiste à trouver une répartition équitable de la charge de travail en minimisant l'impact sur les coûts des tournées. Pour mesurer l'équité, il est nécessaire de définir la mesure de la charge de travail. Les mesures de la charge de travail peuvent être classées comme étant à somme constante, où la somme reste constante pour toutes les solutions (*ex* : le nombre de clients desservis sur la route), ou à somme variable, où la somme dépend de la solution spécifique (*ex* : distance parcourue).

Malgré l'intérêt croissant pour l'équité dans la littérature récente, les choix de modèles et d'algorithmes ne sont pas toujours bien justifiés. En outre, la littérature actuelle concerne essentiellement l'équité journalière alors que les conducteurs peuvent accepter des journées déséquilibrées si l'équité est assurée sur un horizon plus long (par exemple, une semaine ou un mois). En outre, une faiblesse récurrente de la littérature actuelle est l'absence d'évaluation de l'impact de l'équité sur les coûts des tournées. De nombreuses pistes de recherche restent donc ouvertes.

C'est là que se situent les contributions de la thèse : Proposer de nouveaux modèles et algorithmes pour intégrer l'équité dans les solutions des problèmes de tournées et évaluer l'impact de l'équité sur le coût. Nous proposons des solutions pour les métriques à somme constante et à somme variable et utilisons le transport sanitaire comme application réelle. Nous résolvons la

plupart des problèmes de tournées à l'optimalité en mettant en œuvre des algorithmes de type "branch-and-price" (et de génération de colonnes).

Les contributions de cette thèse sont (i) la définition d'un nouveau problème de tournées motivé par le transport sanitaire et un algorithme de branch-and-price dédié pour le résoudre, (ii) la prise en compte de l'équité pour la version multi-période du problème de tournées initial avec une métrique à somme constante, (iii) l'extension du problème de tournées journalier impliquant l'équité sur une métrique à somme variable. Dans ce manuscrit, nous proposons et évaluons de nouvelles contraintes d'équité et des techniques pour les intégrer dans un algorithme de branch-and-price. Les solutions sont évaluées à l'aide de plusieurs indicateurs de performance (KPI).

Projet FITS (ANR)

Cette thèse fait partie du projet FITS - "Systèmes de Transport Intelligents et Flexibles", soutenu par l'Agence Nationale de la Recherche (ANR) sous la subvention ANR-18-CE22-0014. Le projet FITS vise à proposer de nouveaux modèles mathématiques et des approches de solutions afin de développer des outils intelligents pour aider à optimiser les services de livraison.

Dans ce contexte, les plateformes de prestataires de services gèrent l'affectation des demandes des clients aux ressources humaines. En général, ces plateformes ont besoin d'outils intelligents pour garantir une affectation intelligente, en particulier lorsque la demande varie fortement ou pour les services de transport à grande échelle. Le projet FITS vise à relever les défis scientifiques posés par l'optimisation de ces plateformes de prestataires de services.

Une conséquence non désirée des plateformes actuelles est que l'affectation crée des inégalités entre les conducteurs en termes de charge de travail. L'objectif du projet FITS est de trouver des solutions efficaces tout en améliorant l'équilibre de la charge de travail. C'est dans ce contexte que s'inscrit cette thèse.

Structure de la thèse

Cette thèse est divisé en cinq chapitres. Le Chapitre 1 fait état de l'art de la littérature liée à cette thèse. Les Chapitres 2, 3 et 4 correspondent aux contributions de cette thèse. Enfin, le Chapitre 5 conclue cette thèse et présente les perspectives.

Le Chapitre 1 fournit des définitions, des problèmes et des algorithmes liés à cette thèse et passe en revue les principales contributions de la littérature liée à cette thèse. Il est divisé en trois parties. Tout d'abord, le problème de tournées de véhicules (VRP) et ses variantes liées à notre travail sont brièvement passés en revue. Ensuite, le principe des algorithmes de génération de colonnes et de branch-and-price est expliqué. Enfin, la littérature relative à l'équité de la charge de travail dans le VRP est passée en revue.

Le Chapitre 2 présente un nouveau problème de tournées motivé par le transport sanitaire. Les patients ont des rendez-vous à l'hôpital et demandent à être transportés soit de leur domicile à l'hôpital (demande entrante), soit de l'hôpital à leur domicile (demande sortante). Il existe donc deux types de demandes et le problème peut être modélisé soit comme un problème de pickup and delivery, soit comme un problème multi-tours. Le problème est appelé Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD). Un algorithme Branch-and-Price (BP) est développé pour résoudre ce problème. Nous montrons que la formulation multi-tours est plus performante que la formulation pickup and delivery.

