Universitat Politècnica de Catalunya

Department of Statistics and Operations Research

The Multi-Depot VRP with Vehicle Interchanges

by

Victoria Rebillas Loredo

PhD Advisor: Maria Albareda Sambola

Disertation of the degree of

Phylosophy Doctor (PhD)

in Statistics and Operations Research

Barcelona, Catalunya, Spain — november 2018

The multi-depot VRP with vehicle interchanges

by

Victoria Rebillas Loredo

To my familly.

Acknowledgments

During my academic life, I have had the pleasure of meeting many great researchers, from whom I have learnt a lot. First, I want to express entire gratefulness to my advisor Mari Albareda, thank you for all the support, patience and for accompanying me through this research adventure. I also want to thank Elena Fernández and Roger Ríos for trusting me and for giving me the opportunity to begin my PhD.

Thank you to the members of the Departament d'Estadística i Investigació Operativa and to the PhD program for all the attentions received. Specially my gratefulness to Jessica and Diana to be with me, both academically and personally. My deepest thanks to Cris Nuñez for you support, teachings and guidance throughout all these years.

I want to thank all my friends in Barcelona for being an important part in this phase of my life. Mainly to "la mia italiana preferita" Valentina for being unconditional, and for being with me whenever I need you. Thanks to "la meva part catalana" Albert, Laia and Oriol for their friendship and to cheer up me all the time. Thanks a lot to Robin, for staying with me all this last important phase, you are my best life coach, "nous sommes une grande équipe". Thank also to families Degoix and Rodríguez for giving me a place among you in such a short time.

My biggest gratitude to my parents Sylvia and Vidal for giving me the best of you always and for teaching me a lot of important values that have been the basis to arrive here, this achievement is also yours. Thanks to all my family specially to my brothers Viridiana and Daniel, grandparents Toya, Cele and Petrita, and uncles Concepción and Jose Luis, for their unconditional support.

This research has been supported by the Mexican National Council for Science and Technology (CONACyT) through the studentship 381539 and The Mexican Secretariat of Public Education (SEP). Besides, it has been partially supported by the Spanish Ministry of Economy and Competitiveness end EDRF funds through grant MTM2015-63779-R (MINECO/FEDER). These supports are gratefully acknowledged.

Summary

Victoria Rebillas Loredo.

Candidate for the degree of Philosophy Doctor in Statistics and Operations Research.

Universitat Politècnica de Catalunya.

Department of Statistics and Operations Research.

Thesis title:

The Multi-Depot VRP with Vehicle Interchanges

Abstract: In real-world logistic operations there are a lot of situations that can be exploited to get better operational strategies. It is important to study these new alternatives, because they can represent significant cost reductions to the companies working with physical distribution. This thesis defines the Multi-Depot Vehicle Routing Problem with Vehicle Interchanges. In this problem, both vehicle capacities and duration limits on the routes of the drivers are imposed. To favor a better utilization of the available capacities and working times, it is allowed to combine pairs of routes at predefined interchange locations.

The objective of this thesis is to analyze and solve the Multi-Depot Vehicle Routing Problem adding the possibility to interchange vehicles at predefined points. With this strategy, it is possible to reduce the total costs and the number of used routes with respect to the classical approach: the Multi-Depot Vehicle Routing Problem. It should be noted that the Multi-Depot Vehicle Routing Problem is more challenging and sophisticated than the single-depot Vehicle Routing Problem. Besides, most exact algorithms for solving the classical Vehicle Routing Problem are difficult to adapt in order to solve the Multi-Depot Vehicle Routing Problem (Montoya-Torres et al., 2015). From the complexity point of view, the Multi-Depot Vehicle Routing Problem with Vehicle Interchanges is NP-Hard, since it is an extension of the classical problem, which is already NP-Hard.

We present a tight bound on the costs savings that can be attained allowing interchanges. Three integer programming formulations are proposed based on the classical vehicle-flow formulations of the Multi-Depot Vehicle Routing Problem. One of these formulations was solved with a branch-and-bound algorithm, and the other two formulations, with branch-and-cut algorithms. Due to its great symmetry, the first formulation is only able to solve small instances. To increase the dimension of the instances used, we proposed two additional formulations that require one or more families of constraints of exponential size. In order to solve these formulations, we had to design and implement specific branch-and-cut algorithms. For these algorithms we implemented specific separation methods for constraints that had not previously been used in other routing problems. The computational experience performed evidences the routing savings compared with the solutions obtained with the classical approach and allows to compare the efficacy of the three solution methods proposed.

Keywords: combinatorial optimization, routing, the Multi-Depot Vehicle Routing Problem, Rich Vehicle Routing Problem, vehicle interchanges, branch-and-cut algorithm.

Resum: En les operacions logístiques del món real es donen situacions que poden ser explotades per obtenir millors estratègies operacionals. És molt important estudiar aquestes noves alternatives, perquè poden representar una reducció significativa de costos per a les companyies que treballen en distribució de mercaderies. En aquesta tesi es defineix el Problema d'Enrutament de Vehicles amb Múltiples Dipòsits i Intercanvi de Vehicles. En aquest problema, es consideren tant la capacitat dels vehicles com els límits de duració de les rutes dels conductors. Per tal de millorar la utilització de les capacitats i temps de treball disponibles, es permet combinar parelles de rutes en punts d'intercanvi predefinits.

L'objectiu d'aquesta tesi és analitzar i resoldre el problema d'Enrutament de Vehicles amb Múltiples Dipòsits, on es permet l'intercanvi de vehicles. Amb aquesta estratègia, és possible reduir els costos totals i el nombre de les rutes utilitzades respecte l'enfocament clàssic: el problema d'Enrutament de Vehicles amb Múltiples Dipòsits. Cal assenyalar que el problema d'Enrutament de Vehicles amb Múltiples Dipòsits és més desafiant i sofisticat que el problema d'enrutament de vehicles d'un únic dipòsit. A més, molts algoritmes exactes per resoldre el Problema d'Enrutament de Vehicles clàssic son complicats d'adaptar per resoldre el Problema d'Enrutament de Vehicles amb Múltiples Dipòsits (Montoya-Torres et al., 2015). Des del punt de vista de la complexitat, el Problema d'Enrutament de Vehicles amb Múltiples Dipòsits amb intercanvis de vehicles és NP-Dur, perquè és una extensió del problema clàssic, que també ho és.

Presentem una cota ajustada de l'estalvi en els costos de distribució que es pot obtenir permetent els intercanvis. Es proposen tres formulacions de programació sencera basades en la formulació clàssica "vehicle-flow" del Problema d'Enrutament de Vehicles amb Múltiples Dipòsits. La primera formulació, degut a la seva grandària i la seva simetria, només permet resoldre instàncies molt petites. Per augmentar la dimensió de les instàncies abordables, es proposen dues formulacions addicionals que requereixen una o vàries famílies de restriccions de mida exponencial. Per això, per tal de resoldre el problema amb aquestes formulacions, ha calgut dissenyar i implementar sengles algorismes de tipus branch-and-cut. En aquests algorismes s'han implementat mètodes de separació específics per a les restriccions que no s'havien utilitzat prèviament en altres problemes de rutes. L'expèriencia computacional realitzada evidencia els estalvis obtinguts comparació amb les solucions corresponents l'enfocament clàssic. Tambè es compara l'eficàcia dels tres mètodes propostes a l'hora de resoldre el problema.

Paraules clau: optimizació combinatoria, enrutament, Problema d'Enrutament de Vehicles amb Múltiples Depòsits, Problema Ric d'Enrutament de Vehicles, intercanvi de vehicles, algorisme de branch-and-cut.

Resumen: En las operaciones logísticas del mundo real hay muchas situaciones que pueden ser explotadas para obtener sistemas de distribución más eficientes. Es importante estudiar estas nuevas alternativas, porque pueden representar una reducción significativa de costes para las compañías dedicadas a la distribución de bienes. En esta tesis se define el Problema de Ruteo de Vehículos con Múltiples Depósitos e Intercambio de Vehículos. En este problema se imponen tanto capacidades de carga en los vehículos como límites de duración en las rutas de los conductores. Para alcanzar una mejor utilización de la capacidad disponible y de los tiempos de trabajo, se permite combinar pares de rutas en puntos de intercambio predefinidos.

El objetivo de esta tesis es analizar y resolver esta extensión donde se permite el intercambio de vehículos en el problema clásico de Ruteo de Vehículos con Múltiples Depósitos. Con esta estrategia es posible reducir los costes totales y el número de rutas usadas respecto al enfoque clásico: el Problema de Ruteo de Vehículos con Múltiples Depósitos. Queda decir que el Problema de Ruteo de Vehículos con Múltiples Depósitos es más desafiante y sofisticado que el Problema de Ruteo de Vehículos con un solo depósito. Además, muchos algoritmos exactos para resolver el problema clásico de ruteo de vehículos son difíciles de adaptar para resolver el Problema de Vehículos con Múltiples Depósitos (Montoya-Torres et al., 2015). Desde el punto de vista de la complejidad, el Problema de Vehículos con Múltiples Depósitos e Intercambio de Vehículos es NP-Duro ya que es una extensión del problema clásico que es ya NP-Duro.

Presentamos una cota ajustada de los ahorros en los costes de ruteo que se pueden obtener permitiendo intercambios. Se proponen tres formulaciones de programación entera basadas en las formulaciones clásicas “vehicle flow” del problema de Ruteo de Vehículos con Múltiples Depósitos. Una de estas formulaciones se resuelve con el algoritmo de branch-and-bound. Debido a la gran simetría que presenta, esta primera formulación es capaz de resolver sólo instancias de tamaño pequeño. Para incrementar la dimensión de las instancias abordadas, proponemos dos formulaciones adicionales que requieren una o varias familias de restricciones de tamaño exponencial. Por eso, con tal de resolver dichas formulaciones hemos diseñado e implementado sendos algoritmos de tipo branch-and-cut. Para estos algoritmos, hemos implementado algoritmos de separación ad-hoc para restricciones que no han sido previamente usadas en otros problemas de ruteo. La experiencia computacional realizada evidencia los ahorros en el ruteo, comparado con las soluciones del enfoque clásico y permite comparar la eficacia de los tres algoritmos propuestos para resolver el problema.

Palabras clave: optimización combinatoria, ruteo, Problema de Ruteo de Vehículos con Múltiples Depósitos, Problema Rico de Ruteo de Vehículos, intercambio de vehículos, algoritmo de branch-and-cut.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, in the current globalized and competitive world, one of the most important challenges for companies is to offer a quality service with efficient strategies to satisfy all the requirements of the users. Besides, demographic explosion and the decreasing profit margins have led the companies to increase their size and operations to satisfy already large and growing demands. As a consequence, efficient decision-making in logistics systems has increased in complexity too.

Operations Research (OR) is a discipline that deals with the application of advanced analytical methods to help making better decisions. In recent years, more and more companies around the world are applying OR tools due to the economic gains and improvements in logistics operations that they can achieve. For example, an important petroleum company in Brazil applied a Decision Support System to support ship scheduling decisions by maritime transportation and they reached a reduction of approximately 7.5% in the operational costs for long-haul transport (Díaz et al., 2014). In the USA a firm that offers trucking and logistics services used an OR solution in rail-based intermodal freight operations. They wanted to maximize driver productivity and minimize the time and miles not directly associated with moving loaded containers to or from rail ramps (Sun et al., 2014). Another kind of solution offered by software companies focuses on vehicle routing vendors; these companies provide personalized solutions in real time according with the vendors necessities (Hall and Partyka, 2016).

In particular, the major operation costs within physical distribution are commonly related to fuel consumption and wages. Therefore, there is a considerable economical interest in finding distribution policies that allow to reduce these costs. To a large extent, these costs can be reduced by making use of efficient distribution routes. Identifying such routes in large distribution systems is not a trivial task, and operations research offer a variety of useful tools to address it successfully. Indeed, the problems that arise in this context have been intensively studied for more than half a century in combinatorial optimization. These problems are referred to as Vehicle Routing Problems (VRPs) (Laporte, 2009).

The VRP can be simply defined as the designing of least-cost delivery routes from a depot to a set of geographically scattered customers, subject to some side constraints (Toth and Vigo, 2014). This problem is among of the most researched ones because of its wide applicability. In real-world situations there are a variety of operational rules and constraints to consider. As a consequence, there are many variants that emerge from the VRP. Among these situations, it is worth mentioning: periodicity, multiple depots, service time windows, capacity constraints, backhauls, pickups and deliveries and split deliveries (Weise et al., 2009; Lahyani et al., 2015).

Taking advantage of the methodological progress and the development of computer technologies, in recent years the research community has focused on new VRP variants which could not be addressed before due to their complexity. These variants include most of the relevant attributes of the VRP that are essential to the routing of vehicles in real-life. Problems with these features are referred to as Rich Vehicle Routing Problem (RVRP) (Lahyani et al., 2015; Cáceres-Cruz et al., 2015). Also, because of the limited supply capacity in each depot and the geographical dispersion of the customers locations, companies often need to operate from several depots (Montoya-Torres et al., 2015). For example, the American electronic commerce and cloud computing company Amazon during these last 20 years has been incrementing its distribution network (MWPVL-International, 2015). The MultiDepot Vehicle Routing Problem (MDVRP) is the VRP extension that studies this realistic situation where more than one depot is considered (Kulkarni and Bhave, 1985; Laporte et al., 1988).

The main constraints in the MDVRP are: each customer is visited once and its demand is fully satisfied, the total load delivered by a vehicle does not exceed its capacity, the total time spent by each driver is below a prespecified limit, and each driver finishes his route at the depot where he started. These conditions often enter in conflict and, in order to satisfy all of them, some of the available resources cannot be fully used.

In this thesis we present the Multi-Depot Vehicle Routing Problem with Vehicle Interchanges (MDVRPVI). It is a new MDVRP variant that can be classified as a RVRP. With the aim of balancing the usage of the vehicle capacities and the route durations, we allow the drivers to interchange their vehicles/containers at predefined meeting points. To this end, their corresponding routes must be synchronized, which entails an additional source of complexity. However, this approach adds flexibility to the operations and it can improve the overall costs taking advantage of the fact that only drivers need to finish tours at their starting depot while vehicle routes can start and end at different depots. Exchange of drivers, especially in the routing of public transport, has been studied in the past (Andersson et al., 1979). However, we want to integrate this strategy within delivery routing problems. The possibility of distinguishing between drivers and vehicles routes can be encountered besides the traditional land transportation (trucks and roads) in other transportation modes (as air, maritime and rail transportation) (Gopalakrishnan and Johnson, 2005). The MDVRPVI is the extension of the MDVRP where this possibility is considered.

The main objective of this work is to study the properties of the MDVRPVI and to propose exact solution methods for this problem. These exact solution methods will be based on different mathematical programming formulations that will also be proposed along the thesis. Given the complexity of this problem, we only expect to be able to solve exactly moderate size problem instances.

The rest of the thesis is organized as follows, of works addressing VRP extensions that share some of the elements of the MDRP-VI is given in 2. In Chapter 3 the definition of the problem and an upper bound on the savings than can be attained with our policy as compared with the MDVRP is presented together with a brief analysis of the effect of the interchange locations. The mathematical formulations proposed to solve the MDVRPVI are explained in Chapter 4. Chapter 5 contains the details of the branch-and-cut algorithms. All the obtained results are shown in Chapter 6. Finally, Chapter 7 provides the summary, conclusions and proposes several venues of the future research.

Chapter 2

Literature Review

This work is focused on a new MDVRP extension, the MDVRPVI in which drivers can switch their vehicles in certain meeting points. According to the definition provided in Lahyani et al. (2015) the MDVRPVI can be classified as a RVRP. The considered features are: vehicle routing and driver scheduling, synchronization between routes, waiting time\handling constraints and routes with closed driver tours. To the best of our knowledge, the problem addressed in this thesis has not been studied in the literature. In this section, we present a general overview of the solution methods for the MDVRP and a review of works focused on problems closely related to the MDVRPVI.

The literature about the MDVRP in not that extensive as in the case of the VRP, given its applicability, many different techniques to solve it and its variants have been proposed in the literature.

The first exact algorithms for the MDVRP were presented in Laporte et al. (1984) and Laporte et al. (1988). They developed a branch-and-bound algorithm, that allowed them solving instances of up to 80 nodes. More recently, Baldacci and Mingozzi (2009) developed an unified exact algorithm capable to solve different VRP extensions; among these, the MDVRP. This algorithm is based on a set partitioning formulation, and uses three types of bounding procedures based on a LP-relaxation and on the Lagrangean relaxation of the mathematical formulation. The computational results show that instances involving up to 199 customers can be solved. Braekers et al. (2014) present an exact approach for a general heterogeneous dial-a-ride problem with multiple depots. Three and two-index formulations are discussed and a branch-and-cut algorithm for the standard dial-a-ride problem is adapted to exactly solve problem instances with about 50 requests. Contardo and Martinelli (2014) formulated the MDVRP with length constraints based on a vehicle-flow and a set-partitioning formulation. Several valid inequalities are used to strengthen both formulations and a new family of valid inequalities that forbid cycles of an arbitrary length are included. Their exact method has different stages. The three main components are variable fixing, column-and-cut generation and column enumeration. This new algorithm is able to solve MDRVP benchmark instances that had never been solved before, some having over 200 customers. Lahyani et al. (2018) developed five different mathematical formulations to solve the multi-depot fleet size and mix vehicle routing problem (MDFSMVRP). They used a branch-and-bound and branch-and-cut as solution algorithms, and compared the bounds of these formulations with classical benchmark instances, achieving better results with the commodity flow formulations and the capacity-indexed formulation.

Due to the complexity of this problem and its variants, together with the increasing popularity of heuristics and metaheuristics, several algorithms of this type have been proposed in the literature for the MDVRP and some of its extensions. The first heuristic method is presented in Min et al. (1992) to solve the MDVRP with backhauling, the main idea of their algorithm is based on a problem decomposition. Among recent works, Afshar-Nadjafi and Afshar-Nadjafi (2017) present a constructive heuristic for the time-dependent MDVRP where an heterogeneous fleet, hard time windows and limitation on the maximum number of the vehicles in each depot are considered. Vidal et al. (2014) developed an unified metaheuristic to solve some variants of the MDVRP with and without fleet mix. The used approach is an hybrid genetic algorithm with iterated local search and dynamic programming. An hybrid granular tabu search algorithm was developed by Escobar et al. (2014). This algorithm is based on a heuristic framework previously introduced by the authors for the solution of the Capacitated Location Routing Problem. With the computational experiments on benchmark instances from the literature, they enhance several best solutions obtained with previously published methods and find new best solutions. A problem arising in the last mile distribution of e-commerce is studied in Zhou et al. (2018). In this problem, the first level routing is to design the routes transporting the customers demand from the depot to a subset of satellites. The second level is the routing from the satellites to the customers. Besides, customers may provide different delivery options, allowing them to pick up their packages at intermediate pickup facilities. With all the above this is a complex problem, since there are a lot of interconnected decisions this is a complex problem, so they developed a hybrid multi-population genetic algorithm. A real word instance as well as randomly generated instances were tested.

As mentioned before, the routing policy proposed in this thesis has not been studied before within VRPs. So, we review the literature on routing works with the main characteristics of the MDVRPVI.