Le problème MTMPD-RD et l'algorithme BP sont utilisés comme base pour la définition des problèmes et des algorithmes des Chapitres 3 et 4.

Le Chapitre 3 présente une version à moyen terme du problème MTMPD-RD en l'intégrant dans un horizon de plusieurs jours où l'équité entre les conducteurs est prise en compte. Le problème est dynamique dans la mesure où les demandes sont révélées quotidiennement. Chaque jour, un problème de tournées et un problème d'affectation route-conducteur sont résolus. L'objectif est d'atteindre l'équité sur l'ensemble de l'horizon plutôt que sur chaque jour. La charge de travail à équilibrer équitablement entre les conducteurs est la pénibilité des demandes, cette métrique est à somme constante (la pénibilité totale est la même pour toute solution). Cinq approches sont proposées pour garantir l'équité en s'appuyant sur une stratégie d'affectation (potentiellement) couplée à des contraintes d'équité quotidiennes. L'algorithme BP utilisé pour résoudre les problèmes de tournées quotidiens est adapté en fonction du type de contraintes d'équité. Les approches sont comparées de manière approfondie par le biais d'expériences numériques où plusieurs indicateurs clés de performance sont utilisés pour évaluer les solutions.

Le Chapitre 4 étend les travaux des Chapitres 2 et 3 en considérant une métrique à somme variable (le temps de conduite des conducteurs) pour le MTMPD-RD. Une contrainte d'équité est ajoutée au modèle pour limiter l'excès de temps de conduite de chaque conducteur par rapport au temps de conduite moyen. Une méthode exacte et une heuristique toutes deux basées sur l'algorithme BP du Chapitre 2 sont proposées et comparées à l'aide d'expériences numériques. Dans la méthode exacte, les colonnes du problème principal sont indexées par route et par conducteur, ce qui impose des sous problèmes différents pour les conducteurs. Dans la méthode heuristique, l'algorithme branch-and-price est résolu itérativement avec une contrainte d'équité fixe et la structure de la solution guide la recherche pour l'étape suivante. L'impact de l'équité sur le coût des tournées est mesuré par des expériences numériques.

Enfin, le Chapitre 5 résume les principales contributions de la thèse et offre des perspectives prometteuses pour de futures recherches.

Résumés détaillés des Chapitres 2, 3 et 4

Résumé du Chapitre 2

Dans ce chapitre, nous avons introduit un nouveau problème issu du contexte du transport sanitaire. La principale originalité du problème étudié est qu'il existe deux types de demandes : un patient est soit pris en charge à son domicile et transporté à l'hôpital, soit l'inverse. Les demandes sont donc toujours reliées à l'hôpital, et les routes combinent les deux types de demandes. En outre, la qualité du service est préservée grâce à des contraintes de temps de parcours qui obligent les véhicules à revenir régulièrement à l'hôpital. Par conséquent, le problème peut être traité soit comme un problème spécial de pickup and delivery, soit comme un VRP spécial à tours multiples. Nous l'appelons le Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD).

L'objectif principal de ce chapitre était alors d'évaluer s'il était préférable d'aborder le problème sous la forme d'un VRP à tours multiples. Nous avons proposé un algorithme de type "branch-and-price" (MT-BP). Les contraintes complexes de temps et de capacité du problème rendent la définition et la gestion des ressources complexes dans le sous problème. Elles impliquent

également certaines difficultés dans les règles de branchement.

Le problème MTMPD-RD étant nouveau, nous avons généré un nouvel ensemble d'instances réalistes. Nous avons également mis en œuvre un algorithme de branch-and-price basé sur la formulation en pickup and delivery (PD-BP). Les expériences ont montré que le MT-BP est plus performant que le PD-BP et prouvent l'efficacité d'un modèle multi-tours pour ce problème. Le MT-BP a prouvé son efficacité en résolvant de manière optimale des instances impliquant jusqu'à 100 demandes.

Pour valider davantage le MT-BP, nous l'avons adapté à la résolution d'un problème similaire de la littérature, le VRPSPDTW. Nos expériences sur les instances de référence ont permis de clôturer le benchmark pour les petites instances et de clôturer quatre instances à 100 clients. Nous avons également détecté qu'une solution irréalisable avait été publiée dans la littérature. En outre, nous avons évalué la possibilité d'aborder le problème comme un problème d'acheminement de véhicules mono-tour suivi d'une affectation tour-véhicule. Les expériences montrent l'intérêt numérique mais aussi les limites de ce point de vue. Enfin, nous avons effectué d'autres analyses de sensibilité en générant de nouvelles instances difficiles avec un horizon temporel plus large et/ou un temps de trajet acceptable plus important. Les résultats montrent que le temps d'exécution de l'algorithme MT-BP est fortement impacté lorsque le détour maximal est augmenté, mais reste relativement stable pour des horizons temporels plus larges.

Résumé du Chapitre 3

L'équité dans les tournées de véhicules a fait l'objet d'un intérêt croissant au cours des deux dernières décennies. Alors que la majorité des études existantes considèrent l'équité sur une base journalière, nous traitons dans ce chapitre de l'équité de la charge de travail dans les tournées de véhicules sur un horizon de planification de plusieurs périodes. Le problème étudié est dynamique, dans le sens où les demandes sont révélées période par période. La métrique d'équité utilisée dans ce chapitre est dite à somme constante, ce qui signifie que pour toute solution réalisable, la somme totale de la charge de travail affectée aux conducteurs reste constante. Le problème de tournées étudié est motivé par la logistique des soins de santé.

Afin d'équilibrer l'équité entre les conducteurs sur l'ensemble de l'horizon de planification tout en minimisant le coût total des déplacements, nous proposons cinq approches. Il s'agit des approches suivantes : Routage d'abord, Équité ensuite (RFES), Équité constante à une période (SPCE), Équité constante à plusieurs périodes (MPCE), Équité adaptative à plusieurs périodes (MPAE) et Équité adaptative à la dernière période (LPAE). Ces approches sont comparées à une approche de référence, appelé Routage d'abord, Équité ensuite avec Information Parfaite (RFE-SPI) qui tente de garantir une équité parfaite sur l'ensemble de l'horizon temporel où toutes les demandes de tous les jours sont censées être connues à l'avance. L'algorithme de type branch-and-price du Chapitre 2 est utilisé pour résoudre le problème de tournées quotidien et est adapté aux différentes approches.

Nous pouvons tirer plusieurs conclusions de nos expériences. Nous montrons que l'approche RFES ne permet pas nécessairement d'équilibrer la charge de travail sur l'horizon temporel. Le déséquilibre diminue lorsque l'horizon temporel est plus long. Nous montrons que les approches SPCE, MPAE et LPAE ont des forces et des faiblesses différentes. L'approche MPCE ne présente pas d'avantage clair par rapport aux autres approches. Le tableau 3.8 résume les avantages et

les inconvénients des différentes approches. Ce chapitre est l'un des premiers à étudier l'équité dans les tournées de véhicules sur un horizon de plusieurs périodes. Nous pensons qu'il ouvre différentes perspectives de recherche.

Résumé du Chapitre 4

L'intérêt pour l'équité dans la littérature sur les tournées de véhicules s'est récemment considérablement accru. La plupart des études considèrent des modèles bi-objectifs avec deux objectifs explicites, l'un pour la minimisation des coûts et l'autre pour l'équité. Dans ce chapitre, nous préférons garder la minimisation du coût comme objectif unique et gérer l'équité avec des contraintes. Ces contraintes visent à limiter l'écart entre les coûts des routes et le coût moyen d'une route. La métrique d'équité est le coût de la route, une métrique à somme variable, ce qui signifie que la somme total de la charge de travail affectée aux conducteurs varie d'une solution à une autre. Nous utilisons le MTMPD-RD (le problème étudié au Chapitre 2) comme cas d'étude. Nous appelons cette extension MTMPD-RD avec équité de la charge de travail (MTMPD-RD-WE).

Nous montrons que le problème ne peut pas être résolu efficacement par un algorithme standard de branch-and-price avec la formulation simple de partitionnement de l'ensemble et proposons deux méthodes de résolution originales : i) une heuristique basée sur une recherche dichotomique où à chaque étape, une version simplifiée du problème est résolue; bien qu'heuristique, elle prouve l'optimalité dans certains cas et ii) un algorithme de branch-and-price où les variables (colonnes) sont indexées par la route et le conducteur.

Les méthodes sont évaluées sur des instances de 10 et 25 clients, plus des instances de 50 clients pour l'heuristique. Les résultats montrent que l'heuristique est capable de résoudre de manière optimale la plupart des instances. L'algorithme de branch-and-price permet de prouver l'infaisabilité et est efficace pour résoudre les petites instances, mais il montre ses limites pour les instances de taille moyenne. Les résultats montrent que le coût de l'équité est élevé pour les petites instances (10 clients) mais faible pour les instances plus grandes (25 et 50 clients).