In our problem, we differentiate vehicles routes and drivers routes. In the literature there are works considering this fact. To the best of our knowledge, one of the most similar problems to the MDVRPVI studied in the literature is the simultaneous vehicle and crew routing and scheduling for partial-and full-load long-distance road transport proposed in Drexl et al. (2013). In this problem, there is an heterogeneous float of trucks and homogeneous drivers, and thats it, all drivers are able to use any available truck. Besides the typical depots and pick-up and delivery locations, relay stations are considered where drivers can exchange a truck, take a break or take a small shuttle van to another relay station (unlimited). The objective was to minimize the total routing costs, while satisfying the EU regulations. In this work, in contrast to the MDVRPVI, drivers and trucks have to finish at the same initial depot (as in the traditional MDVRP). This has to be satisfied at the end of the planning horizon that is six days. The proposed solution procedure is a two stage heuristic. The first stage consists in solving a Pickup-and-delivery problem (PDP) with time windows and relay stations. In the second stage, the authors solve a vehicle routing problem with time windows and multiple depot and, after that, they integrate both solutions. With all the characteristics considered, this kind of problems are intractable by means of exact algorithms, so that heuristic algorithms is one of the most common tools to solve them. We believe that it is important and useful to have a thorough knowledge of more and more realistic problems and to be able to solve them exactly. This motivates us to study the MDVRPVI, which already has some of the features of such real problems, but it is stylized enough to allow for theoretical analyses and exact solution methods.

Other problems that, although being different are closely related to the MDVRPVI are reviewed next.

Domínguez-Martín et al. (2017) and Domínguez-Martín et al. (2018), solved a problem motivated on air transportation where, as in the MDVRPVI drivers must finish their routes at their home depot, but vehicles need not. In these problems the daily planning of flights, airplanes and crew has to be made, and the crew members are allowed to travel as passengers in flights too, in order to reach their depot. To avoid overnight costs, the flights assigned to each aircraft must be such that it starts the day in one hub airport and ends in the another one. The largest difference between these two problems was in the number of visits. The first problem has been modeled as a scheduling problem where each task must be performed by a unique operator and a unique machine, while in the second one, some locations may be visited by several vehicles and drivers. The authors proposed integer programming formulations for both problems, and branch-and-cut algorithms were used to solve it. We take this idea of designing different routes for the vehicles and for the crew, so usual in air transportation, and apply it in road freight transportation.

Also in the context of air transportation, but now motivated by situations arising in humanitarian and military logistics, (Lam et al., 2015) present the Joint Vehicle and Crew Routing and Scheduling Problem. In this work, The authors highlight how important is to consider the crew routing besides the vehicle routing, because in their problem, limitations on the operating times of the crew need to be considered. In this problem crews are able to interchange vehicles at different locations (which are the same places to serve) and to travel as passengers before and after their operating times. Moreover, if there is a crew interchange, vehicles need to be synchronized. The problem was modeled with a constraint programming formulation and to solve it, they implemented a Large Neighborhood Search, that explored both vehicle and crew neighborhoods.

In general terms, crew scheduling problems are problems that emerge in the planning of mass transit companies, where minimal cost bus driver schedules are sought, considering agreement with labor unions and the transportation schedules Mesquita and Paias (2008) integrated the vehicle scheduling to this problem, so, assignments of vehicles to timetabled routes, as well as of drivers to vehicles are made. As for the solution approach, two mathematical formulations were developed. In the first one, a multi-commodity network flow model for the vehicle scheduling was combined with a set partitioning model for the crew scheduling. In order to provide more flexibility to the model, in the second formulation the authors replaced a subset of set partitioning constraints by set covering constraints. Note that, in problems with the previous features routing of the drivers/vehicles is not performed.

When multiple depots are established in large cities or areas the traditional operation rule is to serve from each depot the customers within a fixed area. In e-commerce logistics, internet orders are flexibly assigned between different distribution centers to share depot resources. Seeking to serve efficiently the customers and to reduce the fuel consumption, this logistic operation has been increasing its application. Motivated by this fact, Li et al. (2018) presented the MDVRP under shared depot resource. They evaluated the benefit ratios between unshared and shared depots on route distance and fuel consumption and showed that they can be up to 2. In this problem, the vehicle is not required to return to the depot from which it started because an information system, allows to manage and position drivers and vehicles. So, vehicles are able to share parking spaces in all distribution centers, and drivers can go home through a convenient and fast subway system after they get off their works.

In order to take customer sharing among a set of carriers, Fernández et al. (2018) developed mathematical programming formulations and a branch-and-cut algorithm to solve a problem with this feature. From delivery services are offered by the carriers multiple depots spread in the city. This produces numerous simultaneous trips on common areas and partial loads. With this new approach, the authors propose to take advantage of collaboration among carriers who must serve common customers in the same time horizon. Collaborating carriers could serve part of the demand for other carriers, as well as combining or exchanging customers orders or requests. On the benchmark instances they could solve optimally (depending on the number of customers and their locations), they obtained benefits that a range from 6.5% to 25.5% as compared with a non collaboration scheme.

Another relevant issue in the MDVRPVI is the need to synchronize routes. Synchronization of routes has been considered in some previous works. Transshipments inside the network in a PDP were studied in Rais et al. (2014). In this problem there are some transshipment locations (the same customers locations) where vehicles may transfer and adjust their loads. The locations of the vehicle origin and final depots were predefined. Drivers can finish their working day in a different depot. Moreover, at transshipment locations, they may switch their vehicles, get release time to adjust to policy-related matters or be replaced by fresh or rested drivers. A mathematical programming formulation with three indexed variables was presented. The main difference of the MDVRPVI than our problem is that this is a PDP, which adds more flexibility to deal with vehicles capacities.

To some extent, synchronization also appears in cross-docking problems. In cross-docks, freight is unloaded from incoming vehicles and directly transferred to outbound vehicles. Therefore, planning cross-dock operations involves the synchronization of the routes of both types of vehicles. This type of problems has been studied since the nineties. A recent example is Dondo and Cerdá (2015), where the authors consider a problem with heterogeneous fleet.

The VRP with Trailers and Transshipment is a problem that arises in raw milk collection at farmyards (Drexl, 2013) where route synchronization is also required. Given an heterogeneous fleet stationed at one or more depots, all the given demand has to be collected. The fleet is integrated by lorries that can be temporarily enlarged with trailers, and load transfers can be carried out at transshipment locations such as parking places or customer premises. This kind of VRP variation exhibits additional synchronization requirements in regard to spatial, temporal, and load aspects.

The synchronized routing of active and passive means of transport arising in container and conducting situations was solved in Meisel and Kopfer (2014). In this problem, pick up and delivery requests have to be fulfilled with active and passive means. To perform the task, active means go to the passive means to transport them through the locations, where they are dropped off and unloaded. As in the MDVRPVI, in this work an important challenge is the synchronization of the routes. Considering the fact that passive means are not independent to travel, each travel must be synchronized. In this case time windows and compatibility constraints between requests of passive and active means need to be taken into account. They propose a mathematical formulation but, as this problem is NP-hard, for non-trivial instances an Adaptive Large Neighborhood Search metaheuristic was implemented. This problem is similar to the capacitated truck-and-trailer routing problem solved in Bartolini and Schneider (2018) using a two-commodity flow formulation. Due to their inaccessibility, there are some customers that cannot be served with a trailer. So, with a restricted fleet of trucks and trailers from a central depot, the composite vehicles can temporally park their trailer at any accessible customer and transfer load from it. This action allows to perform a subtour serving the inaccessible customers and then come back to the trailer to resume the journey. Huber and Geiger (2014) consider a similar problem. They consider different times associated with attaching/detecting or swapping trailers. Since the number of such actions is not limited, it is important to handle correctly synchronization times at swap locations. A maximum driving time cannot be exceeded, the truck must return to the depot with the same trailer it had started with, and it is not permitted to transfer load partially or completely to another trailer. A solution method, an Iterated Variable Neighborhood Search was employed. To diversify the considered local optima, the authors design some different neighborhoods with intra and inter-tour operators specific to this problem.

There are other problems where routes are allowed to stop at intermediate points. For example in Crevier et al. (2007) the authors solved an extension of the MDVRP, where vehicles may be replenished at intermediate depots along their routes. Similarly, Angelelli and Speranza (2002) presented an extension of the Periodic Vehicle Routing Problem also including this possibility.

In both cases, routes may finish at an intermediate depot which is different from the starting point. In Crevier et al. (2007) a set partitioning formulation allowed to solve only small instances. In both cases, larger instances are tackled by means of heuristic methods. Crevier et al. (2007) make use of the adaptive memory principle and create solutions by combining elements of previous ones, whereas Angelelli and Speranza propose a Tabu Search algorithm. In the same line Baldacci et al. (2016) solved the Vehicle Routing Problem with Transhipment Facilities. This problem emerged from a company operating in the production and distribution of non-perishables. The objective was to minimize the total distribution costs optimizing inter-dependent decisions. These decisions were selecting transshipment facilities, allocating customers to these facilities and the design of vehicle routes. They proposed two integer-programming formulations and developed and efficient algorithm to solve large instances both, from the literature as from the real-world.

Another particularity of the MDVRPVI is that some vehicles may finish at different depot than the starting one. Problems involving either routes of this type, or routes that are considered to be finished when the last customer is reached are known in the literature as Open Vehicle Routing Problems (OVRP). We can encounter real-life applications as school bus routing (Bektaş and Elmastaş, 2007) or ambulance routing (Tlili et al., 2017). We can highlight the OVRP with Decoupling Points (Atefi et al., 2018). This is an application motivated by companies transporting their products over very long distances. They proposed to use a decoupling point of the route, where it is possible to use more than one carrier to perform a delivery. The idea is that the first carrier leaves from the depot, performs part of the deliveries and drops off all remaining load at one of the decoupling points. Then, the rest of the customers have to be served by a second carrier. An Iterated Local Search algorithm for this problem is developed. The authors tested their algorithm on real data and showed that using decoupling points is a profitable policy.

A quite general problem involving different types of constraints to fulfill diverse typical requirements is considered in Ceselli et al. (2009). In this work, the authors developed a model with a software-planning tool for distribution of logistics companies. To solve the problem, they employed a column generation algorithm that required solving a particular resource-constrained elementary shortest-path problem as the pricing problem. Another OVRP, but also considering time windows was presented by Li et al. (2014). An integer programming formulation was used to solve small instances, and for large instances an hybrid genetic algorithm with adaptive local search was implemented.

In Table 2.1 we summarize the characteristics of the works addressing the most similar problems to the MDVRPVI.

		Solution method			**VRP Variant**				**Features**					
Reference	**Problem Application**	Exact	Heuristic	Description	VRP	MDVRP	SDVRP	PDP	Heter. fleet	Time wind.	Transfer	Synchr.	Different initial & final depot	Intermediate stops
Angelelli and Speranza (2002)	Waste collection		*	Tabu search				*						*
Crevier et al. (2007)	Grocery distribution	*	*	Algorithm combining integer programming, the adaptive memory principle and tabu search		*							*	*
Ceselli et al. (2009)	Good deliveries	*		Column generation		*			*	*			*	
Drexl et al. (2013)	Transportation under EU regulations		*	Two stage heuristic		*		*	*		*	*	*	
Drexl et al. (2013)	Raw milk collection	*		Branch-and-cut and branch-and-cut and price algorithms	*				*	*	*	*		
Huber and Geiger (2014)	Transport deliveries		*	Iterated Variable Neighborhood Search					*	*	*	*		
Li et al. (2014)	Collaborative Transportation		*	Hybrid genetic algorithm with adaptive local search		*				*			*	
Meisel and Kopfer (2014)	Container and conducting situations		*	Adaptive large neighborhood search			*		*			*		*
Rais et al. (2014)	Transport systems	*		Three index formulation and model constraints for some particular cases				*	*		*	*	*	*
Lam et al. (2015)	Humanitarian and military logistics		*	Large neighborhood search	*					*	*	*		*
Atefi et al. (2018)	Food routing	*	*	Iterated Local Search	*						*			*
Domínguez-Martín et al. (2018)	Local air-traffic	*		Branch-and-cut		*					*	*	*	*
Li et al. (2018)	E-commerce		*	Hybrid genetic algorithm with adaptive local search		*						*	*	

Table 2.1: Related work

Chapter 3

The Multi-Depot Vehicle Routing Problem with Vehicle Interchanges

In this Chapter, we present the definition, assumptions and characteristics of the MDVRPVI together with an upper bound on the relative savings that can be obtained by allowing vehicle interchanges. This bound will be proven to be tight by means of an asymptotic example.

3.1 Problem definition

We are given a set of geographically scattered points divided into customers, depots and interchange locations. Transportation costs and travel times (or distances) between any pair of such points are also given (the triangle inequality is assumed). Each customer has a certain demand, and each depot an available fleet of capacitated vehicles to serve all the customers. There is also a limit on the duration of the drivers routes. The MDVRPVI consists in determining a set of vehicle and driver routes which minimize the total distribution costs (total transportation plus vehicle utilization costs) so that: each customer is serviced exactly once by a vehicle, the total demand of each vehicle route does not exceed the vehicle capacity, the total driver working time does not exceed the maximum available time and each driver finishes at the same depot where the started. In the MDVRP routes are designed so that each vehicle and its associated driver must start and end at the same depot.

As opposite, in our problem the vehicle may not end at the depot where it departed from, but drivers must start and finish at their home depot. To do that, a couple of routes can be synchronized and stop at a certain interchange location where drivers exchange their vehicles or containers. We will refer to a pair of such routes as "exchanged routes". With this strategy it is possible to combine better the vehicle capacities and the allowable driving times. We want to take advantage of this fact and hence reduce the total routing costs. We assume that at most one interchange per route is permitted and also that at most one interchange per interchange location is performed. This last assumption can be satisfied by replicating interchange locations if more interchanges are allowed. No waiting times at the interchange locations are imposed, since we do not force all vehicles to start their journey at the same time, and, therefore, they are assume to start in a coordinated way, so that they arrive to the interchange location at the same time. Within this thesis, we do not distinguish between the time required to traverse an edge (used in the driving time constraints), and the cost to do it (needed for the objective function). However, all the arguments used here can be extended to the case where they are two distinct sets of values. Note also that the possible time required to perform an interchange has not been explicitly defined. In most cases this time is negligible. However, if an application requires including such times, they can be simply dealt with by adding them to the times associated with arcs ending at an interchange location. The MDVRP is classified as NP-Hard (Laporte et al., 1988). Since the MDVRPVI contains as a particular case the MDVRP (the set of interchange sites is empty), the MDVRPVI is NP-Hard too.

To illustrate a MDVRPVI solution consider the situation depicted on top of Figure 3.1 where four customers (circles), two depots (squares) and an interchange location (triangle) are placed in a line. The time required to traverse the whole segment between A and B equals the maximum driving time, T, the vehicle capacity is Q, and the customers demands are those given inside the circles representing them. In this case, if it is only allowed to move along the horizontal line, the optimal MDVRP solution is to serve from depot A the customers with demand $Q-1$ in two separate routes and use another tour from depot B for serving the other two customers. The value of this optimal solution is $Z(MDVRP) = 2(\frac{T}{2} - 2\epsilon) + 2(\frac{T}{2} -$

$\epsilon)+2(\frac{T}{2}-\epsilon) = 3T-8\epsilon$. If interchanges are allowed, the optimal solution is composed of two routes, each starting at one depot, collecting the demand of one small and one big customer and finishing at the other depot. For making this possible, the two routes must meet at the interchange point (the triangle) so that drivers can switch their vehicles and finish their journey at their home depot. This solution is feasible for both drivers and vehicles, and the total cost is $Z(MDVRPVI) = 2T$. Thereby, for arbitrarily small ϵ values, in this example, $Z(MDVRP) \rightarrow \frac{3}{2}Z(MDVRPVI)$. In Figure 3.1 both solutions are shown.

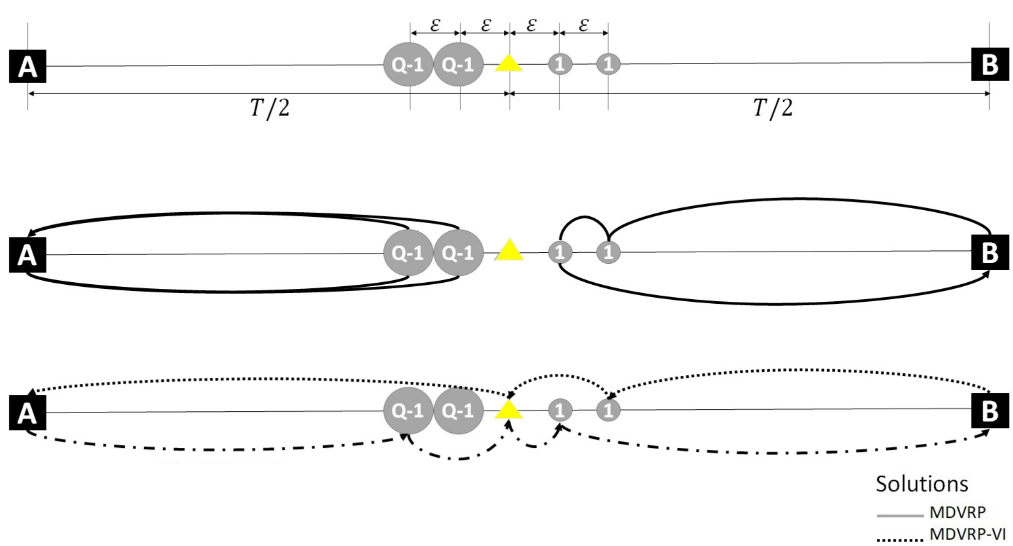
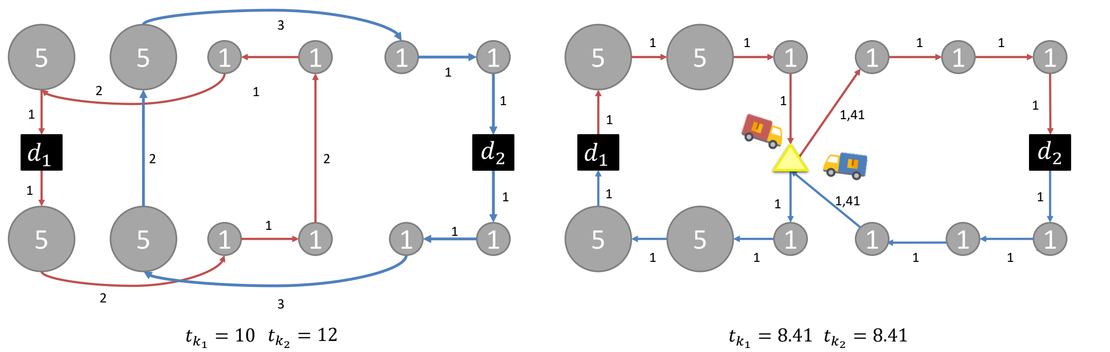


Figure 3.1: Example of MDVRPVI solution.

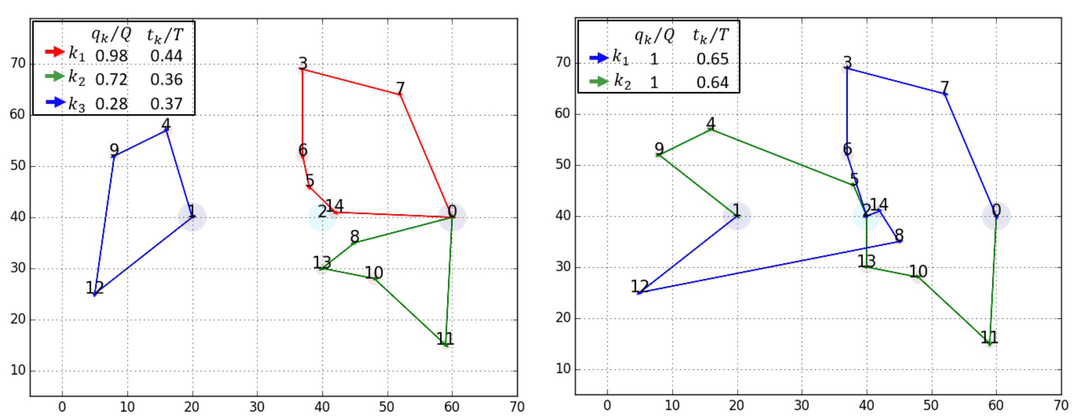
In Figure 3.2 we provide another example to illustrate the possible advantage of a MDVRPVI solution with respect to the MDVRP solution. In this instance customers, depots and an interchange point are located on a 3×5 grid and Euclidean distances are considered. The vehicles capacity is $Q = 14$ and $T = 12$ is the maximum available time. As we see in Figure 3.2(a) the MDVRP optimal solution consists of two routes, with a total length $Z^* = 22$. On the other hand, in the MDVRPVI solution (3.2(b)) drivers arrive to the interchange point (triangle) and with this strategy, each of the routes has length 8.41. As a result $Z^* = 16.82$ is the optimal value, which represents a 23.5% reduction with respect to the MDVRP solution.