Conclusion

Dans cette thèse, nous avons introduit de nouveaux modèles et méthodes de résolution pour incorporer l'équité dans les problèmes de tournées de véhicules. Nous avons utilisé le transport sanitaire comme application et avons défini dans le Chapitre 2 un problème de référence appelé le Multi-Trip Vehicle Routing Problem with Mixed Pickup and Delivery, and Release and Due dates (MTMPD-RD). Pour résoudre ce problème, nous avons développé un algorithme de branch-and-price basé sur un modèle multi-tours. Les expériences montrent que ce modèle est plus performant qu'un modèle standard de pickup and delivery. Les Chapitres 3 et 4 définissent des extensions du MTMPD-RD en intégrant l'équité de la charge de travail, et les différentes méthodes proposées sont toutes basées sur ou adaptées de l'algorithme de branch-and-price proposé pour résoudre le MTMPD-RD.

Cette thèse a apporté des contributions au domaine de l'équité de la charge de travail dans les problèmes de tournées de véhicules. Elle a introduit des algorithmes novateurs basés sur la méthode de branch-and-price pour traiter les considérations d'équité et a analysé l'impact de l'équité sur les coûts des solutions. Ces contributions ont amélioré notre compréhension de

la relation complexe entre l'équité et l'optimisation des tournées, fournissant des indications précieuses pour de futures recherches dans ce domaine.

Perspectives

L'équité de la charge de travail est un sujet émergent dans la littérature du problème de tournées de véhicules, et il y a plusieurs perspectives à explorer. Certaines de ces perspectives sont des extensions directes du travail présenté dans cette thèse. En particulier, une extension directe du Chapitre 3 consiste à prendre en compte une perception variable de la pénibilité en fonction des conducteurs. Cela peut permettre de mieux comprendre l'intégration des facteurs individuels dans l'analyse de l'équité de la charge de travail. Une perspective directe pour le Chapitre 4 est de proposer une nouvelle méthode exacte pour résoudre des instances plus importantes. Une méthode se basant sur un modèle où les contraintes d'équité seraient exprimées sur les nœuds au lieu des conducteurs est une perspective prometteuse. Cela permettrait de remédier aux limites observées dans les expériences numériques.

Une autre perspective pertinente consiste à appliquer nos méthodes à d'autres problématiques afin d'évaluer leur efficacité. Cela permettrait de mieux comprendre l'applicabilité et la généralisation des approches proposées. Il serait également pertinent d'étudier la version dynamique du MTMPD-RD, dans lequel des demandes (urgences) peuvent émerger pendant l'exécution des routes. Ce problème dynamique nécessite le développement de nouvelles méthodes capables de s'adapter aux changements en temps réel. De plus, la prise en compte de la variante stochastique du problème nécessiterait l'introduction de nouvelles approches capables de gérer efficacement l'incertitude. Enfin, la prise en compte de plusieurs entreprises de transport et l'analyse de l'équité en termes de bénéfices entre elles constituent une voie d'exploration supplémentaire. Les efforts de collaboration, impliquant l'échange de demandes entre les entreprises, ont le potentiel d'améliorer l'équité et les bénéfices globaux accumulés par ces entreprises.

List of Figures

1.1	Column generation principle	12
1.2	Example with two solutions: an efficient and an equitable	13
2.1	Route timing example	26
2.2	Customers distribution of instance $I_{100/s/25/5(15)}$	41
3.1	Data discovery and decision process	60
3.2	Trade-off between equity and cost on 5-days instances	73
3.3	Trade-off between equity and cost on 10-days instances	73
3.4	Trade-off between equity and cost on 20-days instances	74
3.5	Daily equity on 5-days instances	76
3.6	Long term equity evolution on 5-days instances	76
4.1	Example of curve $C(Z)$	94
4.2	$Q(Z)$: Decision tree	96

List of Tables

1.1	Satisfaction of the monotonicity axiom and the Pigou–Dalton transfer principle according to the type of equity metric	17
2.1	Notation	27
2.2	Decision variables	27
2.3	Comparison with the best-known solutions of the small-size instances for the VRPSPDTW	39
2.4	Comparison with the best-known solutions of the 100-customers instances for the VRPSPDTW	39
2.5	Comparison of the PD-BP and the MT-BP on 50-customers instances	43
2.6	Comparison of the PD-BP and the MT-BP on 100-customers instances	44
2.7	Results of the MT-BP algorithm on 50-customers instances	46
2.8	Results of the MT-BP algorithm on 100-customers instances	47
2.9	Summary of obtained solutions with MT-BP	47
2.10	Impact of fleet size for instance $I_{50/s/25/3}$	48
2.11	Impact of fleet size for instance $I_{50/m/25/4}$	48
2.12	Impact of fleet size for instance $I_{100/s/25/4}$	48
2.13	Impact of fleet size for instance $I_{100/m/25/5}$	49
2.14	Summary of obtained solutions with MT-BP for the E_1 instances	50
2.15	Summary of obtained solutions with MT-BP for the E_2 instances	50
2.16	Summary of obtained solutions with MT-BP for the E_3 instances	50
3.1	RFESPI assignment decision variables	66
3.2	Feasibility of RFESPI and RFES frameworks	71
3.3	Feasibility of SPCE and MPCE frameworks	71
3.4	Feasibility of MPAE and LPAE frameworks	71
3.5	Computing times for RFESPI and RFES frameworks	77
3.6	Computing times for SPCE and MPCE frameworks	77
3.7	Computing times for MPAE and LPAE frameworks	77
3.8	Pros and cons of the different frameworks	78
4.1	Decision variables	86
4.2	Instances of set S_{10}	106
4.3	Instances of set S_{25}	107
4.4	Instances of set $S_{50/s}$	109

4.5	Instances of set $S_{50/m}$	110
A.1	Results of the MT-BP algorithm on the 50-customers instances of set E_1	118
A.2	Results of the MT-BP algorithm on the 100-customers instances of set E_1	119
A.3	Results of the MT-BP algorithm on the 50-customers instances of set E_2	120
A.4	Results of the MT-BP algorithm on the 100-customers instances of set E_2	121
A.5	Results of the MT-BP algorithm on the 50-customers instances of set E_3	122
A.6	Results of the MT-BP algorithm on the 100-customers instances of set E_3	123

List of Algorithms

2.1	Column generation()	31
3.1	BEA strategy	63
4.1	Heuristic	98

Bibliography

- Agius, M., Absi, N., Feillet, D., and Garaix, T. (2022). A branch-and-price algorithm for a routing problem with inbound and outbound requests. *Computers & Operations Research*, page 105896.
- Angelelli, E. and Mansini, R. (2002). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In *Quantitative approaches to distribution logistics and supply chain management*, pages 249–267. Springer.
- Apte, U. M. and Mason, F. M. (2006). Analysis and improvement of delivery operations at the san francisco public library. *Journal of Operations Management*, 24(4):325–346.
- Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.
- Baños, R., Ortega, J., Gil, C., Márquez, A. L., and De Toro, F. (2013). A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & industrial engineering*, 65(2):286–296.
- Battarra, M., Cordeau, J.-F., and Iori, M. (2014). Chapter 6: pickup-and-delivery problems for goods transportation. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 161–191. SIAM.
- Bektaş, T., Repoussis, P. P., and Tarantilis, C. D. (2014). Chapter 11: Dynamic vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 299–347. SIAM.
- Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31.
- Bertazzi, L., Golden, B., and Wang, X. (2015). Min–max vs. min–sum vehicle routing: A worst-case analysis. *European Journal of Operational Research*, 240(2):372–381.
- Blakeley, F., Argüello, B., Cao, B., Hall, W., and Knolmayer, J. (2003). Optimizing periodic maintenance operations for schindler elevator corporation. *Interfaces*, 33(1):67–79.
- Borgulya, I. (2008). An algorithm for the capacitated vehicle routing problem with route balancing. *Central European Journal of Operations Research*, 16:331–343.
- Cappanera, P. and Scutellà, M. G. (2015). Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach. *Transportation Science*, 49(4):830–852.

- Cattaruzza, D., Absi, N., and Feillet, D. (2016). Vehicle routing problems with multiple trips. *4OR*, 14(3):223–259.
- Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3):573–586.
- Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M. M., and Soumis, F. (2000). *The VRP with time windows*. Citeseer.
- Cordeau, J.-F. and Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594.
- Cordeau, J.-F. and Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of operations research*, 153(1):29.
- Costa, L., Contardo, C., and Desaulniers, G. (2019). Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4):946–985.
- Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.
- Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. (2002). Vrp with pickup and delivery. *The vehicle routing problem*, 9:225–242.
- Desaulniers, G., Madsen, O. B., and Ropke, S. (2014). Chapter 5: The vehicle routing problem with time windows. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 119–159. SIAM.
- Desrochers, M., Desrosiers, J., and Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2):342–354.
- Desrosiers, J., Soumis, F., and Desrochers, M. (1984). Routing with time windows by column generation. *Networks*, 14(4):545–565.
- Doerner, K. F. and Salazar-González, J.-J. (2014). Chapter 7: Pickup-and-delivery problems for people transportation. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 193–212. SIAM.
- Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *4or*, 8(4):407–424.
- Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks: An International Journal*, 44(3):216–229.
- Fischetti, M., Toth, P., and Vigo, D. (1994). A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. *Operations Research*, 42(5):846–859.
- Fleischmann, B. (1990). The vehicle routing problem with multiple use of vehicles. *Forschungsbericht Fachbereich Wirtschaftswissenschaften, Universität Hamburg*.