(a) MDVRP solution (b) MDVRPVI solution

Figure 3.2: Example MDVRP versus MDVRPVI.

Additionally to the transportation cost reduction, in some cases the MDVRPVI gives the possibility to reduce the number of routes needed as it is the case in the synthetic example of Figure 3.1. We provide a last example. Consider the instance depicted in Figure 3.3, with depots at nodes 0 and 1 an interchange location at node 2, and customers at nodes $\{3, ..., 14\}$. Euclidean distances are used. The corresponding demands are as follows $\{13, 4, 7, 15, 6, 4, 9, 15, 9, 4, 15, 17\}$ vehicles capacity is $Q = 59$ and the route duration limit $T = 200$. The optimal MDVRP solution, with value $Z^* = 327.077$ uses three vehicles whereas the optimal MDVRPVI solution, with value $Z^* = 291.217$ requires only two. Besides, we can observe with the load ratios (q_k/Q) that all the vehicles capacity is used. About the time ratios (t_k/T), our policy in this case reflects a better balance between drivers.



(a) MDVRP solution using 3 routes (b) MDVRPVI solution using 2 routes

Figure 3.3: Example where the MDVRPVI uses one route less than MDVRP.

Effect of the location of the interchange points

We are aware of the fact that the interchange points locations can represent a significant variation in the possible savings to obtain by using the interchange policy. As we have defined the MDVRPVI, the problem is already very complex, so unfortunately optimizing this location is not within the scope of this thesis. Nonetheless, in order to know the behavior in the savings that could be achieved, in one particular instance we tested some different interchange locations around a specific region to observe the variation of the obtained gains.

Figure 3.4 shows an instance with two depots (d_1 and d_2), 13 customers (with size proportional to the demands), and one interchange point, for which three different locations have been tested (i_1, i_2 and i_3). Again, Euclidean distances are used. In all of them, the shape of the solution is the same, but the savings with respect to the MDVRP are different. The largest one is 9.10%, obtained with the point i_1, the second one is i_3 with 7.34% and finally i_2 gives the smallest saving, 6.15%. So we can see that even small variations on the interchange point location can have a large effect on the obtained gains.

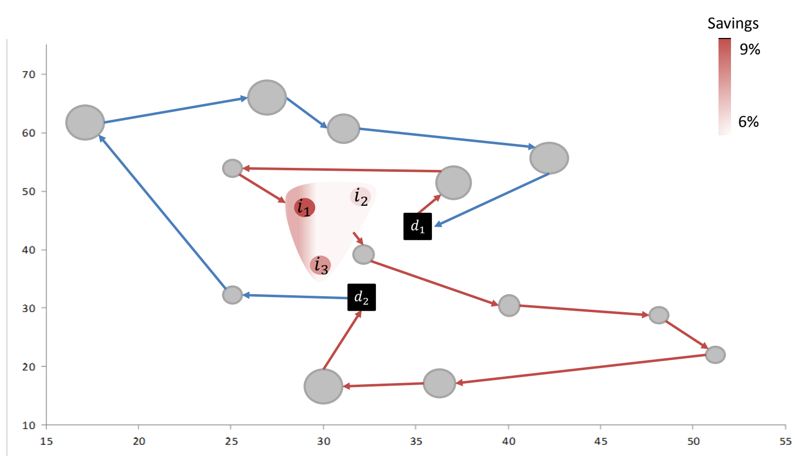


Figure 3.4: Different interchange point locations tested and their corresponding savings.

3.2 Analysis of the maximum achievable savings

In this section we provide a theoretical tight bound on the savings that can be attained by using this policy, as compared with the classical MDVRP. In the classical MDVRP each driver must finish his route at the depot where he started (closed route). Due to these constraints the driver route duration limit and vehicles capacity may be not completely used. As mentioned above, in the MDVRPVI, we allow that pairs of vehicle routes switch their respective depots so that, if the corresponding drivers also switch, they can finish their journey at their home depot. As it has been shown in the previous examples, in some occasions this may help balancing the use of the available resources.

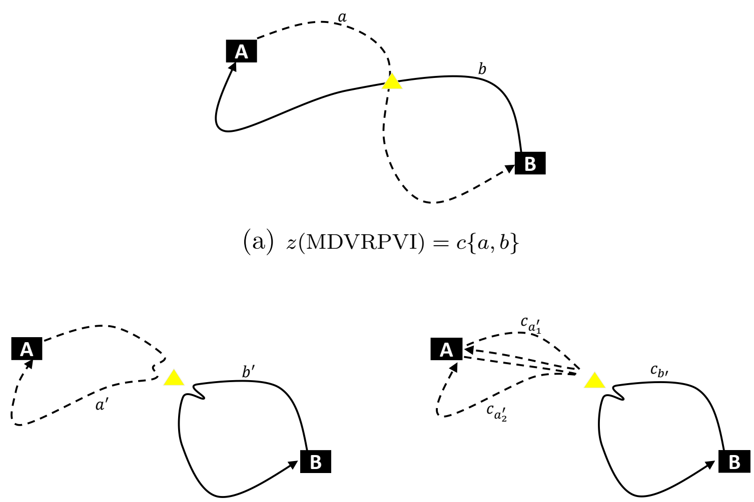
To take advantage of combining vehicle loads and drivers operation times, a new design of routes is performed and a significant routing cost reduction can be attained. We want to bound the largest value that this reduction can take. Starting from a fixed instance, we denote the costs of the optimal solutions to both problems by $Z(MDVRP)$ and $Z(MDVRPVI)$, respectively. We next analyze the ratio between these values.

In that follows, we refer to a route which is feasible with respect to the capacity constraint as a Q-feasible route. Analogously, a route that is feasible regarding the time constraint is called T-feasible. Keep in mind that the only assumption made on the distances is that they satisfy the triangle inequality.

Theorem 1: $\frac{Z(MDVRP)}{Z(MDVRPVI)} < 2$, and this bound is tight.

The simplest case where exchanged routes can appear is an instance with two depots and one interchange point. We will focus on this case for simplicity. All the ideas used here naturally extend to instances with more depots and interchange points.

Let us consider an MDVRPVI instance for which the optimal solution contains two exchanged routes a, and b, as depicted in Figure 3.5(a). After serving some customers each, drivers meet at the interchange location (triangle) where they switch their vehicles. Then, they can serve the rest of the customers and finish in their home depot. Now, from the previous MDVRPVI solution we will build a feasible MDVRP solution and compute its cost. This solution is initially formed by routes a' and b'. Each of them follows the previous drivers routes, but skipping the interchange point (see Figure 3.5(b)). These will be further modified if they are not feasible. We will show in what follows that the value of the new solution is at most $2Z(MDVRPVI)$. Since the new solution will be feasible for the MDVRP, its value will bound above $Z(MDVRP)$.



(a) $z(\text{MDVRPVI}) = c\{a, b\}$

(b) MDVRP solution?

(c) $z(\text{MDVRP}) \leqslant c\{a'_1, a'_2, b'\}$

Figure 3.5: From a MDVRPVI solution to a MDVRP solution.

1. **If routes a' and b' were feasible**, taking advantage of the triangle inequality, the solution $\{a', b'\}$ would be optimal for the MDVRPVI, because it has a total driving distance not larger than that of $\{a, b\}$. In this case, $Z(MDVRP) = Z(MDVRPVI)$.

2. **Even if a' and/or b' are infeasible, note that they must be both T-feasible:** due to the length constraints, and the triangle inequality, route a' is not longer than the route of the driver that started at depot A (the same for b'). Therefore, the only unfeasibilities of a' and/or b' are related with the capacity constraints.

   - **If a and b are Q-feasible then at least one of a' or b' is Q-feasible too:** Let Q_1 and Q_2 be the aggregate demand of the customers in the first and second path of route a, respectively, and let Q_3 and Q_4 be the aggregate demand of the customers in the first and second path of route b, respectively. If a' is infeasible it must be because $Q_1 + Q_4 > Q$, if b' is infeasible, $Q_2 + Q_3 > Q$. Then, if both were infeasible, the total demand would satisfy $Q_1 + Q_2 + Q_3 + Q_4 > 2Q$ and this cannot happen if a and b were Q-feasible. Therefore, at most one of the routes a' and b' can be infeasible. We assume, without loss of generality, that a' is not Q-feasible.

If the Q-infeasible route is a', it can be split into two Q-feasible routes a'_1 and a'_2 as in Figure 2.c. Route a'_1 starts at the depot, follows route a' until the last customer before skipping the interchange point, and then goes straight back to the depot. Route a'_2 resumes route a' from the first customer not visited by route a'_1. By the triangle inequality neither a'_1 nor a'_2 are longer than a', and, by construction, both are Q-feasible since each is part of one of the original Q-feasible routes a and b and demands are non negative. Therefore routes $\{a'_1, a'_2, b'\}$ define a feasible MDVRPVI solution.

Next, we compare the transportation cost of the solution built above with the cost of the original one. Since $\{a, b\}$ is the optimal MDVRPVI solution, and $\{a'_1, a'_2, b'\}$ is a feasible MDVRP solution, any upper bound on $\frac{c(\{a,b\})}{c(\{a'_1,a'_2,b'\})}$ will be an upper bound on $\frac{Z(MDVRP)}{Z(MDVRPVI)}$ for this instance. Indeed, by the triangle inequality, the potentially feasible solution $\{a', b'\}$ described in point 1 will never be more expensive than $\{a'_1, a'_2, b'\}$, so that it is not necessary to consider it further in this analysis.

Figure 3.5 shows the costs involved in both solutions. We split the cost of route a into costs $c_a^1 + c_a^2$ to manage the costs in the route before and after the interchange point, respectively. Analogously, the cost of route b has been split into $c_b^1 + c_b^2$. Similarly, we denote with $c_{a'_1}$, $c_{a'_2}$ and $c_{b'}$ the costs of routes a'_1, a'_2 and b', respectively. Therefore,

$$\frac{c(\{a,b\})}{c(\{a'_1, a'_2, b'\})} = \frac{c_a^1 + c_a^2 + c_b^1 + c_b^2}{c_{a'_1} + c_{a'_2} + c_{b'}} \tag{3.1}$$

Now, we have two possibilities: either $\mathbf{c_a^1} + \mathbf{c_b^2} \leq \mathbf{c_b^1} + \mathbf{c_a^2}$ or $\mathbf{c_a^1} + \mathbf{c_b^2} > \mathbf{c_b^1} + \mathbf{c_a^2}$.

Case 1: considering that $\mathbf{c_a^1} + \mathbf{c_b^2} \leq \mathbf{c_b^1} + \mathbf{c_a^2}$ then

$$c_{b'} + c_{a'_1} + c_{a'_2} \leq 2(c_a^1 + c_b^2) + c_b^1 + c_a^2 \leq \frac{3}{2}(c_a^1 + c_b^2 + c_b^1 + c_a^2) = \frac{3}{2}Z^R(MDVRPVI).$$

where Z^R stands for the routing cost of the optimal MDVRPVI solution.

Case 2: suppose that $\mathbf{c_a^1} + \mathbf{c_b^2} > \mathbf{c_b^1} + \mathbf{c_a^2}$ then

$$c_{b'} + c_{a'_1} + c_{a'_2} \leq 2(c_a^1 + c_b^2 + c_b^1 + c_a^2) = 2Z^R(MDVRPVI). \quad (3.2)$$

In Case 2 we can see that the delivery cost never reduces to less than 50% if interchange routes are allowed. As a consequence, in both cases the ratio (3.2) is bounded by 2. The same reasoning can be repeatedly applied if in a solution there are more than one pair of exchanged routes. As for the vehicle utilization costs, note that in the MDVRP solutions built as above, the number of vehicles is increased, at most, by 50%. Therefore, for any instance, $\frac{Z(MDVRP)}{Z(MDVRPVI)} < 2$.

To see that the bound is tight, Figure 3.6 shows a synthetic example where allowing the combination of two routes results in cost savings that, asymptotically, reach this bound. Given $\in \mathbb{N}$ consider the instance with capacity $2n+2$ customers with $Q = 2n+1$, a huge value of the driving time limit, network distances, and the demands shown in the figure. The optimal MDVRP solution is composed by two routes rooted at depot A delivering all the customers with demand 2 (one above and one below) and one route rooted at depot B serving the remaining two customers. Allowing interchanges, the optimal solution is composed by two routes with similar shape: starting in one depot, they deliver n customers with demand 2 and one with demand 1, going to the interchange location and finishing the route in the opposite depot (see Figure 3.6). So, the value of Z(MDVRP) is equal to $(4n+3)\epsilon$ and Z(MDVRPVI) is $2(n+4)\epsilon$. As n grows, the ratio $Z(MDVRP)/Z(MDVRPVI) \rightarrow 2$.

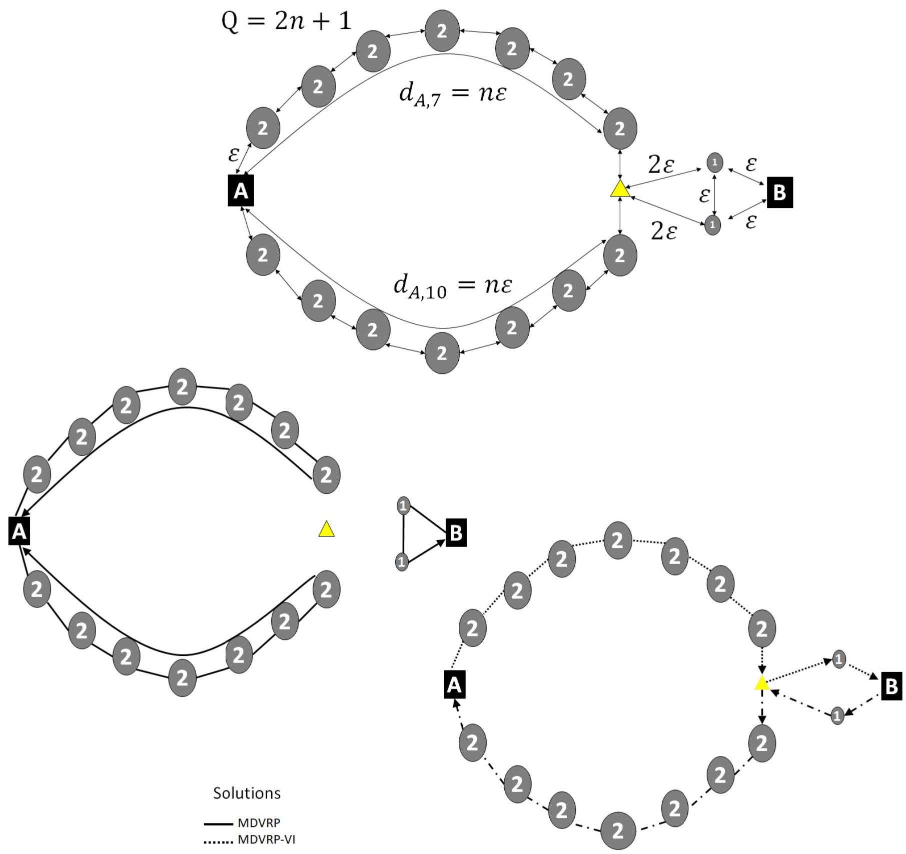


Figure 3.6: Example of Theorem 1.

Chapter 4

Mathematical programming formulations

Next we formally define the MDVRPVI and state the notation used in the reminder of this thesis. Consider a weighted and directed graph $G = (V, A)$, where the set of nodes are partitioned as $V = D \cup C \cup I$ and the set of arcs is $A = (V \times V) \backslash \{(D \times D) \cup (I \times I)\}$. D is the set of depots, C is the set of customers and I the set of predefined interchange locations, where a couple of drivers can meet to interchange their vehicles/containers. The set of arcs contains all possible arcs except the ones connecting directly two depots, or two interchange points. Regardless the fact that the considered transportation costs are symmetrical or not, it is convenient to use directed formulations for the MDVRPVI. This is because some paths can affect feasibility in the solutions. Especially where there are exchange routes, due to the sense of the arcs, determine charge of the different drivers/trucks and the interchange times. Moreover, this notably simplifies the modeling of route synchronization. With each customer $j \in C$ is associated a demand q_j to be satisfied. Each arc has its corresponding cost, (distance or travel time) c_{ij} . We assume that demands and distances are non negative ($q_j, c_{ij} \geq 0$) and distances satisfy the triangle inequality. There is a fleet of homogeneous vehicles K, each one with capacity Q and a fixed vehicle usage cost g. T represents the maximum driver working time, that is composed by the sum of the traveling time along the edges used and the synchronization waiting time, if the routes are switched.

4.1 3-INDEX FORMULATION

The first and most intuitive mathematical formulation to the MDVRPVI is a three-index vehicle-flow formulation, which has been extended from the formulation presented by (Kulkarni and Bhave, 1985) to solve the MDVRP. To this end, we define the following decision variables. For each arc (i, j) and each vehicle k we define the binary variable x_{ij}^k that is equal to 1 if route k uses arc (i, j). For each $k \in K$, $d \in D$ binary variables o_{dk} and f_{dk} determine whether route k is starts or ends at depot d, respectively. We are assuming only one pair of exchanged routes at each interchange location, so the binary variables $e_{kk'}^r$ are equal to 1, if routes $k, k' \in K$ interchange their vehicles at interchange location $r \in I$. These variables can be defined as integer, in the case that more than one pair of exchanged routes is allowed. Finally, we have the non negative continuous variables: ℓ_i as the time from the beginning of the working day until the moment when a vehicle departs from $i \in C \cup I$ and s_i which is the driving time from node i to the end of the route that visits it. In the case of interchange points, ℓ_i is the maximum of these values among the routes that visit it. With this notation, the MDVRPVI can be formulated as:

$$\text{minimize } g \sum_{o \in D} \sum_{k \in K} o_{dk} + \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \tag{4.1}$$

subject to

$$\sum_{j \neq i} x_{ij}^k = \sum_{j \neq i} x_{ji}^k, \qquad i \in C \cup I, k \in K, \tag{4.2}$$

$$\sum_{i \in D \cup C} \sum_{k \in K} x_{ir}^k = 2 \sum_{k,k' \in K} e_{kk'}^r, \qquad r \in I, \tag{4.3}$$

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \qquad i \in C, \tag{4.4}$$

$$\sum_{i \in C} q_i \sum_{j \in V} x_{ij}^k \leq Q, \qquad k \in K, \tag{4.5}$$

$$\ell_i + s_i \leq T, \qquad i \in C \cup I, \tag{4.6}$$

$$\ell_j \geq \ell_i + c_{ij} - (T + c_{ij})(1 - \sum_{k \in K} x_{ij}^k), \qquad i \in C \cup I, j \in C, \tag{4.7}$$

$$\ell_j \leq \ell_i + c_{ij} + (T + c_{ij})(1 - \sum_{k \in K} x_{ij}^k), \qquad i, j \in C \cup I, \tag{4.8}$$

$$s_i \geq s_j + c_{ij} - (T + c_{ij})(1 - \sum_{k \in K} x_{ij}^k), \quad i, j \in C \cup I, \tag{4.9}$$

$$s_i \leq s_j + c_{ij} + (T + c_{ij})(1 - \sum_{k \in K} x_{ij}^k), \quad i, j \in C \cup I, \tag{4.10}$$

$$\ell_j \geq c_{dj} \sum_{k \in K} x_{dj}^k, \quad d \in D, j \in C \cup I, \tag{4.11}$$

$$s_i \geq c_{id} \sum_{k \in K} x_{id}^k, \quad d \in D, i \in C \cup I, \tag{4.12}$$

$$f_{dk} \geq o_{dk} - \sum_{k' \in K, r \in I} e_{kk'}^r, \quad d \in D, k \in K, \tag{4.13}$$

$$o_{dk} + \sum_{\substack{d' \in D \\ d' \neq d}} f_{d'k'} + \sum_{r \in I} e_{kk'}^r \leq 2, \quad d \in D, k, k' \in K, \tag{4.14}$$

$$\sum_{k' \in K, r \in I} e_{k,k'}^r \leq 1, \quad k \in K, \tag{4.15}$$

$$o_{dk} = \sum_{i \in C \cup I} x_{di}^k, \quad d \in D, k \in K, \tag{4.16}$$

$$f_{dk} = \sum_{i \in C \cup I} x_{id}^k, \quad d \in D, k \in K, \tag{4.17}$$

$$e_{kk'}^r \leq \sum_{i \in C \cup D} x_{ir}^k, \quad k, k' \in K, r \in I, \tag{4.18}$$

$$e_{kk'}^r \leq \sum_{i \in C \cup D} x_{ir}^{k'}, \quad k, k' \in K, r \in I, \tag{4.19}$$

$$x_{ij}^k \in \{0, 1\}, \quad (i, j) \in A, k \in K, \tag{4.20}$$

$$o_{dk}, f_{dk} \in \{0, 1\}, \quad k \in K, d \in D, \tag{4.21}$$

$$e_{kk'}^r \in \{0, 1\}, \quad k, k' \in K, r \in I, \tag{4.22}$$

$$\ell_r, s_r \in \mathbb{R}^+, \quad r \in C \cup I. \tag{4.23}$$

With objective function (4.1) we attempt to minimize the total cost, which is the fixed cost for using each vehicle plus the sum of the routing costs. Constraints (4.2) are the flow conservation constraints and (4.3) set $e_{kk'}^r$ variables to their corresponding value. Constraints (4.4) impose that exactly one vehicle stops at each customer and (4.5), state that the available capacity for each vehicle cannot be exceeded. Feasibility of the drivers routes with respect to duration limits is guaranteed by constraints (4.6). Constraints (4.7) - (4.10) force the values of the time variables ℓ_r, s_r and binary variables x_{ij}^k to be consistent. They are also used as subtour elimination constraints (SECs). With (4.11) - (4.12) the time corresponding to starting and ending each route is accounted for. Constraints (4.13) focus on "traditional" routes and force them to end at the same depot where they started. Constraints (4.14) ensure that, if k and k' interchange drivers, they interchange depots too, so that drivers end at their home depot. Constraints (4.15) indicate that routes can be interchanged only by pairs. We ensure that the different families of binary variables take consistent values using (4.16) - (4.19), on the one hand o_{dk},f_{dk} and x_{ij}^k, on the other $e_{kk'}^r$ and x_{ij}^k are related. The remaining constraints set the domains of the variables. This formulation uses $O(|V|^2|K|)$ binary variables and $O(|V|^2 + |V||K|)$ constraints. Note that, using constraints (4.16) and (4.17), variables o_{dk} and f_{dk} can be easily eliminated from this formulation. We have kept them here for ease of reading.

Additionally to the large number of variables it requires, this formulation presents a high symmetry. An integer linear program is considered to be *symmetric if its variables can be permuted without changing the structure of the solution* (Margot, 2010). It is well known that three index flow formulations for VRPs present a lot of symmetries. This is induced by the third index that represents the vehicles, since switching two identical vehicles in a feasible solution yields an equivalent solution. Thus, three-index vehicle-flow formulations are known in the literature for having a limited practical interest. The symmetry (number of equivalent solutions) grows exponentially with the number of requests and vehicles (Toth and Vigo, 2014). This is a drawback that affects the performance of optimization software that principally works with the branch-and-bound method, because symmetry produces many similar nodes in the branch-and-bound tree and the solution time becomes unaffordable. However, these formulations naturally provide tight LP bounds (Contardo et al., 2013). In the particular case of VRPs this symmetry has been dealt with using different alternatives. With the aim of partially avoiding the symmetry effects, we applied the following strategies:

Use constraint branching: Given a solution where a given link is fractionally used, we first branch by forcing/forbidding the use of that link in the solution, and only when a link is fully used, we branch on the variables associated with vehicles, if it is required.

Route sorting: Imposing some ordering among the routes of identical vehicles, may become helpful in eliminating some redundant (equivalent) solutions, guaranteeing at the same time that at least one solution representing each equivalence class is included in the resulting feasible set. In this case, we ordered the set of vehicles $k \in K$ according to their load, with the inequalities

$$\sum_{i \in C} q_i \sum_{(i,j) \in A} x_{ij}^k \geq \sum_{i \in C} q_i \sum_{(i,j) \in A} x_{ij}^{k+1}. \tag{4.24}$$

4.2 2-INDEX FORMULATION

In order to reduce the size of the above formulation and eliminate the symmetries, we propose a 2-index flow formulation for the MDVRPVI based on the formulation propossed by Contardo and Martinelli (2014). We still work with the directed graph and sets as in the three index formulation. Each depot $i \in D$ has a fleet of homogeneous vehicles K with capacity Q.

The binary variables in this 2-index formulation, emerge from the natural aggregation of the binary variables in the three-index formulation x_{ij}^k and $e_{kk'}^i$, respectively. So, for each $a = (i, j)$, we have x_a equal to 1 if the arc (i, j) is traversed exactly once in the solution. Therefore, $x_a = \sum_{k \in K} x_{ij}^k$. For $i \in I$, ν_i is equal to 1 only if at location $i \in I$ there is a route exchange (one allowed). That is, now we use $\nu_i = \sum_{k,k' \in K} e_{kk'}^i$. The non negative continuous variables ℓ_i and s_i are defined as before. The following notation is also used: $\forall S \subseteq V, \delta(S)$, denotes the cut-set of S; that is, the set of arcs with one end-node in S and the other in $V \backslash S$. As we are working with a directed graph, $\delta^-(S)$ and $\delta^+(S)$ indicates the sense in the arcs entering or departing from set S. $\gamma(S)$, denotes the set of arcs with both end-nodes in S. Additionally, $x((S : S'))$ $(S : S')$, denotes the set of edges with one end-node in S and the other in S'. $\forall S \subseteq V$ and $\forall S' \subseteq V \backslash S$ will be simplified to $x(S:S')$. $r(S)$ is a lower bound of the minimum number of vehicles needed to serve S (we will use $r(S) = \lceil D(S)/Q \rceil$, where $D(S) = \sum_{j \in S} d_j, \forall S \subseteq C$, is the total demand in S).

Then, the 2-index formulation is:

$$\text{minimize } g \sum_{d \in D, j \in C \cup I} x_{dj} + \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{4.25}$$

subject to

$$x(\delta^{-}\{j\}) = 1, \qquad j \in C, \tag{4.26}$$

$$x(\delta^{+}\{j\}) = 1, \qquad j \in C, \tag{4.27}$$

$$x(\delta^{-}\{i\}) \leq |K|, \qquad i \in D, \tag{4.28}$$

$$\sum_{j \in C \cup I} x_{ij} = \sum_{j \in C \cup I} x_{ji}, \qquad i \in D, \tag{4.29}$$

$$x(\delta^{+}\{i\}) = 2\nu_i, \qquad i \in I, \tag{4.30}$$

$$x(\delta^{-}\{i\}) = 2\nu_i, \qquad i \in I, \tag{4.31}$$

$$\ell_i + s_i \leq T, \qquad i \in C \cup I, \tag{4.32}$$

$$\ell_j \geq \ell_i + c_{ij} - (T + c_{ij})(1 - x_{ij}), \qquad i, j \in C \cup I, \tag{4.33}$$

$$s_i \geq s_j + c_{ij} - (T + c_{ij})(1 - x_{ij}), \qquad i, j \in C \cup I, \tag{4.34}$$

$$\ell_j \geq c_{dj} x_{dj}, \qquad d \in D, j \in C \cup I, \tag{4.35}$$

$$s_i \geq c_{id} x_{id}, \qquad d \in D, i \in C \cup I, \tag{4.36}$$

$$x_{j'i} + x(i : D \backslash \{j'\}) \leq 1, \qquad j' \in D, i \in C \cup I, \tag{4.37}$$

$$x(\delta^{+}\{S\}) \geq r(S), \qquad S \subseteq C, \tag{4.38}$$

Multi-Cut inequalities

Q-Path Constraints

I-Path Constraints

S-Path Constraints

$$x_{ij} \in \{0, 1\} \qquad i, j \in A, \tag{4.39}$$

$$\nu_i \in \{0, 1\} \qquad i \in I, \tag{4.40}$$

$$\ell_i \in \mathbb{R}^{+}, \qquad i \in C \cup I, \tag{4.41}$$

$$s_i \in \mathbb{R}^{+}, \qquad i \in C \cup I. \tag{4.42}$$

The goal is to minimize the objective function (4.25) which includes the fixed costs for vehicle utilization plus the transportation costs. Degree constraints at customers are (4.26)-(4.27). With constraints (4.28) we indicate the maximum number of vehicles that can be used, at each depot. Constraints (4.29) guarantee the flow conservation at the depots, and constraints (4.30)-(4.31) force the interchange locations to have an appropriate degree. Constraints (4.32) establish T as the maximum available driver time. (4.33)-(4.36) are the constraints that involve the coherence between routing and time variables. With (4.37) we prevent solutions with routes visiting just one customer or interchange point starting and ending at different depots. Constraints (4.38) are the well-know capacity constraints, used for the CVRP. Recall that at least $r(S)$ vehicles are needed to serve all the customers in S. Basically, all the above constraints can be obtained from constraint in the 3-index formulation by aggregation and basic constraints operations. Unfortunately, since the vehicle that uses each used arc is not identified now, additional constraints are needed to forbid wrong solution structures. The rest of families of constraints are presented and explained next in detail. Finally, (4.39)-(4.42) define the domains of the variables.

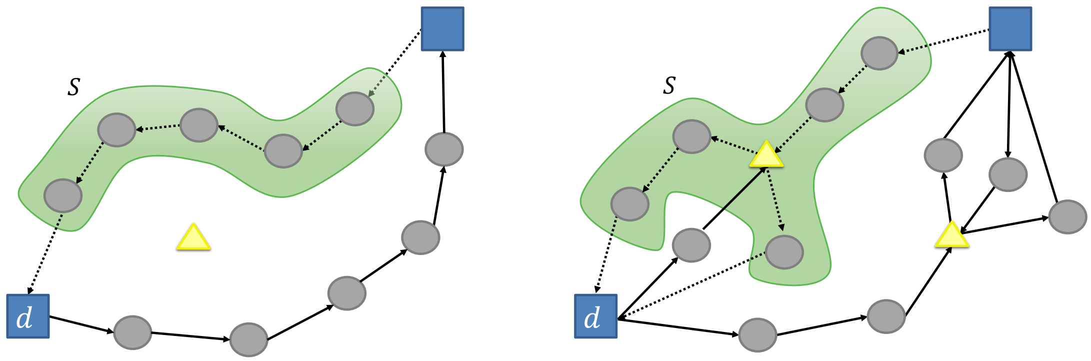
Next we will present and discuss the pending families of constraints.

Multi-Cut inequalities

As in the MDVRP, we need to forbid paths starting and ending in different depots (Figure 4.1(a)). Additionally, in the MDVRPVI, any pair of exchanged routes using the interchange location, must interchange their startin-finishing depots. Figure 4.1(b) shows an infeasible solution with patterns where this fact is not satified, due to an unbalance between the flows entering and leaving from/to depot t from set S. Note that this expression is formally different from the MCC definition given in Bektaş et al. (2017). In that paper, these constraints are expressed as:

$$x(d : S) + x(\bar{S} : S) \geqslant y_d - x(\bar{S} : d)$$

where y_d accounts for the degree at depot d. Since we do not use these variables, we replace $y_d - x(\bar{S} : d)$ with $x(S : d)$, which is equivalent.



(a) Classical path constraint violated (b) Example with exchanged routes unbalanced

Figure 4.1: Examples with violated multicut inequalities.

Let $d \in D$ and sets $S \subseteq C \cup I, \bar{S} = (C \cup I) \setminus S$. The following constraint must be satisfied.

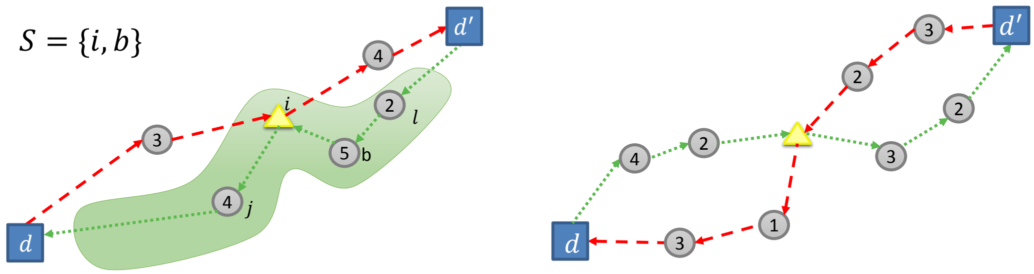
$$x(S : d) \leqslant x(d : S) + x(\bar{S} : S). \tag{4.43}$$

These constraints are valid, since $x(S : d)$ is bounded above by the number of vehicles that visit set S and finish their journey at d. This last value is in turn equal to the number of drivers based at d that visit set S. Each of these drivers must enter set S at least once, either directly from depot d (increasing thus $x(d : S)$) or from some other customer or interchange point (accounted for in $x(\bar{S} : S)$).

Q-Path constraints

Constraints (4.38) ensure that non-interchanging routes satisfy the capacity constraints, and also that they are satisfied by any leg of an interchanging route (depot-interchange point or interchange point-depot). On the other hand, constraints (4.43) forbid non-interchanging routes starting and ending at different depots. Despite these two constraints, it is possible to get solutions with a route exchange violating the available capacity. For example, in Figure 4.2(a) we have an instance where the vehicles capacity is $Q = 10$ and the solution is composed by a pair of exchanged routes.

A first route starting at depot d is satisfying the capacity constraint, but the second one, departing from depot d' delivers a total of 11 units and is, therefore, unfeasible.



(a) Solution a possible Q-feasibility (b) Solution a non possible Q-feasibility

Figure 4.2: Solutions violating the Q-feasibility.

Note however that the set of connections used by unfeasible solutions like those of Figure 4.2 can become feasible by choosing appropriate directions in some cases, but not always. This is the case of Figure 4.2(a), that can be turned feasible as shown in Figure 4.3. As opposite, the second situation shown in Figure 4.2(b) cannot be made feasible by just modifying arc directions.

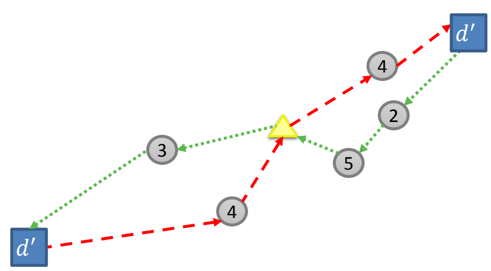


Figure 4.3: Solution with exchanged routes and Q-feasibility.

For this reason, to forbid this type of situations "directed" constraints seem more appropriate. Motivated from the "classical" path constraints (4.50) proposed in Belenguer et al. (2011) Q-Path constraints seek to prohibit the violated path as follows. Taking $j, l \in (C \cup I)$ and sets $S \subseteq (C \cup I) \backslash \{j, l\}$ with $r(S \cup \{j, l\}) \geqslant 2$, $D' \subset D$ and $\gamma_0(S) \subset \gamma(S)$ such that $|\gamma_0(S) \cap \delta^+(i)| = 1$ and $|\gamma_0(S) \cap \delta^-(i)| = 1$ $\forall i \in S$ the following constraint must hold. Then, the following constraint holds:

$$x(\gamma_0(S)) + x(\{j\} : S) + x(S : \{l\}) + x(D' : \{j\}) + x(\{l\} : D \backslash D') \leq |S| + 2. \quad (4.44)$$

The path shown in the first example of Figure 4.2(a) would be forbidden by the constraint from set (4.44) with $D' = d$, and S, j, l taken as shown in the figure.

Note that, given the degree constraints at the customers and he conditions set on γ_0, the left hand side of constraints (4.44) can only reach $|S| + 3$ if the used arcs form a path from a depot in D' to a depot in $D \setminus D'$ going through all customers in $S \cup \{j, l\}$, which is not feasible when $r(S \cup \{j, l\}) \geq 2$.

I-Path constraints

For instances with $|I| > 1$ it is necessary to ensure that x variables will not define any path connecting two interchange locations (Figure 4.4).

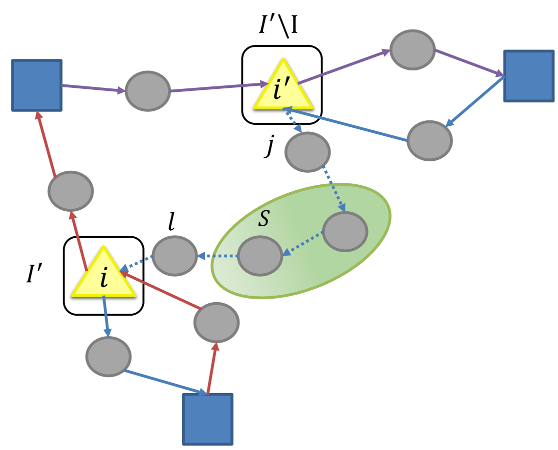


Figure 4.4: Solution with violated I-Path constraint.

To this end, it is possible to use the path elimination constraints. So, in the same manner, taking $j, l \in C, S \subseteq C\backslash\{l, j\}$, but using a set of interchange locations I instead of a set of depots D and $I' \subset I$, we have the I-Path constraints (4.45).

$$x(\gamma(S \cup \{l, j\})) + x(\{j\} : I') + x(\{l\} : I\backslash I') \leq |S| + 2. \tag{4.45}$$

Alternatively, we can derive constraints from the Multi-cut inequalities to forbid the above paths as follows: Let $i \in I$ and sets $S \subseteq C, \bar{S} = (C \cup D) \setminus S$. The following constraint must be satisfied.

$$x(S : i) \leqslant x(\bar{S} : S). \tag{4.46}$$

The rationale behind these constraints is similar to 4.43. The value of $x(S : i)$ equals the number of vehicles that go directly from set S to i. Each of those vehicles must have entered to the set of customers either from a customer not in S, or from a depot, but not from another interchange point.

S-Path constraints

Another set of necessary constraints defining the feasible set of the MDVRPVI is formed by the following S-Path constraints. For instances where $|D| > 2$ and $|I| > 1$ it is possible to get patterns as in Figure 4.5 where two interchanging routes do not swap their home depots, so that even if vehicles are interchanged, the corresponding drivers would not finish their journey at their home depot. In the unfeasible solution depicted in Figure 4.5, identifying the pair of routes associated with the interchange point i, we can see that only one of them is performing a correct trip. For example, taking as first route the one that the two routes that meet at this interchange point (depicted in red and green in the figure) visit completely different depots so that if drivers interchange their vehicles at i, the driver that started his journey at d_2 would finish at d_4, and the one who started at d_3 would finish at d_1.