- Fogue, M., Sanguesa, J. A., Naranjo, F., Gallardo, J., Garrido, P., and Martinez, F. J. (2016). Non-emergency patient transport services planning through genetic algorithms. *Expert Systems with Applications*, 61:262–271.
- Garaix, T., Artigues, C., Feillet, D., and Josselin, D. (2010). Vehicle routing problems with alternative paths: An application to on-demand transportation. *European Journal of Operational Research*, 204(1):62–75.
- Golden, B. L., Kovacs, A. A., and Wasil, E. A. (2014). Chapter 14: Vehicle routing applications in disaster relief. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 409–436. SIAM.
- Golden, B. L., Laporte, G., and Taillard, É. D. (1997). An adaptive memory heuristic for a class of vehicle routing problems with minmax objective. *Computers & Operations Research*, 24(5):445–452.
- Goodson, J. C. (2014). Election day routing of rapid response attorneys. *INFOR: Information Systems and Operational Research*, 52(1):1–9.
- Gulczynski, D., Golden, B., and Wasil, E. (2011). The period vehicle routing problem: New heuristics and real-world variants. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):648–668.
- Halvorsen-Weare, E. E. and Savelsbergh, M. W. (2016). The bi-objective mixed capacitated general routing problem with different route balance criteria. *European Journal of Operational Research*, 251(2):451–465.
- Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2014). A new exact algorithm to solve the multi-trip vehicle routing problem with time windows and limited duration. *4or*, 12(3):235–259.
- Hernandez, F., Feillet, D., Giroudeau, R., and Naud, O. (2016). Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows. *European Journal of Operational Research*, 249(2):551–559.
- Hof, J. and Schneider, M. (2019). An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery. *Networks*, 74(3):207–250.
- Irnich, S., Toth, P., and Vigo, D. (2014). The family of vehicle routing problems. *Vehicle Routing: Problems, Methods and Applications. Philadelphia: Society for Industrial and Applied Mathematics*, 119.
- Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511.
- Jozefowiez, N., Semet, F., and El-Ghazali, T. (2007). Target aiming pareto search and its application to the vehicle routing problem with route balancing. *Journal of Heuristics*, 13(5):455.

- Jozefowicz, N., Semet, F., and Talbi, E.-G. (2002). Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. In *Parallel Problem Solving from Nature—PPSN VII: 7th International Conference Granada, Spain, September 7–11, 2002 Proceedings*, pages 271–280. Springer.
- Jozefowicz, N., Semet, F., and Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309.
- Karsu, Ö. and Morton, A. (2015). Inequity averse optimization in operational research. *European Journal of Operational Research*, 245(2):343–359.
- Khorsi, M., Chaharsooghi, S. K., Bozorgi-Amiri, A., and Kashan, A. H. (2020). A multi-objective multi-period model for humanitarian relief logistics with split delivery and multiple uses of vehicles. *Journal of Systems Science and Systems Engineering*, 29(3):360–378.
- Kim, B.-I., Kim, S., and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642.
- Koc, C. and Karaoglan, I. (2011). A branch and cut algorithm for the vehicle routing problem with multiple use of vehicles. In *41st International Conference on Computers & Industrial Engineering*, pages 554–559.
- Koç, Ç., Laporte, G., and Tükenmez, İ. (2020). A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, 122:104987.
- Kohl, N., Desrosiers, J., Madsen, O. B., Solomon, M. M., and Soumis, F. (1999). 2-path cuts for the vehicle routing problem with time windows. *Transportation science*, 33(1):101–116.
- Lacomme, P., Prins, C., Prodhon, C., and Ren, L. (2015). A multi-start split based path relinking (msspr) approach for the vehicle routing problem with route balancing. *Engineering Applications of Artificial Intelligence*, 38:237–251.
- Laporte, G. (2009). Fifty years of vehicle routing. *Transportation science*, 43(4):408–416.
- Laporte, G. and Nobert, Y. (1983). A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5:77–85.
- Lehuédé, F., Péton, O., and Tricoire, F. (2020). A lexicographic minimax approach to the vehicle routing problem with route balancing. *European Journal of Operational Research*, 282(1):129–147.
- Lim, A., Zhang, Z., and Qin, H. (2017). Pickup and delivery service with manpower planning in hong kong public hospitals. *Transportation Science*, 51(2):688–705.
- Liu, M., Luo, Z., and Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81:267–288.
- Liu, R., Xie, X., and Garaix, T. (2013). Weekly home health care logistics. In *2013 10th IEEE international conference on networking, sensing and control (ICNSC)*, pages 282–287. IEEE.