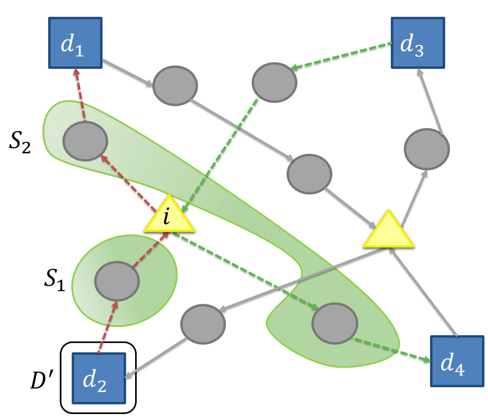


Figure 4.5: Example with violated S-Path constraints.

To prohibit the above situations, we define the S-Path constraints. For $d \in D, i \in I, S_1, S_2 \subset C$ such that $S_1 \cap S_2 = \emptyset$, let $D' = D \setminus \{d\}$, then, the following constraint must be satisfied:

$$x(\gamma(S_1)) + x(\gamma(S_2)) + x(D' : S_1) + x(S_1 : \{i\}) + x(S_2 : \{i\}) + x(\{i\} : S_2) + x(S_2 : D \backslash D') \\ \leq |S_1| + |S_2| + 2\nu_i. \tag{4.47}$$

- Note that, if $\nu_i = 0$, then, by constraints (4.47) $x_{di} = x(S_1 : i) = x(i : S_2) = x(S_2 : i) = 0$. The value of $x(d, S_1) + x(\gamma(S_1))$ is bounded above by the sum of the in-degrees of the nodes in set S_1, which is $|S_1|$. Analogously, considering the out-degree of the nodes in S_2, $+x(\gamma(S_2)) + x(S_2 : D') \leqslant |S_2|$. Adding up both inequalities we obtain the above constraint for this case.

- Let us consider now the case with $\nu_i = 1$. Note that, if subtour elimination constraints are satisfied, the only feasible subcircuits are driver tours associated with pairs of exchanged routes. Therefore they contain both, one depot and one interchange point. Taking this into account, no subcircuit can be closed if neither arcs leaving an interchange point nor entering a depot are considered. Using this fact, we obtain:

  $$x_{di} + x(d, S_1) + x(\gamma(S_1)) + x(S_1 : i) = x\left(\gamma(S_1 \cup \{d, i\}) \setminus \left(\delta_i^- \cup \delta_d^+\right)\right) \leqslant |S_1| + 1. \tag{4.48}$$

  Moreover, the equality can only hold if all nodes in $S_1 \cup \{d, i\}$ are connected, which taking into account the considered arcs, and the degree constraints on the customers, implies that the used arcs contain a path from d to i.

  We consider now the remaining terms in the left hand side of Eq. (4.47). Taking into account the out-degree of nodes in $S_2 \cup \{i\}$, we know that:

  $$\underbrace{x(i : S_2)}_{A} + \underbrace{x(S_2 : i)}_{B} + \underbrace{x(\gamma(S_2))}_{C} + \underbrace{x(S_2 : D')}_{D} \leqslant |S_2| + 2\nu_i \tag{4.49}$$

  Moreover $B + C + D \leqslant |S_2|$ because of the out-degree constraints of nodes in S_2, and $A + B + D \leqslant |S_2|$ because no circuit involving only nodes in $S_2 \cup \{i\}$ can be formed. Therefore, being $\nu_i = 1$, an equality in equation (4.49) would imply that $A \geqslant 2$, $C \geqslant 2$, and that all arcs incident to nodes in S_2 are among those considered in either A, B, C or D, which implies that they contain at least two paths from i to depots in D'.

Summarizing, in case $\nu_i = 1$, equation (4.47) will hold unless both equations (4.48) and (4.49) hold as an equality which is unfeasible. Indeed, if one route departing from d arrives to i, one of the two routes going through i must finish at d; they cannot finish both at D'.

Like in the previous cases, Figure 4.5 provides the sets corresponding to the S-Path constraint that forbids the situation depicted in the figure.

For this 2-index flow formulation there are $|A|+|I|$ binary variables and $2(|I|+|C|)$ continuous variables. Regarding the constraints with polynomial size, we have $2|D| + 2|I| + 2|C|$ related with in-out degree (4.26)-(4.31). To handle the times (4.32)-(4.36) and by the constraint (4.37) there are $5(|I| + |C|) + 3|D|(|I| + |C|)$. A group of constraints (4.38)-(4.43) are constraints with exponential size growing as the number of customers, so, we added it only in the moment that are needed.

4.2.1 Valid inequalities

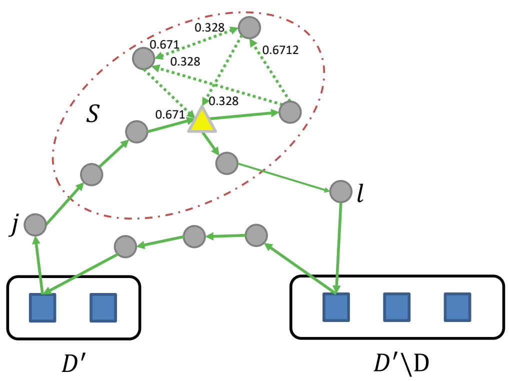
- Path elimination constraints: As we mentioned before, in the MDVRP and in Location Routing Problems too, it is common to use the so-called path elimination constraints in order to avoid paths that connect two distinct depots (See Figure 4.1(a)). For instance, Belenguer et al. (2011) proposed the following family of constraints, defined for each $D' \subset D, j, l \in C$, and $S \subseteq C \setminus \{j, l\}$:

$$x(\gamma(S \cup \{j, l\})) + x(D' : \{j\}) + x(\{l\} : D \backslash D') \leq |S| + 2. \tag{4.50}$$

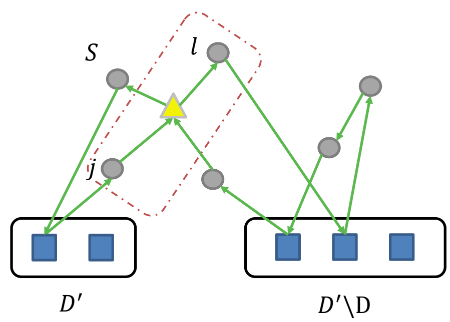
We have adapted these constraints for the MDVRPVI taking into account that, now, paths connecting two depots are allowed, but only in case they are formed by parts of two routes that interchange their vehicles. The obtained family of constraints is defined for $j, l \in C \cup I, S \subseteq C \cup I \backslash \{j, l\}, D' \subset D$ and is expressed as:

$$x(\gamma(S \cup \{j,l\})) + x(D' : \{j\}) + x(\{l\} : D\backslash D') \leq |S\backslash I| + 2 + 2 \sum_{i \in (S \cup \{j,l\}) \cap I} \nu_i. \tag{4.51}$$

With the above constraint if ν_i is in a solution we have two options: a violated path (Figure 4.6(a)) a non-violated path (Figure 4.6(b)), lets analyze this case. These paths with a special combination are allowed if there is a pair of exchanged routes. Note that $x(D' : \{j\}) + x(\{l\} : D\backslash D') \leq 2$, and considering that $\nu_i = 1$ for $i \in (S \cup \{l,j\}) \cap I$, then $x(\gamma(S \cup \{l,j\})) \leq |S\backslash I| + 1 + 2\nu_i$ because of the SECs. Holding the inequality as an equality, customers in $(S \cup \{l,j\} \backslash I)$ are consecutive in a path, it is still valid to forbid subcircuits, since because we take into account a reduced number of arcs incident to ν_i. If $x(D' : \{j\}) = 1$ it means that the path is connected to a depot in D' and with our constraint it is possible that $x(\{l\} : D\backslash D') = 1$.



(a) Violated path constraint using ν_i



(b) Non violated path constraint using ν_i

- Minimum number of needed routes: we have as parameters customers demands and the capacities of the vehicles, so it is possible to compute a lower bound on the number of vehicles that are needed to serve all the customers.

$$\sum_{i \in D, j \in C \cup I} x_{ij} \geq \left\lceil \sum_{j \in C} q_j / Q \right\rceil. \tag{4.52}$$

- Relating the number of routes with the number of interchanges: the total number of used vehicles, which can be computed as the number of used arcs leaving a depot, must be at least twice the number of interchanges.

$$\sum_{d \in D, j \in C \cup I} x_{dj} \geq 2 \sum_{i \in I} \nu_i. \tag{4.53}$$

4.3 Load based formulation

Commonly, VRP formulations have decision variables with a vehicle index to indicate the arcs traversed by each vehicle. This fact involves a large number of decision variables and, as mentioned before, serious symmetry problems. The so-called load-based formulations try to deal with this difficulty. In this kind of formulations, the decision variables identified with the arcs used in the solutions are not explicitly associated with the vehicles that traverse them (Letchford and Salazar-González, 2015).

For the MDVRPVI, the binary variables in this load formulation emerge if the 2-index flow variables are related with the depot where the routes start and end, we will named this as "depot-route". For an exchanged route, it will be associated with the depot it departed from until it arrives to the interchange point, then it will be related to the ending depot from then on. For this formulation, we still work with the set of arcs A initially defined, and we denote by A^* the set of arcs in $(C \times C) \cup (I \times C) \cup (C \times I)$. Formally, binary variables x_{ij}^d with $(i, j) \in A$, $d \in D$ that indicates whether a route associated with depot d traverses the arc (i, j). Similarly, new continuous variables w_{ij}^d will represent the load of the vehicle that traverses arc (i, j) in a route associated with depot d. The rest of variables are maintained as in the 2-index formulation.

The load-based formulation for the MDVRPVI is the following:

$$\text{minimize } g \sum_{\substack{d\in D\\ j\in C\cup I}} x_{dj}^d + \sum_{\substack{d\in D\\ j\in C\cup I}} (c_{dj}x_{dj}^d + c_{jd}x_{jd}^d) + \sum_{\substack{d\in D\\ (i,j)\in A^*}} c_{ij}x_{ij}^d \tag{4.54}$$

subject to

$$\sum_{d\cup D} x_{dj}^d + \sum_{\substack{d\in D\\ i\in C\cup I\setminus\{j\}}} x_{ij}^d = 1, \qquad j \in C \tag{4.55}$$

$$\sum_{d\cup D} x_{jd}^d + \sum_{\substack{d\in D\\ i\in C\cup I\setminus\{j\}}} x_{ji}^d = 1, \qquad j \in C, \tag{4.56}$$

$$\sum_{i\in C\cup I} x_{id}^d \leq |K|, \qquad d \in D, \tag{4.57}$$

$$\sum_{d\in D} x_{di}^d + \sum_{\substack{j\in C\\ d\in D}} x_{ji}^d = 2\nu_i, \qquad i \in I, \tag{4.58}$$

$$\sum_{d\in D} x_{id}^d + \sum_{\substack{j\in C\\ d\in D}} x_{ij}^d = 2\nu_i, \qquad i \in I, \tag{4.59}$$

$$\ell_i + s_i \leq T, \qquad i \in C\cup I, \tag{4.60}$$

$$\ell_j \geq \ell_i + c_{ij} - (T + c_{ij})(1 - \sum_{d\in D} x_{ij}^d), \qquad i,j \in C\cup I, \tag{4.61}$$

$$s_i \geq s_j + c_{ij} - (T + c_{ij})(1 - \sum_{d\in D} x_{ij}^d), \qquad i,j \in C\cup I, \tag{4.62}$$

$$\ell_j \geq \sum_{d\in D} c_{dj}x_{dj}^d, \qquad j \in C\cup I, \tag{4.63}$$

$$s_i \geq \sum_{d\in D} c_{id}x_{id}^d, \qquad i \in C\cup I, \tag{4.64}$$

$$\sum_{j\in C\cup I} x_{dj}^d = \sum_{j\in C\cup I} x_{jd}^d, \qquad d \in D, \tag{4.65}$$

$$x_{dj}^d + \sum_{\substack{i\in C\cup I\\ (i,j)\in A^*}} x_{ij}^d = x_{jd}^d + \sum_{\substack{i\in C\cup I\\ (j,i)\in A^*}} x_{ji}^d, \qquad d\in D, j \in C\cup I, \tag{4.66}$$

$$\sum_{d\in D} x_{ij}^d \leqslant 1, \qquad (i,j) \in A, \tag{4.67}$$

$$w_{dj}^d \leqslant Qx_{ij}^d, \qquad d\in D, j \in C\cup I, \tag{4.68}$$

$$w_{ij}^d \leqslant Qx_{ij}^d, \qquad (i,j)\in A, d \in D, \tag{4.69}$$

$$\sum_{d\in D}\sum_{\substack{k\in C\cup I\cup\{d\}\\ k\neq j}} (w_{jk}^d - w_{kj}^d) = -q_j \sum_{\substack{k\in C\cup I\cup\{d\}\\ k\neq j}} x_{kj}^d, \qquad d\in D, j \in C, \tag{4.70}$$

$$\sum_{j \in D \cup C} w_{ij}^{d} \leqslant \sum_{\substack{d' \in D \\ d' \neq d \\ j \in D \cup C}} w_{ji}^{d'}, \qquad i \in I, d \in D, \tag{4.71}$$

Multi-Cut inequalities*

$$x_{ij}^{d} \in \{0,1\} \qquad (i,j) \in A, d \in D, \tag{4.72}$$

$$\nu_i \in \{0,1\} \qquad i \in I, \tag{4.73}$$

$$w_{ij}^{d} \in \mathbb{R}^{+} \qquad (i,j) \in A, d \in D, \tag{4.74}$$

$$\ell_i \in \mathbb{R}^{+}, \qquad i \in C \cup I, \tag{4.75}$$

$$s_i \in \mathbb{R}^{+}, \qquad i \in C \cup I. \tag{4.76}$$

The objective is the same one that we have been using, minimize the costs with (4.54). Constraints (4.55)-(4.64) are maintained as in the 2-index formulation, by aggregating x variables corresponding to the same arc. The new constraints added to this formulation are group (4.65)-(4.71). Flow conservation constraints on the entering and leaving arcs at each depot-route, and for the customers are defined with constraints (4.65)-(4.66) respectively. Constraints (4.67) establish that each arc is related only to one depot-route. With constraints (4.68)-(4.69) we ensure that the load of vehicles does not exceed their capacity. Indeed, in case $i \in C$, these constraints can be slightly reinforced to

$$w_{ij}^{d} \leqslant (Q - q_i) x_{ij}^{d}.$$

Constraints are (4.70) the load conservation constraints, which are imposed for each customer and depot-route. Constraints (4.71) are necessary when there are exchanged routes. They are used with two objectives. On the one hand, they switch the depots to which each route is linked, according to the definition. (the route is linked to its starting depot until it arrives to the interchange point, and it is linked to the vehicle destination depot after it). On the other hand, they ensure that the loads of the two vehicles arriving to the interchange point are properly kept. The following, are the I-Multi-Cut inequalities adapted the new sets of variables. Let

$i \in I$ and sets $S \subseteq C, \bar{S} = (C \cup D) \setminus S$, we have:

$$\sum_{\substack{j \in S \\ d' \in D}} x_{ji}^{d'} \leq \sum_{\substack{k \in S, j \in \bar{S} \\ d' \in D}} x_{jk}^{d'}. \tag{4.77}$$

These constraints are needed to prohibit a path between two interchange locations. And finally, (4.39)-(4.42) define the domains of the variables. The number of binary variables defining this load based formulation is $(|A| + |D| + |I|)$, and $2(|I| + |C|) + |A| + |D|$ continuous variables. The number of polynomial constraints is $(9|D| + 11(|C| + |I|))$ and constraints (4.77) whose size grows exponentially with the number of customers and interchange points.

Chapter 5

Branch-and-cut algorithms

In this chapter first, we will present some definitions and concepts necessary to explain our solution methods. Then, we explain the branch-and-cut algorithms (Branch and Cut (B&C)) we propose to solve the MDVRPVI, which are based on the 2-index formulation and the load based formulation (respectively) presented in Chapter 4.

- **Branch-and-bound:** This technique consists in dividing the problem into smaller subproblems, which are solved or divide again. Commonly, the Branch and Bound (B&B) is presented such a divided and conquer approach through an enumeration three, where an initial relaxation of the problem is the root node and each subproblem is represented by a node in the tree.

  To solve an (Integer Program (IP)) with B&B algorithm, the first step is to solve the Linear Relaxation (LP) relaxation of the IP. If the solution to the LP and all its corresponding variables have integer values, the obtained solution for the original IP is optimal. In other case, some variable x_i with a fractional value f, is selected to create two new problems. In the first problem, it is necessary to add the constraint $x_i \leq \lceil f \rceil$, and in the second the constraint $x_i \geq \lfloor f \rfloor$. The new problems are recursively solved until all the subproblems are examined. If a subproblem has an integer solution, this value is saved as the incumbent, provided it has a better (lower) than the current best integer solution. The incumbent (value) is used to refuse subproblems whose linear relaxation solution is equal or worse (greater) than the incumbent value, reducing the number of subproblems to explore and at the same time, it reduces the enumeration procedure. In a B&B it is important to get good global upper bounds in short time, because this fact allows to build (in general) a smaller exploration tree compared to the tree produced with a complete enumeration.

- **Cutting planes:** Starting in a domain containing all feasible points, the method iteratively refines that area adding cuts i.e., linear inequalities that are valid (they are satisfied by all integer solutions) but cut-off some parts of the domain of the relaxed problem with attractive objective function values. Most of the cases, Integer Programs contains families of inequalities containing an exponential number of inequalities that cannot be added a priori. So, in such cases it is better to add them iteratively. The first step is to obtain a solution x^* that is the result to solve the LP relaxation of the IP. If this solution is not integer, a separation problem for x^* is solved. For each family of valid inequalities, the separation problem consists in finding a constraint of that family that is not satisfied at x^* or proving that none exists. If at least one violated inequality is found, then this it is added to the current LP, which is then reoptimized. The process keeps until no violated constraints exist from any of the used families. If the used families are enough to define the convex hull of the set of integer solutions, then this process terminates at the optimal solution.

Branch-and-cut (B&C): is a technique that combines of B&B and a generation of cutting planes that is applied to the nodes of the enumeration tree, to tighten the LP relaxations. This approach is often used to solve NP-hard combinatorial optimization problems. The first step of the B&C is to compute the LP relaxation of the IP. If the solution has at least one variable with non-integer value, a cutting plane phase is applied. So, a violated or valid inequalities are added to the linear program to reinforce the relaxation. After that, two new subproblems are defined (according to the branching phase). And so, the algorithm continue through each new subproblem.

Next, the different elements specific to our B&C algorithms are described.

5.1 Separation algorithms

As we already mentioned if in an Integer Problem there are constraints from the exponential-size families, in a B&C algorithm these are adding iteratively. In our case, these families are equations 4.38-4.43 plus valid inequalities. To identify which are these constraints, it is necessary to employ some separation procedures. An exact separation algorithm for a given class of inequalities is a routine which takes as input a LP solution vectors x^* and outputs one or more violated inequalities in that class (if any exist). At each iteration of the algorithm, we find exactly a valid inequality violated with the separation problems for the different possible cases, some kinds of inequalities that are specific to the MDVRPVI. Lets consider $G[\bar{x}, \bar{\nu}]$ that is the weighted graph induced by the arcs a with positive weight, defined with variables $(\bar{x}, \bar{\nu})$. Moreover, $(\hat{x}, \hat{\nu})$ denote if the current LP solution is integer.

- **Separation of the capacity constraints** (4.38)

  As a generalization of the CVRP, any LP solution of our problem can be transformed into an LP solution of a CVRP instance by shrinking all the depots and interchange locations into a single one. In this manner, procedures for separating CVRP constraints can be applied. We used one of the CVRPLIB routines provided by Lysgaard (2004). We have take a heuristic routine of this library. It is well-know that the separation of constraints (4.38) is $NP-hard$, so they resorted to use an heuristic algortihm. The interested reader can find a detailed description of these procedures in Lysgaard et al. (2004).

- **Separation of Multi-cut constraints** (4.43)

  The separation problem for this family of constraints can be stated as follows. Let $\bar{x}$ be a (possibly fractional) solution to a relaxation of the formulations 4.2-4.3. The separation can be solved as a max-flow problem in a network defined as follows:

  $N_x = (V_x, A_x)$ where set V_x contains one node for each $i \in C \cup I$ and two additional nodes, s and t that act as a source and as a terminal, respectively. The set of arcs contains one arc (i, j) for each pair $i, j \in C \cup I$ such that $\bar{x}_{ij} > 0$, and, in fact, $\bar{x}_{ij}$ will be its capacity. Additionally, A_x contains arcs (s, i) and (i, t) for each $i \in C \cup I$. The capacities of these additional arcs are defined as follows. For each node $i \in C \cup I$, let $b_i = \bar{x}_{di} - \bar{x}_{id}$ and take $b_i^+ = \max\{b_i, 0\}$ and $b_i^- = -\min\{b_i, 0\}$. Then, arc (s, i) will have capacity b_i^+, and arc (i, t), b_i^-. Let (U, W) be an $s - t$ cut in this graph, with $s \in U$ and $t \in W$ (Figure 5.1). Then, the capacity of this cut will be:

$$\begin{aligned} &\sum_{\substack{i\in U\setminus\{s\}\\ j\in W\setminus\{t\}}} \bar{x}_{ij} + \sum_{i\in W\setminus\{t\}} b_i^+ + \sum_{i\in U\setminus\{s\}} b_i^- \\ = &\underbrace{\sum_{\substack{i\in U\setminus\{s\}\\ j\in W\setminus\{t\}}} \bar{x}_{ij} + \sum_{i\in W\setminus\{t\}} b_i^+ - \sum_{i\in W\setminus\{t\}} b_i^-}_{v_C} + \sum_{i\in V_x\setminus\{s,t\}} b_i^- \end{aligned} \tag{5.1}$$

Note that the last term of the second expression of the capacity of the cut is a constant and, therefore, the cut of minimum capacity corresponds with the cut that minimizes the value of v_C, which is, in fact, the slack of the Multi-cut constraint associated with $S = W \setminus \{t\}$ in the current solution. Therefore, if the minimum capacity of an $s - t$ cut in this graph is smaller than $\sum_{i\in V_x\setminus\{s,t\}} b_i^-$, W identifies a violated Multi-cut constraint. Otherwise, a minimum capacity larger than this amount proves that no Multi-cut constraint associated with the considered depot is violated.

This separation is needed equivalent to the separation proposed in Bektaş et al. (2017) for the same constraints. As observed in that work, these constraints are valid also if a subset of depots $D' \cup D$ is considered, instead of one in that case.

The extension of our separation to the case of with D' with $D' \neq 1$ requires only redefining the parameters b_i as $b_i = \bar{x}_{di} - \bar{x}_{id}$ without increasing the size of the auxiliary graph.

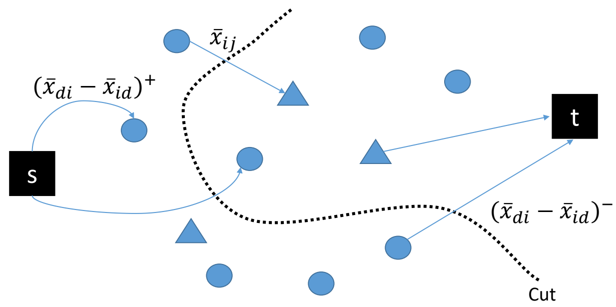


Figure 5.1: Separation graph to our Multi-cut separation.

- **Separation of Q-Path constraints** (4.44)

  As we mentioned before, Q-path constraint are left as lazy constraints and separated by inspection only in integer solutions. After ensuring that the four paths connecting one interchange location with a depot, satisfy the capacity constraint these constraint can be applied. It is possible to get solutions where there is an interchange but the vehicle capacity in a route, to forbid these cases we formulated the Q-Path constraints.

- **Separation of I-Path constraints** (4.45)

  We can see the I-Path constraints as the classical path constraints, where interchange locations play then role of depots instead of a pair of depots. So, we used the same separation procedure as Belenguer et al. (2011). Note that we are working in a directed graph and, thus, the arcs direction is considered. Each connected component of the solution is considered separately, and only those connected components containing more than 2 interchange points are considered; in the other cases no I-Path constraint can be violated. The connected components are identified by using the separation algorithm. Let be $G(\overline{x})$ the support graph, we want to find the most violated set for the current solution. In the LHS, to maximize $x(\{j\} : I') + x(\{l\} : I \backslash I')$ is easy to find the most connected j to I'. Now, following with $x(\gamma(S \cup \{l, j\}))$, adding the degree constraint for the customers in $S \cup \{l, j\}$ we have:

$$2(|S| + 2) = 2\overline{x}(\gamma(S \cup \{l, j\})) + \overline{x}(\delta(S \cup \{l, j\}))$$

then

$$-\overline{x}(\delta(S \cup \{l, j\})) = 2\overline{x}(\gamma(S \cup \{l, j\})) - 2(|S| + 2)$$

Using the max flow min cut theorem, the second part is maximum when the minimum $\overline{x}(\delta(S \cup \{l, j\}))$ is attained.

- **Separation of S-Path constraints** (4.47)

  These group of constraint are left as lazy constraints and if the solution is integer, separated by inspection (as Q-Path constraints). Getting the integer connected component, if there are paths ending in a wrong depot, then identifying all the necessary elements to apply this constraint.

- **Separation of path elimination valid constraints** (4.50)

  As in the I-Path separation, this procedure is as the developed by Belenguer et al. (2011). Similar as before, each connected component of the solution is considered separately, and only connected components containing more than 1 depot are considered; otherwise no path constraint can be violated. Path elimination constraints are determined by a pair of customers or interchange points $\{j, l\}$, $S \subseteq C \cup I \backslash \{j, l\}$, denoted by S^* and a subset of depots $D' \subset D$. All these must be perform the same connected component. We denote with $A(j, l, D') = \overline{x}(\{j\} : D') + \overline{x}(D \backslash D' : \{l\})$. Then, to identify whether there is any violated constraint for this choice of j, l and D' it is necessary to find the set S^* that maximizes is necessary to maximize $\overline{x}(\gamma(S^*))$. Adding the in-degree constraints for nodes in S^* we have :

$$|S^* \backslash I| + 2 + 2 \sum_{i \in (S^* \cap I)} \nu_i = \overline{x}(\gamma(S^*)) + \overline{x}(\delta^+(S^*))$$

  here we see that maximizing $\overline{x}(\gamma(S^*))$ is equivalent to minimize $\overline{x}(\delta^+(S^*))$ so, to find the minimum j-l cut in S^*. Substituting in constraint (4.51),

$$|S^* \backslash I| + 2 \sum_{i \in (S^* \cap I)} \nu_i - \overline{x}(\delta^+(S^*)) + \overline{x}(\{j\} : D') + \overline{x}(D \backslash D' : \{l\}) \leq |S^* \backslash I| + 2 \sum_{i \in (S^* \cap I)} \nu_i.$$

Then if $\overline{x}(\delta^+(S^*))$ if less or equal than $A(i, j, D')$ the path constraint associated with S^* is violated by the current solution $\overline{x}$.

As mentioned above, I-Path constraints can be replaced with I-MultiCut constraints. Some preliminary computational tests recommended to use the separation of the I-path constraints but add the cuts corresponding to the I-Multicut associated with the same set.

Initial relaxation

The initial linear program includes the objective function, the degree constraints for customers, depots and interchange locations (4.26)-(4.31), the group of constraints related with the driver working time (4.32)-(4.36) the valid inequalities (4.52)-(4.53) and the next clearly violated constraints.

- Inter-customer-inter: this is a particular case of the I-Path constraints where $|S| = 1$ and $|I| \leq 2$, we can avoid these paths from the beginning.

$$x_{j'i} + x(j : I\backslash\{j'\}) \geq 1, j' \in I, i \in C. \tag{5.2}$$

Separation Strategy

Deciding which separation procedures will be called and their order is an important issue in a branch-and-cut algorithm. We tested some alternative cutting to determine which strategy is the most effective. For 2-index formulation, we tested three B&C algorithms. In Algorithm 1 we explained step by step the B&C1.

About the load based formulation only one B&C is needed, because we need to separate only the constraint (4.77).

Algorithm 1 Pseudocode of the branch-and-cut algorithm

At the root node, generate and insert all valid inequalities into the program.
Subproblem solution. Solve the LP relaxation of the node.
if There are no more nodes to evaluate - Termination check **then**
    Stop.
else
    Select one node from the branch-and-cut tree.
while $(\bar{x}, \bar{\nu})$ contains at least one violated constraint **do**
    Identify connected components
    Determine whether the component contain a violated constraint with
    if $|D| = 1$ **then**
        use CVRPSep.
    else if $|D| > 1$ **then**
        use PathSep.
    else if $(\hat{x}, \hat{\nu})$ & $|D| = 2$ & $|I| = 1$ **then**
        use Q-PathSep.
    else if $|I| \geq 2$ **then**
        use I-PathSep.
    else if $(\hat{x}, \hat{\nu})$ & $|D| > 2$ & $|I| \geq 1$ **then**
        use S-PathSep.
    else if $|D| \geq 2$ & $|I| \geq 1$ **then**
        use one of the Multi-CutSep.
    Add the result violated constraint.
    Subproblem solution. Solve the LP relaxation of the node.
if $(\hat{x}, \hat{\nu})$ **then**
    Go to the termination check.
else
    Branching: branch on one of the fractional variables.
    Go to the termination check.

Chapter 6

Computational experience

In this section we present the computational experience performed to collect empirical insight on the MDVRPVI solutions, the behaviour of the formulations and the efficiency of the algorithms. All the mentioned algorithms have been implemented in C++ using the callable libraries. As metioned in Chapter 5, we have also used some of the algorithms in the library CVRPSEP Lysgaard (2004). Tests were executed on a computer with an Intel Core i7-4790 processor at 3.60 Ghz with 16 GB.

6.1 Instances tested

Since the MDVRPVI is a new problem, no benchmark instances are available. So, we generated a set of instances by transforming instances of similar problems taken from the literature. On the one hand, to test our approach with Euclidean distances we generated instances from the "SDVRPLIB" instances proposed by Belenguer et al. (2000) (referred to with an "SD" label) and some MDVRP instances from Cordeau et al. (1997) (labelled "MD"). In the case of SDVRP instances, extra depots had to be added. For the MDVRP instances, from a set of four depots, we selected two of them randomly. Customers were chosen randomly and after that, the interchange location was chosen as a central point among the selected customers and depots locations. To concentrate on the routing component of the solutions, we tested some instances with no fixed costs for vehicle utilization ($g = 0$).

Description of network distances

To test with network distances, we took real world instances from Quebec used in Anaya-Arenas et al. (2016). We tested three different depot/interchange configurations, by adding extra locations to the unique depot location from the original instance.

Two groups of network instances were generated. The first one with size $|D| = 2$, $|I| = 1$ and $|C| = 18$ and four different demands. The second one with $|D| = 3$, $|I| = 2$ and $|C| = 40$ and five different demands. The demand vectors were generated randomly and each one was tested with all the locations. These demands were generated with a normal distribution and different means and standard deviations (see Table 6.1).

Instance	μ	σ
1	6	4
2	10	3
3	36	20
4	41	14
5	86	40

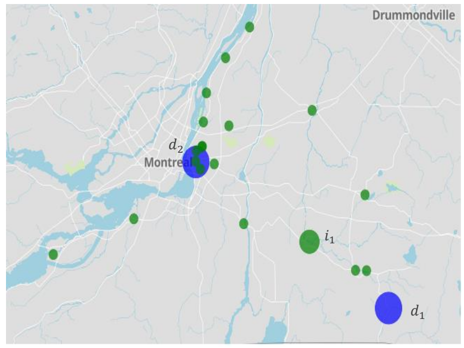
Table 6.1: Parameters by the normal distribution to generate vector demands.

- QC1 instances locations

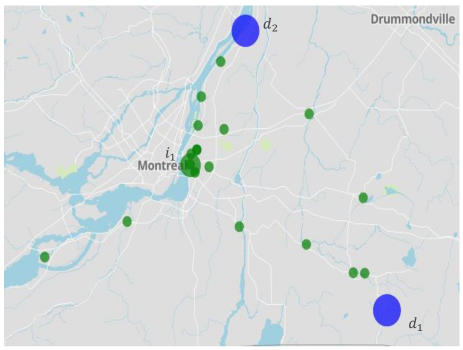
  By the first location (Figure 6.1(a)), additionally to the original depot (d_2 in Anaya-Arenas et al. (2016)) an extra depot was located at the down-right corner of the region, and the interchange point was located in a city between them. In the next two cases, the original depot location was used for the interchange point, and two depots were allocated in extreme points of the region (Figures 6.1(b) and 6.1(c)).

- QC2 instances locations

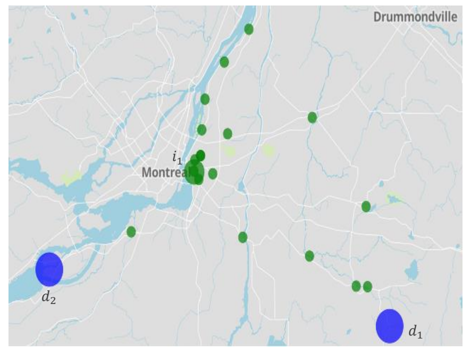
  Departing of the above locations, we added more locations to increase the size of the instances. Location of the depots and interchange location were in the same manner as before, depots at the extreme points of the region and interchange locations at the city center. In Figure 6.2 we can see the three different locations for group two.



(a) Locations 1



(b) Locations 2



(c) Locations 3

Figure 6.1: Locations of the group QC1 network distances tested.

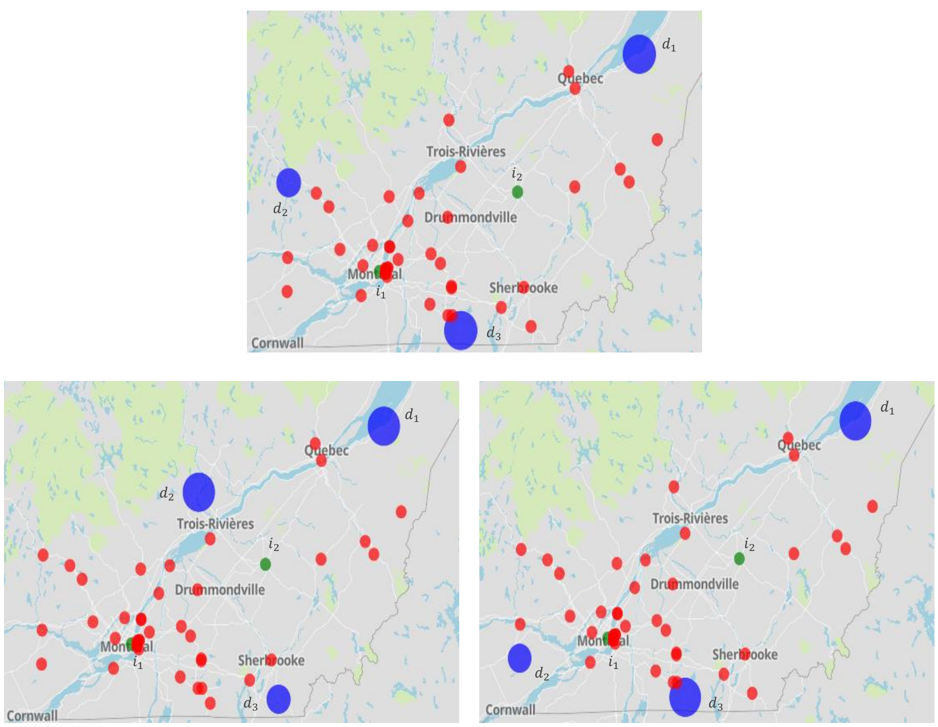


Figure 6.2: Locations of the group QC2 Network distances tested

In Table 6.2 we summarize the dimensions with our tested instances. The column "Range" indicates the initial and final number in the sets of instances. We auxiliary define some sets to referring us a specific instance. For the Euclidean instances *"a"* indicate the number of instance. In the case of "QC1" and "QC2" we use *"b"* to refer the three different locations and *"c"* to the four different demands.

Label	Range	$\|D\|$	$\|I\|$	$\|C\|$
SDa	$\|a\| \in$[1-18]	2	1	[12-20]
MDa	$\|a\| \in$[1-12]	2	1	[12-21]
QC1DLbdc	$\|b\| \in$ [1-3],$\|c\| \in$ [1-4]	2	1	18
QC2DLbdc	$\|b\| \in$ [1-3],$\|c\| \in$ [1-5]	3	2	40

Table 6.2: Dimensions of the instances tested.

6.2 3-index formulation results

The three-index flow-formulation was implemented using Xpress-Mosel Version 7.7. To compare our results with the classical MDVRP, the mathematical formulation proposed by Kulkarni and Bhave (1985) was also implemented with the same software, and the time limit of 7200 sec. (2h.).

In spite of the incorporation of the strategies to reduce symmetry, our formulation is able to solve optimally only small instances ($|D| = 2$, $|I| = 1$ and $|C| \in [12, 20]$). For both, Euclidean and network distances we tested with $|K| = 1$ and $|K| = 2$ available vehicles per depot.

Next tables and figures show the obtained results. In particular, Tables 6.3-6.4 and 6.5 show the results for Euclidean and network instances, respectively. In all cases, results are shown for the MDVRP and for the MDVRPVI, starting with their optimal value, under the heading Z^*. In the case of the Euclidean distances we provide the percent gap of the linear relaxation under *LP gap*, and the time required to solve the MDVRPVI, under *Time (sec.)*. In the case of the network instances, since some of them were interrupted because of the time limit, instead of the execution time we provide the percent gap at termination, both, for the MDVRP and for the MDVRPVI. In both cases, in the rightmost column we provide

		MDVRP		MDVRPVI			
Instance	$\|C\|$	Z^*	LP gap	Z^*	LP gap	Time (sec.)	Saving(%)
SD1	12	215.284	30.7	215.284	29.3	15.83	0.00
SD2	12	256.044	25.0	256.044	23.8	8.28	0.00
SD3	12	237.077	30.8	237.077	31.7	25.21	0.00
SD4	12	246.150	23.6	246.150	22.7	14.11	0.00
SD5	12	215.284	30.7	215.284	29.3	26.49	0.00
SD6	12	215.284	30.7	215.284	29.3	5.09	0.00
SD7	12	235.717	30.4	235.717	31.3	35.08	0.00
SD8	12	263.906	27.2	263.906	26.1	29.36	0.00
SD9	12	237.077	30.8	237.077	31.7	34.57	0.00
MD1	12	370.840	26.3	370.840	25.9	9.02	0.00
MD2	12	532.890	18.4	532.890	17.4	2.88	0.00
MD3	12	455.965	42.3	455.965	41.8	26.08	0.00
MD4	12	489.091	25.5	489.091	24.4	3.00	0.00
MD5	12	361.638	30.9	361.638	30.9	31.56	0.00
MD6	12	481.584	26.4	481.584	22.4	7.94	0.00
MD7	12	577.570	33.4	577.570	32.6	88.30	0.00
MD8	12	476.408	29.8	476.408	27.5	7.61	0.00
MD9	12	522.807	28.2	522.807	27.1	13.53	0.00
MD10	12	430.967	23.8	430.967	23.2	26.44	0.00
SD10	13	166.480	32.0	166.480	31.0	23.12	0.00
SD11	13	242.710	29.0	242.710	28.1	15.04	0.00
SD12	13	271.540	21.9	271.540	21.6	27.96	0.82
SD13	13	243.650	28.8	243.650	27.6	10.11	0.00
MD11	16	246.800	33.4	246.800	31.5	115.69	0.00
MD12	16	285.830	34.5	285.830	32.8	466.85	0.00
SD14	16	292.210	25.8	292.210	24.5	465.86	0.00
SD15	20	263.320	17.9	263.320	18.0	194.72	0.00
SD16	20	279.580	17.4	279.580	15.4	1010.82	1.80
SD17	20	282.640	21.8	282.640	17.7	5494.50	3.70
SD18	20	308.430	25.2	308.430	24.3	7200.60	0.00

Table 6.3: Results for Euclidean instances, $|K| = 2$

the savings attained by allowing route interchanges.