- Liu, W., Dridi, M., Fei, H., and El Hassani, A. H. (2021). Solving a multi-period home health care routing and scheduling problem using an efficient matheuristic. *Computers & Industrial Engineering*, 162:107721.
- Luo, Z., Liu, M., and Lim, A. (2019). A two-phase branch-and-price-and-cut for a dial-a-ride problem in patient transportation. *Transportation Science*, 53(1):113–130.
- Marinakis, Y., Marinaki, M., and Migdalas, A. (2018). Variants and formulations of the vehicle routing problem. *Open Problems in Optimization and Data Analysis*, pages 91–127.
- Matl, P., Hartl, R. F., and Vidal, T. (2018). Workload equity in vehicle routing problems: A survey and analysis. *Transportation Science*, 52(2):239–260.
- Matl, P., Hartl, R. F., and Vidal, T. (2019). Workload equity in vehicle routing: The impact of alternative workload resources. *Computers & Operations Research*, 110:116–129.
- Mingozi, A., Roberti, R., and Toth, P. (2013). An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, 25(2):193–207.
- Molenbruch, Y., Braekers, K., and Caris, A. (2017). Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, 259(1):295–325.
- Mosheiov, G. (1998). Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering*, 34(3):669–684.
- Nekooghadirli, N., Gendreau, M., Potvin, J.-Y., and Vidal, T. (2022). Workload equity in multi-period vehicle routing problems. *arXiv preprint arXiv:2206.14596*.
- Oyola, J. and Løkketangen, A. (2014). Grasp-asp: An algorithm for the cvrp with route balancing. *Journal of Heuristics*, 20:361–382.
- Pandi, R. and Muralidharan, B. (1995). A capacitated general routing problem on mixed networks. *Computers & operations research*, 22(5):465–478.
- Paradiso, R., Roberti, R., Laganá, D., and Dullaert, W. (2020). An exact solution framework for multitrip vehicle-routing problems with time windows. *Operations Research*, 68(1):180–198.
- Parragh, S. N., Cordeau, J.-F., Doerner, K. F., and Hartl, R. F. (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR spectrum*, 34(3):593–633.
- Pessoa, A., Sadykov, R., Uchoa, E., and Vanderbeck, F. (2020). A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 183(1):483–523.
- Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.
- Privé, J., Renaud, J., Boctor, F., and Laporte, G. (2006). Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, 57(9):1045–1052.

- Qu, Y. and Bard, J. F. (2013). The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, 32:1–20.
- Qu, Y. and Bard, J. F. (2015). A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science*, 49(2):254–270.
- Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter, L. E. (2003). On the capacitated vehicle routing problem. *Mathematical programming*, 94:343–359.
- Ribeiro, R. and Ramalhinho Dias Lourenço, H. (2001). A multi-objective model for a multi-period distribution management problem. In *Proceedings of MIC'2001*.
- Shi, Y., Boudouh, T., and Grunder, O. (2018). An efficient tabu search based procedure for simultaneous delivery and pick-up problem with time window. *IFAC-PapersOnLine*, 51(11):241–246.
- Shi, Y., Zhou, Y., Boudouh, T., and Grunder, O. (2020). A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup–delivery and time window. *Engineering Applications of Artificial Intelligence*, 95:103901.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265.
- Subramanian, A., Uchoa, E., Pessoa, A. A., and Ochi, L. S. (2013). Branch-cut-and-price for the vehicle routing problem with simultaneous pickup and delivery. *Optimization Letters*, 7(7):1569–1581.
- Toth, P. and Vigo, D. (2002). Branch-and-bound algorithms for the capacitated vrp. In *The vehicle routing problem*, pages 29–51. SIAM.
- Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.
- Triki, N., Garaix, T., and Xie, X. (2014). A two-phase approach for periodic home health care planning. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 518–523. IEEE.
- Ulmer, M. W., Goodson, J. C., Mattfeld, D. C., and Thomas, B. W. (2020). On modeling stochastic dynamic vehicle routing problems. *EURO Journal on Transportation and Logistics*, 9(2):100008.
- Valle, C. A., Martinez, L. C., Da Cunha, A. S., and Mateus, G. R. (2011). Heuristic and exact algorithms for a min–max selective vehicle routing problem. *Computers & Operations Research*, 38(7):1054–1065.
- Vidal, T., Laporte, G., and Matl, P. (2020). A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, 286(2):401–416.
- Wang, C., Mu, D., Zhao, F., and Sutherland, J. W. (2015). A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows. *Computers & Industrial Engineering*, 83:111–122.
- Wang, H.-F. and Chen, Y.-Y. (2012). A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & industrial engineering*, 62(1):84–95.