Results using $|K| = 2$ with Euclidean instances are in Table 6.3. We can see that only in instances SD12, SD16 and SD17 there are some savings (0.8%-3.7%) applying MDVRPVI. More significant benefits are shown in Tables 6.4 and 6.5 using $|K| = 1$, where the fleet of vehicles is more limited. We see that the MDVRPVI is able to provide cheaper solutions and take advantage of available resources than the MDVRP although, in some instances our policy cannot improve on the MDVRP solution. Among the 30 Euclidean instances tested, interchanges allowed to reduce the costs in 19 of them. Indeed, we could improve by up to 7.73% with respect to the MDVRP routing costs (instance SD10). The column *LP gap* (for both, $|K| = 2$ and $|K| = 1$) reflects that in general our MDVRPVI formulation provide slightly better bounds than the MDVRP formulation (the gap was smaller in 26 - 29 respectively out of 30 instances and regardless there are savings or not). This is because the constraints that control drivers times and synchronization of routes are useful to strengthen the bound.

		MDVRP		MDVRPVI			
Instance	$\|C\|$	Z^*	LP gap	Z^*	LP gap	Time (sec.)	Saving(%)
SD1	12	231.936	35.7	223.792	32.0	7.88	3.51
SD2	12	292.310	34.3	284.309	31.4	38.47	2.74
SD3	12	244.660	32.9	240.180	32.6	96.35	1.83
SD4	12	246.150	20.7	246.150	22.7	7.25	0.00
SD5	12	231.940	35.7	223.792	32.0	5.53	3.51
SD6	12	241.110	38.1	231.217	34.2	14.58	4.10
SD7	12	249.820	34.3	243.905	33.6	72.78	2.37
SD8	12	298.450	35.7	296.489	34.2	158.21	0.66
SD9	12	278.430	41.1	259.370	37.6	373.63	6.85
MD1	12	449.470	39.2	422.850	35.0	63.11	5.92
MD2	12	581.620	25.3	575.570	23.5	10.01	1.04
MD3	12	463.818	43.3	459.988	42.3	13.00	0.83
MD4	12	489.091	25.5	489.091	24.4	1.32	0.00
MD5	12	361.638	30.9	361.638	30.9	3.06	0.00
MD6	12	561.853	37.0	530.657	29.6	81.83	5.55
MD7	12	577.570	33.4	577.570	32.6	57.88	0.00
MD8	12	476.408	29.8	476.408	27.5	1.37	0.00
MD9	12	531.807	29.4	531.807	28.3	9.44	0.00
MD10	12	438.761	25.2	436.495	24.2	16.69	0.52
SD10	13	206.700	45.2	190.730	39.8	165.67	7.73
SD11	13	242.710	29.0	242.710	28.1	5.07	0.00
SD12	13	277.280	23.5	272.840	22.6	13.06	1.60
SD13	13	250.270	30.7	250.270	29.5	19.10	0.00
MD11	16	255.070	35.5	255.070	33.7	59.63	0.00
MD12	16	287.710	34.9	286.070	32.9	93.84	0.57
SD14	16	309.380	29.9	309.380	28.7	436.12	0.00
SD15	20	269.920	19.9	268.160	19.5	26.94	0.65
SD16	20	279.580	17.4	274.560	15.4	82.83	1.80
SD17	20	282.640	21.8	272.190	17.7	72.38	3.70
SD18	20	308.430	25.2	308.430	24.3	979.67	0.00

Table 6.4: Results for Euclidean instances, $|K| = 1$

In the case of network distances (with $|C| = 18$) delivery routes for both problems are the same for $|K| = 1$ and $|K| = 2$, so, we present the obtained results using $|K| = 1$. In these results, we can see that the profitability of the interchanges depends very much on the locations of depots and interchange points.

As it can be seen in Table 6.5 with the first configuration of depots and interchange points no savings can be attained. As opposite, with location patterns 2 and 3 the savings are much more significant. Even though some of these instances could not be solved optimally (they reached the time limit) we observed savings ranging in 3.7%-5.3%. The fact that already some of these small instances could not be solved in less than two hours gives an idea of the computational complexity in this routing problem. Indeed, looking at this fact, and the CPU times of the Euclidean instances,

Depot location	Demand	MDVRP Z^*	Gap	MDVRPVI Z^*	Gap	Saving(%)
	1	426.36	0	426.36	0.00	0
1	2	426.36	0	426.36	0.00	0
	3	426.36	0	426.36	0.00	0
	4	427.02	0	427.02	0.00	0
	1	487.77	0	464.09	6.60	4.9
2	2	484.99	0	465.06	7.83	4.1
	3	489.84	0	466.22	8.49	4.8
	4	484.99	0	464.86	8.12	4.2
	1	476.94	14.71	459.21	16.96	3.7
3	2	486.13	15.97	465.01	17.63	4.3
	3	486.93	13.26	462.32	16.23	5.1
	4	484.99	12.63	459.45	14.16	5.3

Table 6.5: Results for QC1 instances, $|K| = 1$

one can conclude that the difficulty of solving a particular instance is more related with the distribution of customers and locations than with the instance size, at least for the small cases.

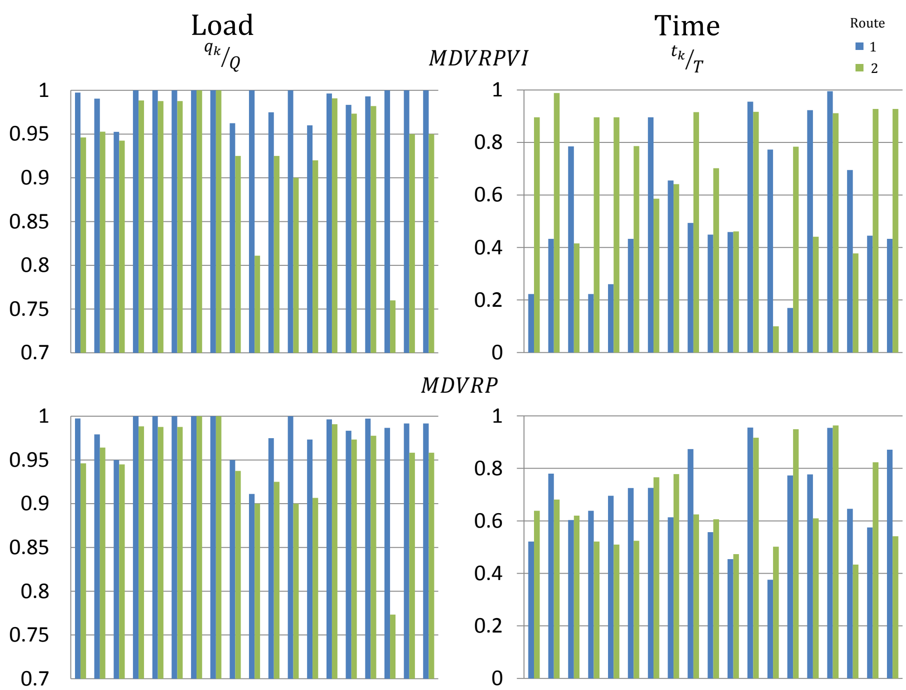


Figure 6.3: Utilization of vehicle capacity and driving time (Euclidean distances).

In order to further analyze the obtained solutions, we decided to look for the resource utilization in the optimal routes for both problems in the instances with $|K| = 1$, where the optimal solution is formed by two routes. So, in all of these instances where savings were obtained we computed a ratio indicating to what extent each resource was used in each route. To compute these ratios, we have computed the total demand served by each route, q_k, and divided it over the available vehicle capacity Q. In the same way, we have computed the driver time in route, t_k, divided by the driver working time T. All those ratios are depicted in Figures 6.3 and 6.4 for Euclidean and network instances, respectively.

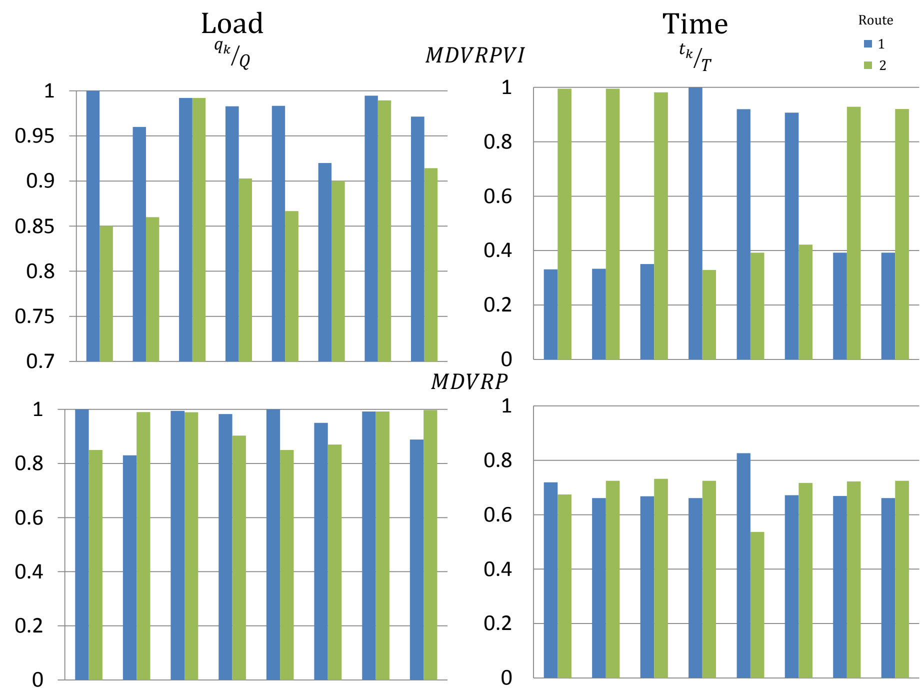


Figure 6.4: Utilization of vehicle capacity and driving time (QC1 instances).

For the Euclidean instances we can observe that, regarding the vehicle load, both solutions are quite balanced and the capacity is almost fully used. Indeed, no route uses less than 75% of the vehicle capacity, and in most cases 90% or more is used. In the case of network distances, we can appreciate the largest ratio between the capacity utilization of the two routes in a solution is 15% (instance DL2D1). Concerning the driver times, in general these constraints are looser and there are more variations in the utilization ratios. Surprisingly, the MDVRPVI instances present more imbalance in one route with respect to the other, because the time is associated with the driver. Therefore, since drivers must be synchronized at the interchange location, if only one of them performs a long route before this location the final routes become quite unbalanced; otherwise, one of them would be infeasible.

6.3 2-index formulation results

The branch-and-cut code was implemented in C++ using Xpress-BCL callable library. All pre-processing, heuristics and separation of valid inequalities implemented by XPRESS were turned off. Except for this, default XPRESS settings were used.

To test this algorithm, we generate more network instances with size $|D| = 3$, $|I| = 2$ and $|C| = 40$, with the same characteristics as the previous ones. The time limit stated is 10800 sec. (3h.)

Since the most demanding instances we tested with the 3-index formulation were those corresponding to the Euclidean instances and two vehicles per depot, in this section we compare these results with those of the 2-index formulation on the same subset of instances.

Table 6.6 displays the results obtained, again with a CPU time limit of 3 hours. Since many instances in this subset could not be solved within this time limit, the best solution found by either algorithm need not be the same. The best solution obtained among the two algorithms is given under heading *BKS*. Then, two columns are provided for each formulation. The first one, with heading *gap* provides the gap reported by Xpress at termination, while the second one, with heading %*dev* provides the percent deviation of the value of the best solution obtained with each algorithm from the best known solution so far.

As it can be seen, the 2-index formulation represents in general terms an improvement on the previous one even if it was not able to solve more instances than the 3-index formulation. On the one hand, the gaps at termination with this formulation are significantly smaller. Indeed, the maximum percent gap observed is now 5.5% is about half the average gap for the previous formulation was 11.2%, with a maximum of 21.1%. On the other hand, the best solution was found with the new formulation in 10 of the 12 instances and the deviations for the other two are very small (0.21% and 0.03%, respectively) while the 3-index formulation provided the best solution only in 6 instances, and for the others it yielded solutions which were up to 5% more expensive than the best known one.

Depot location	Demand	BKS	3-index Gap	3-index %dev	2-index Gap	2-index %dev
	1	426.36	0.0	0.00	0.0	0.00
1	2	426.36	0.0	0.00	0.0	0.00
	3	426.36	0.0	0.00	0.0	0.00
	4	426.36	0.0	0.00	0.0	0.00
	1	463.31	10.4	0.00	3.7	0.21
2	2	463.43	12.9	0.74	3.8	0.00
	3	465.25	16.5	4.12	4.1	0.00
	4	462.68	15.4	5.03	2.5	0.00
	1	459.66	19.3	0.65	5.5	0.00
3	2	459.54	19.4	0.00	5.1	0.03
	3	464.78	19.3	0.19	1.4	0.00
	4	455.96	21.1	1.55	5.0	0.00

Table 6.6: 2-index vs 3-index formulation. QC1 intances, $|K| = 2$.

As already mentioned in Chapter 5, the 2-index formulation includes many exponential-size families that need to be separated within the branch and cut. In the design of the final algorithm we explored different cutting strategies. The best performing ones are summarized in Table 6.7. Recall that, in this formulation,

	At integer solutions						**At fractional solutions**			
Order	1	2	3	4	5	6	1	2	3	4
B&C1	cap.	Path	Q-Path	I-Path	S-Path	MC*	cap.	Path	MC	I-Path
B&C2	cap.	MC	Path	Q-path	I-Path	S-Path	cap.	MC	Path	I-Path
B&C3	cap.	MC	Q-path	I-path	S-Path		cap.	MC	I-Path	

(*MC stands for Multi-Cut)

Table 6.7: Separation strategies tested

Path constraints act as valid inequalities and, therefore, they need not be separated, actually. For this reason, the last algorithm variant does not use them.

Since some of the constraints that are initially relaxed can only apply in instances with several interchange locations and depots(I-Path, S-Path), we have used the set of larger instances (with 3 depots, 2 interchange locations and 40 customers) with no fixed costs per vehicle utilization to test the four algorithm variants. For this experiment we set a CPU time limit of 3 hours. Table 6.8 provides its main results. Again, the value of the best solution found using the three algorithm is given under *BKS* after that, deviations of the solutions obtained with each algorithm with respect to this value are given under heading *%dev wrt best*, and percent gaps at termination, under *gap*. In these last columns, the smallest value is highlighted in boldface. From the results above, we observe that the first variant of the branch and

Depot location	Demand	BKS	%dev wrt best B&C1	B&C2	B&C3	gap B&C1	B&C2	B&C3
	1	2138.42	0.0	4.8	6.2	**17.8**	21.7	22.6
1	2	2325.35	4.4	0.0	4.9	27.2	**22.8**	27.3
	3	2254.95	4.0	13.2	0.0	28.6	34.4	**25.6**
	4	2219.17	8.1	1.7	0.0	29.7	25.4	**23.9**
	1	2161.65	9.1	9.9	0.0	21.1	21.7	**14.0**
2	2	2293.54	0.0	32.2	1.7	**17.9**	38.0	19.2
	3	2576.52	0.0	29.6	3.1	**32.9**	48.2	34.9
	4	2162.44	3.8	35.4	0.0	22.8	40.8	**19.9**
	1	1995.13	0.0	2.2	4.0	**9.5**	11.4	12.9
3	2	2321.63	0.0	10.8	1.6	**21.3**	29.2	22.8
	3	2122.56	0.0	16.0	2.6	**21.2**	32.3	23.3
	4	2231.00	0.0	0.8	7.3	**24.8**	25.5	29.7

Table 6.8: Separation strategies. QC2 instances with $|C| = 40, g = 0$.

cut provides the best results. The average percent gap at termination is 22.9% for this variant, while it is 29.3% for B&C2, and 23.0% for B&C3. Also, the number of instances where it provides the best solution is 7, much larger than what we observe for the other variants (1 and 4, respectively). Variant B&C2 is the worst of the three in all respects. As for the quality of the obtained solutions when they are not the best ones, again B&C2 yields the poorest results, giving in some occasions solutions that are over 30% more expensive than the other variants. With this respect, variants 1 and 3 provide quite similar results. Their average deviations from the best solution are 2.5% and 2.6%, respectively, and the maximum ones, 9.1% and 7.3%.

We conclude that B&C2 has worse behavior than the other two variants and B&C3, although giving slightly worse global values behaves similarly to B&C1. Therefore, it seems to us that the difference between these two variants does not pay the additional complexity to include one family of valid inequalities on top of the already many families of constraints of exponential size. For this reason, we keep B&C3 as our choice for the 2-index formulation.

Figure 6.5-6.7 display the number of cuts of each type that were found during the process. As it could be expected, capacity cuts represent the largest proportion of cuts (over 80% in almost all instances for all three strategies). It can be also seen how,

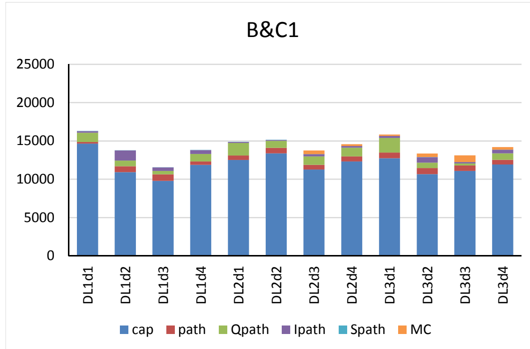


Figure 6.5: Cuts generated with B&C1

in the two strategies were multi-cut constraints are separated early in the algorithm, many more of them are obtained. However, this alone does not explain the behavior of the algorithms, since the amount of such constraints found with B&C2 and B&C3 is similar, and their results in terms of solution quality and percent gaps are quite different. With the three figures we also observe that the number of separated cuts

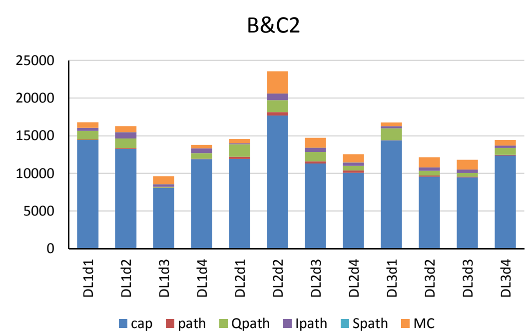


Figure 6.6: Cuts generated with B&C2

is most often below 15000, but there is one instance for which algorithm B&C2 required over 20000 (23560, in fact). Note that, this particular instance is one of the two where this algorithm gave the worst percent gap. As can be seen in the figure, violated S-path constraints are seldom found. Actually, on average, they represent a 0.02% of the violated constraints identified. Given this small value, the differences in the number of such cuts violated are not noticeable in the figures. We have observed that B&C1 tends to find more such constraints. Only in two of the instances there was no constraint of this family violated throughout the application of B&C1, while the number of such instances for the other two variants was 6 and 7, respectively.

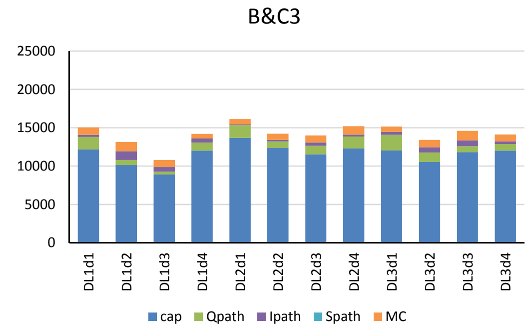


Figure 6.7: Cuts generated with B&C3

6.4 Load-based formulation results

Finally, we present in this section the comparison of the two last formulations. Since one single family of constraints are relaxed in the branch and cut for the formulation with load variables, one single variant of the branch and cut is considered here and it is compared with the results obtained with B&C3 from the previous section.

We first report the results obtained on the smaller Network instances with $|K| = 2$. Recall from Table 6.6 that none of the instances with depot locations 2 and 3 could be solved to optimality within 3 hours with B&C3. We report in Table 6.9 the results obtained on those instances with the load-based formulation. Again, we compare the solution obtained with the solution of the classical MDVRP.

As can be seen, there are two instances for which the MDVRP could not be solved to optimality. In the case of the MDVRPVI, with this formulation we could solve all the instances but three, for which the remaining gaps range between 0.24% and 2.94%. This gives us a first hint on the superiority of the performance of this formulation.

Since now we know the optimal solution in most instances, we can evaluate again the cost saving attained by using vehicle interchanges. Observe that one of the obtained savings is -0.16%. This awkward value corresponds to a MDVRPVI instance that could not be solved to optimality. Therefore, it is clear that the best solution found at termination was not the optimal solution, which is, at most, as expensive as the MDVRP solution. With the third configuration of depots and interchange points, savings between 2% and 4% have been obtained.

Depot location	Demand	MDVRP Z^*	MDVRP Gap(%)	MDVRP Time(sec.)	MDVRPVI **Load** Z^*	MDVRPVI **Load** Gap(%)	MDVRPVI **Load** Time(sec.)	Saving(%)
1	1	426.36	0	6344.48	426.36	0.0	1074.95	0
	2	426.36	0	2384.03	426.36	2.0	10799.50	0
	3	426.36	4.2	10799.60	426.36	0.0	8960.96	0
	4	426.36	0	2946.63	427.04	2.9	10799.00	-0.16
2	1	462.68	0	145.42	462.68	0.0	266.72	0
	2	462.68	0	106.18	462.68	0.0	327.12	0
	3	469.52	0	1467.63	465.25	0.0	900.34	0.91
	4	462.68	0	109.92	462.68	0.0	205.40	0
3	1	468.33	0	869.95	459.19	0.0	172.29	1.95
	2	473.169	0	2184.68	459.52	0.2	10799.90	2.88
	3	475.05	0	4764.82	460.04	0.0	367.41	3.16
	4	472.59	2.3	10800.00	455.25	0.0	8742.64	3.67

Table 6.9: Results of QC1 with load-based formulation, $|K| = 2$.

Now, together with the instances used in the previous sections, we will also consider the set of instances with the same locations, but with nonzero fixed cost per vehicle utilization (recall that, in this case, this cost has been set as $g = 100$). Tables 6.10 and 6.11 provide the obtained results, again, with a 3h time limit.

Depot location	Demand	BKS	%dev wrt best 2-index	%dev wrt best load	gap 2-index	gap load
	1	2257.24	0.6	0	29.2	28.4
	2	2437.53	0.1	0	37.5	37.5
1	3	2191.81	2.9	0	34.4	30.6
	4	2089.23	6.2	0	31.5	23.8
	5	2209.94	11.2	0	46.8	32.0
	1	2161.65	0.0	18.3	16.2	37.5
	2	2331.39	0.0	23.2	23.7	52.5
2	3	2224.31	19.4	0	49.0	24.7
	4	2147.38	0.7	0	24.1	23.2
	5	2162.98	10.2	0	37.9	25.1
	1	2074.49	0	22.2	14.8	40.4
	2	2359.78	0	2.5	29.6	32.8
3	3	2177.58	0	2.5	30.4	33.7
	4	2047.51	16.9	0	42.3	21.8
	5	2168.89	15.4	0	49.4	29.5

Table 6.10: Comparison of 2-index and load-based formulations. QC2 instances with $|C| = 40, g = 0$.

As it happened before, no instance in the two groups could be solved to optimality with either formulation and, therefore, we report under heading *BKS* the value of the best solution found with both formulations, and under *%dev wrt best* the percent deviation of each solution value with respect to the best one. Finally, we give the final gaps reported by Xpress in each case.

As for the complexity of the two different sets of instances, we must say that the presence of a fixed cost for vehicle utilization does not have a great impact on the results obtained. Both, the gaps at termination and the deviations from the best solution are similar in tables 6.10 and 6.11.

Depot location	Demand	BKS	%dev wrt best 2-index	load	gap 2-index	load
	1	2498.72	0	20.5	10.4	33.0
	2	2710.35	7.7	0	28.8	19.5
1	3	2624.11	48.1	0	87.1	26.3
	4	2731.1	8.9	0	43.0	31.3
	5	2783.06	13.3	0	51.7	33.9
	1	2890.71	0	9.8	22.5	34.6
	2	3048.61	23.9	0	58.8	28.1
2	3	2953.43	5.5	0	42.9	35.5
	4	2699.15	10.3	0	39.7	26.6
	5	2776.88	6.4	0	39.1	30.6
	1	2459.92	0	25.1	6.8	33.5
	2	3018.23	1.2	0	31.1	29.6
3	3	2688.45	17.2	0	52.2	29.9
	4	2550.38	6.9	0	31.1	22.6
	5	2650.4	6.0	0	35.9	28.1

Table 6.11: Comparison of 2-index and load-based formulations. QC2 instances with $|C| = 40, g = 100$.

The gap at termination with the load-based formulation was smaller in 9 of the 15 instances with $g = 0$, and in 12 of the 15 instances with $g = 100$. Although the relative differences are not always very large, it must be pointed out that in the computation of those gaps, both, the upper and the lower bounds are different. By concentrating on the columns that provide the deviations from the best solution, we can see that the load based formulation provided the best solution in 10+12=22 of the 30 instances, and in the others it provided a solution very close to the best one. As opposite, the 2-index formulation gave solutions that are quite worse than the best one. The average deviation in the $g = 0$ instances was 5.6%, with values raising up to 19.4%. In the case of the instances with $g = 100$, these values are 10.54% and 48.1%, respectively.

These results tell us that the load based formulation provides weaker bounds, and it takes much longer to increase these bounds but, at the same time, it allows to identify good quality solutions much faster. This can be further appreciated in the Figure 6.8 where the deviation of the lower bound obtained at the root node from the best known solution is depicted.

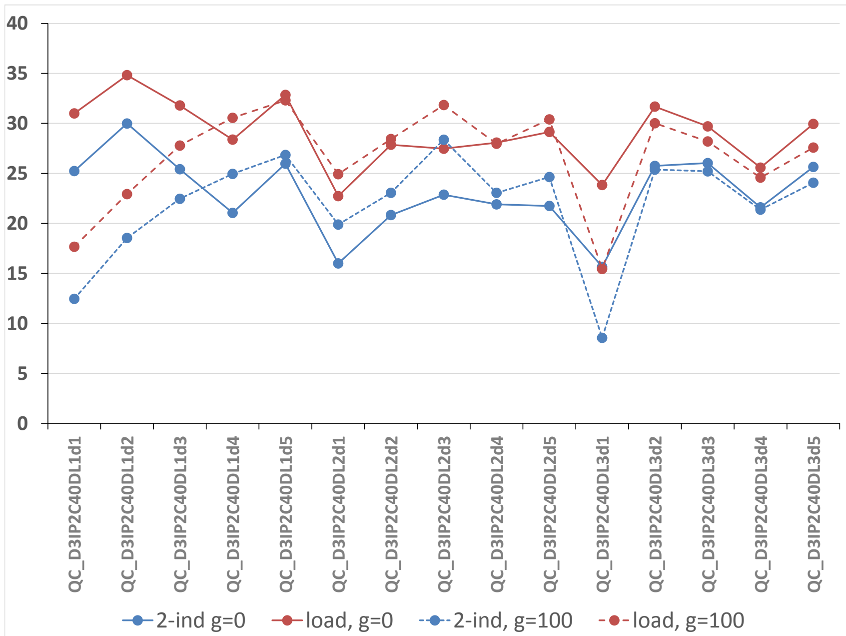


Figure 6.8: Bounds at the root node. Deviations with respect to best known solution.

Chapter 7

Conclusions

In this thesis we introduce a new variant of the MDVRP: the MDVRPVI. In this problem drivers can exchange their vehicles in predefined meeting points, so that vehicles capacities and working times can be better combined to reduce transportation costs. To assess the benefits of this new policy, we show that using driver interchanges it is possible to reduce the routing costs in 50% with respect to the MDVRP. Additionally, we evaluate the empirical savings on a set of test instances. To this end, we present a first mathematical formulation for this problem, using three-indexed flow variables. Due to the complexity of the MDVRPVI and the known limitations of the 3-index flow-formulations, only very small instances can be solved. Despite of this, the first results show relevant savings with our approach. The difficulty to solve instances of reasonable sizes with the initial 3-index formulation has motivated us to develop two additional alternative formulations. The first one uses 2-indexed variables. In this formulation, binary variables are associated with the use of the arcs of the underlying graph, but have no information on the vehicle using them. The correct definition of the set of feasible solutions using this variables involves five families of constraints of exponential size. Additionally to these families, an extra family of valid inequalities have been adapted from other Multi-Depot vehicle routing problems from the literature.

To solve this second formulation we have devised a specific branch-and-cut algorithm. In its design, some separation procedures have been taken from the literature, some others have been designed or adapted to our particular problem, and 2 families are separated by inspection only in integer solutions.

The second alternative formulation uses variables associated with the load of the vehicle that traverses a given arc. Additionally, binary variables has again a third index but, now, it is not associated with the vehicle but with a depot, which is the home depot of the driver traversing each arc. The nature of this third index represents a trade-off between the size and symmetries of the 3-index formulation, and the need for additional exponential-size families of constraints of the 2-index variables formulation. In this case, only one exponential size family of constraints is required to avoid paths connecting two interchange points.

According to the results, the best performing formulation in terms of its ability to provide good quality solutions relatively fast is the load-based formulation. Unfortunately, this nice behavior comes together with a rather loose lower bounds, that make it extremely time consuming to close the gaps and prove optimality. Therefore, a first direction of future research is to devise valid inequalities and other enhancements of this formulation that allow to improve this bound.

Finally, a natural extension of this thesis is to develop a heuristic method for the MDVRPVI that allows to find good quality solutions efficiently; not only in order to be able to provide solutions for instances of reasonable size, but also to speed up any exact algorithm.

List of Abbreviations

B&B Branch-and-bound.

B&C Branch-and-cut.

IP Integer Program.

LP Linear Relaxation.

MDVRP Multi-Depot Vehicle Routing Problem.

MDVRPVI Multi-Depot Vehicle Routing Problem with Vehicle Interchanges.

RVRP Rich Vehicle Routing Problem.

VRP Vehicle Routing Problem.

Bibliography

Afshar-Nadjafi, B. and Afshar-Nadjafi, A. (2017). A constructive heuristic for time-dependent multi-depot vehicle routing problem with time-windows and heterogeneous fleet. *Journal of king saud university-Engineering sciences*, 29(1):29–34.

Anaya-Arenas, A. M., Chabot, T., Renaud, J., and Ruiz, A. (2016). Biomedical sample transportation in the province of Quebec: a case study. *International Journal of Production Research*, 54(2):602–615.

Andersson, P.-Å., Hermansson, Å., Tengvald, E., and Scalia-Tomba, G.-P. (1979). Analysis and simulation of an urban bus route. *Transportation Research Part A: General*, 13(6):439–466.

Angelelli, E. and Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, 137(2):233 – 247.

Atefi, R., Salari, M., Coelho, L. C., and Renaud, J. (2018). The open vehicle routing problem with decoupling points. *European Journal of Operational Research*, 265(1):316–327.

Baldacci, R. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380.

Baldacci, R., Ngueveu, S. U., and Calvo, R. W. (2016). The vehicle routing problem with transhipment facilities. *Transportation Science*, 51(2):592–606.

Bartolini, E. and Schneider, M. (2018). A two-commodity flow formulation for the capacitated truck-and-trailer routing problem. *Discrete Applied Mathematics*, https://doi.org/10.1016/j.dam.2018.07.033.

Bektaş, T. and Elmastaş, S. (2007). Solving school bus routing problems through integer programming. *Journal of the Operational Research Society*, 58(12):1599–1604.

Bektaş, T., Gouveia, L., and Santos, D. (2017). New path elimination constraints for multi-depot routing problems. *Networks*, 70(3):246–261.

Belenguer, J.-M., Benavent, E., Prins, C., Prodhon, C., and Calvo, R. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6):931 – 941.

Belenguer, J.-M., Martinez, M. C., and Mota, E. (2000). A lower bound for the split delivery vehicle routing problem. *Operations Research*, 48(5):801–810.

Braekers, K., Caris, A., and Janssens, G. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166 – 186.

Cáceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., and Juan, A. A. (2015). Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, 47(2):32.

Ceselli, A., Righini, G., and Salani, M. (2009). A column generation algorithm for a rich vehicle-routing problem. *Transportation Science*, 43(1):56–69.

Contardo, C., Cordeau, J.-F., and Gendron, B. (2013). A computational comparison of flow formulations for the capacitated location-routing problem. *Discrete Optimization*, 10(4):263 – 295.

Contardo, C. and Martinelli, R. (2014). A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129 – 146.

Cordeau, J.-F., Gendreau, M., and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30(2):105–119.

Crevier, B., Cordeau, J.-F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756 – 773.

Díaz, G., Scavarda, L., Rocha, R., and Hamacher, S. (2014). Decision support system for PETROBRAS ship scheduling. *Interfaces*, 44(6):555–566.

Domínguez-Martín, B., Rodríguez-Martín, I., and Salazar-González, J.-J. (2017). An exact algorithm for a vehicle-and-driver scheduling problem. *Computers & Operations Research*, 81:247–256.

Domínguez-Martín, B., Rodríguez-Martín, I., and Salazar-González, J.-J. (2018). The driver and vehicle routing problem. *Computers & Operations Research*, 92:56–64.

Dondo, R. and Cerdá, J. (2015). The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. *Computers & Chemical Engineering*, 76:42 – 62.

Drexl, M. (2013). Applications of the vehicle routing problem with trailers and transshipments. *European Journal of Operational Research*, 227(2):275 – 283.

Drexl, M., Rieck, J., Sigl, T., and Press, B. (2013). Simultaneous vehicle and crew routing and scheduling for partial-and full-load long-distance road transport. *Business Research*, 6(2):242–264.

Escobar, J. W., Linfati, R., Toth, P., and Baldoquin, M. G. (2014). A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. *Journal of Heuristics*, 20(5):483–509.

Fernández, E., Roca-Riu, M., and Speranza, M. G. (2018). The shared customer collaboration vehicle routing problem. *European Journal of Operational Research*, 265(3):1078–1093.

Gopalakrishnan, B. and Johnson, E. L. (2005). Airline crew scheduling: state-of-the-art. *Annals of Operations Research*, 140(1):305–337.

Hall, R. and Partyka, J. (2016). Vehicle routing software survey: Higher expectations drive transformation. *OR/MS Today*, 43(1):40–44.

Huber, S. and Geiger, M. J. (2014). Swap body vehicle routing problem: A heuristic solution approach. In *International Conference on Computational Logistics*, pages 16–30. Springer.

Kulkarni, R. and Bhave, P. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1):58 – 67.

Lahyani, R., Coelho, L. C., and Renaud, J. (2018). Alternative formulations and improved bounds for the multi-depot fleet size and mix vehicle routing problem. *OR Spectrum*, 40(1):125–157.

Lahyani, R., Khemakhem, M., and Semet, F. (2015). Rich vehicle routing problems: From a taxonomy to a definition. *European Journal of Operational Research*, 241(1):1 – 14.

Lam, E., Van Hentenryck, P., and Kilby, P. (2015). Joint vehicle and crew routing and scheduling. In *International Conference on Principles and Practice of Constraint Programming*, pages 654–670. Springer.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416.

Laporte, G., Nobert, Y., and Arpin, D. (1984). *Optimal solutions to capacitated multidepot vehicle routing problems*. École des hautes études commerciales.

Laporte, G., Nobert, Y., and Taillefer, S. (1988). Solving a family of multi-depot vehicle routing and location-routing problems. *Transportation Science*, 22(3):161–172.

Letchford, A. N. and Salazar-González, J.-J. (2015). Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730–738.

Li, J., Li, Y., and Pardalos, P. (2014). Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, pages 1–18.

Li, J., Wang, R., Li, T., Lu, Z., and Pardalos, P. M. (2018). Benefit analysis of shared depot resources for multi-depot vehicle routing problem with fuel consumption. *Transportation Research Part D: Transport and Environment*, 59:417–432.

Lysgaard, J. (2004). A CVRP package. `http://www.hha.dk/~lys/CVRPSEP.htm`.

Lysgaard, J., Letchford, A. N., and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2):423–445.

Margot, F. (2010). Symmetry in integer linear programming. In Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., Rinaldi, G., and Wolsey, L. A., editors, *50 Years of Integer Programming 1958-2008*, pages 647–686. Springer Berlin Heidelberg.

Meisel, F. and Kopfer, H. (2014). Synchronized routing of active and passive means of transport. *OR spectrum*, 36(2):297–322.

Mesquita, M. and Paias, A. (2008). Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem. *Computers & Operations Research*, 35(5):1562–1575.

Min, H., Current, J., and Schilling, D. (1992). The multiple depot vehicle routing problem with backhauling. *Journal of Business Logistics*, 13(1):259.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., and Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79:115–129.

MWPVL-International (2015). Amazon global fullfullment center network. http://www.mwpvl.com/html/amazon_com.html, Last visited: November,2018.

Rais, A., Alvelos, F., and Carvalho, M. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, 235(3):530 – 539.

Sun, X., Garg, M., Balaporia, Z., Bailey, K., and Gifford, T. (2014). Optimizing transportation by inventory routing and workload balancing: Optimizing daily dray operations across an intermodal freight network. *Interfaces*, 44(6):579–590.

Tlili, T., Harzi, M., and Krichen, S. (2017). Swarm-based approach for solving the ambulance routing problem. *Procedia Computer Science*, 112:350–357.

Toth, P. and Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. (2014). Implicit depot assignments and rotations in vehicle routing heuristics. *European Journal of Operational Research*, 237(1):15–28.

Weise, T., Podlich, A., and Gorldt, C. (2009). Solving real-world vehicle routing problems with evolutionary algorithms. In *Natural intelligence for scheduling, planning and packing problems*, pages 29–53. Springer.

Zhou, L., Baldacci, R., Vigo, D., and Wang, X. (2018). A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. *European Journal of Operational Research*, 265(2):765–778.