- Wang, Y., Zhao, L., Savelsbergh, M., and Wu, S. (2022). Multi-period workload balancing in last-mile urban delivery. *Transportation Science*, 56(5):1348–1368.
- Wen, M., Cordeau, J.-F., Laporte, G., and Larsen, J. (2010). The dynamic multi-period vehicle routing problem. *Computers & Operations Research*, 37(9):1615–1623.
- Zajac, S. and Huber, S. (2021). Objectives and methods in multi-objective routing problems: a survey and classification scheme. *European Journal of Operational Research*, 290(1):1–25.
- Zhang, Z., Liu, M., and Lim, A. (2015). A memetic algorithm for the patient transportation problem. *Omega*, 54:60–71.

Résumé

Le Problème de Tournées de Véhicules est un problème largement étudié dans le domaine de la Recherche Opérationnelle. Il englobe un large éventail d'applications du monde réel. Récemment, il y a eu un intérêt croissant dans la littérature pour le concept d'équité, qui concerne la répartition équitable de la charge de travail entre les conducteurs. Le défi consiste à trouver une allocation équitable de la charge de travail tout en minimisant l'impact global sur les coûts. Dans ce contexte, nous nous concentrons spécifiquement sur l'application du transport sanitaire. Les contributions de cette thèse résident dans la proposition de modèles et d'algorithmes novateurs qui intègrent des considérations d'équité. De plus, nous visons à évaluer l'impact de l'équité sur les coûts, approfondissant ainsi notre compréhension de cet aspect critique.

Cette thèse est structurée en trois parties principales. La première partie introduit un nouveau problème de tournées qui découle du contexte du transport sanitaire. Pour résoudre ce problème, un algorithme de type "branch-and-price" est développé et évalué sur des instances basées sur le monde réel. Dans la deuxième partie, nous étudions la variante multi-période du problème de tournées initial et nous nous concentrons sur l'intégration de considérations d'équité basées sur la pénibilité du travail. Nous proposons plusieurs méthodes de résolution pour intégrer l'équité et évaluons leur efficacité à l'aide de plusieurs indicateurs de performance. Enfin, dans la troisième partie, nous étendons le problème de tournées initial en introduisant des contraintes d'équité visant à atteindre une répartition équitable des coûts de tournées. Pour relever ce défi, nous proposons et évaluons deux méthodes originales qui intègrent efficacement les contraintes d'équité dans le problème de tournées. En divisant cette thèse en ces trois parties, nous contribuons au domaine de la recherche opérationnelle en abordant des défis spécifiques liés au transport sanitaire, en proposant des algorithmes novateurs pour gérer l'équité et en examinant l'impact de l'équité sur les tournées des solutions.

Mots clés : recherche opérationnelle, problème de tournées de véhicules, équité de la charge de travail, branch-and-price, génération de colonnes

Abstract

The Vehicle Routing Problem is an extensively researched problem within the field of Operations Research. It encompasses a wide range of real-world applications. Recently, there has been a growing interest in the literature on the concept of workload equity, which concerns the equitable distribution of workload among drivers. The challenge is to find a fair allocation of workload while minimizing the overall cost impact. In this context, we specifically focus on the real-world application of non-emergency transportation of patients. The contributions of this thesis lie in proposing innovative models and algorithms that incorporate equity considerations. Additionally, we aim to evaluate the impact of equity on costs, further enhancing our understanding of this critical aspect.

This thesis is structured into three main parts. The first part introduces a novel routing problem that arises from the context of non-emergency transportation of patients. To address this problem, a branch-and-price algorithm is developed and evaluated on real-world based instances. In the second part, we investigate the multi-period variant of the initial routing problem and focus on integrating equity considerations based on the painfulness of work. We propose multiple solution frameworks to incorporate equity and evaluate their effectiveness with several performance indicators. Finally, in the third part, we extend the initial routing problem by introducing equity constraints that aim to achieve fair balancing of routing costs. To tackle this challenge, we propose and evaluate two original solution methods that effectively integrate equity constraints into the routing problem. By dividing the thesis into these three parts, we contribute to the field of operations research by addressing specific challenges of non-emergency transportation of patients, proposing novel algorithms to handle equity and examining the impact of equity on routing solutions.

Keywords: operations research, vehicle routing problem, workload equity, branch-and-price, column generation