A matheuristic for the consistent vehicle routing problem with service level agreements: a case study in the pharmaceutical distribution sector

Pedro Filipe Dias Belchior Campelo

Masters' Dissertation

Supervisor: Prof. Pedro Sanches Amorim

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Industrial e Gestão

2015-07-01

Abstract

Distributors are being pressured to increase the frequency of deliveries and to provide better service as customers look to them as business partners that allow them to decrease stock levels and to have a quicker and more reliable response to stock-outs. Moreover, in highly competitive markets, overall customer satisfaction and the provided service level are the main drivers behind customer retention. For these reasons, companies in this environment are looking towards customer-focused solutions, such as consistent routing strategies. However, in complex and unpredictable environments with stochastic customers and demand, robust routes with driver consistency are hard to plan and to implement.

A mathematical model was then developed to tackle the Consistent Vehicle Routing Problem while considering customers with multiple daily deliveries and different service level configurations, such as time windows, daily period specifications and order deadlines right before vehicle departure. In addition, an instance size reduction algorithm and a Fix-and-optimize based matheuristic are presented. This solution approach allows the use of the model to solve instances of larger size, thus enabling its application in real-life scenarios.

A case study in a Portuguese pharmaceutical distribution company is then analyzed, in which a database and key performance indicators dashboard is developed. The proposed model and matheuristic are then used to plan their routes in five different warehouses, with instances having up to 905 total number of orders. Finally, a simulation model is created to test the performance of the new robust routes plans under real historical data, while also enabling the evaluation of what-if scenarios.

By applying the matheuristic to historical data, the model was able to reduce the overall objective value by 7.2% and the total traveled distance by 5.5%. Simulations were then run to compare the performance of the robust routes plan proposed by our model to the routes currently in place which were mainly developed using a commercial route planner. In the simulated horizon, using the routes suggested by our model while optimizing the route sequence, the total traveled distance is reduced by 17.4% and the total route durations decrease by 8.4%. These improvements have an estimated impact of 12.7% in the monthly distribution cost of the company considered to be influenceable by the project.

Resumo

As empresas de distribuição têm sido pressionadas para aumentarem a frequência das entregas e para forncerem um serviço de maior qualidade visto que os seus clientes os vêem como parceiros de negócio que permitem uma redução dos níveis de existências e uma resposta mais rápida quando há falta de produtos. Em mercados altamente competitivos, a satisfação dos clientes e o nível de serviço são os fatores mais importantes na retenção de clientes. Por estas razões, as empresas que operam nestes setores consideram cada vez mais soluções focadas no cliente como sistemas de rotas fixas. No entanto, em ambientes mais complexos e imprevisíveis, com clientes e procura estocásticos, as rotas robustas com consistência de condutor são difíceis de planear e de implementar.

Um modelo matemático foi então desenvolvido para atacar o Consistent Vehicle Routing Problem, a considerar clientes com diversas entregas diárias e com configurações de nível de serviço diferentes tais como janelas de entrega, especificações de período diário de entrega e limites de pedido próximos da hora de saída do veículo. Um algoritmo de redução de tamanho de instancias por agrupamento de entregas e uma matheurística baseada no Fix-and-Optimize são apresentadas. Este método de solução permite o uso do modelo com instâncias significativamente maiores, o que torna possível a sua aplicação em cenários reais.

Um caso de estudo numa empresa portuguesa de distribuição de medicamentos é depois analisado, no qual uma base de dados e um mapa de indicadores de desempenho são desenvolvidos. O modelo e a matheurística propostos são então usados para planear as rotas de cinco centros de distribuição diferentes, tendo as instâncais um total de até 905 entregas. Por fim, o desempenho dos novos planos de rotas robustas é testado num modelo de simulação com dados reais, sendo que o mesmo modelo também permite a análise de condições de negócio alternativas.

Ao aplicar a matheurística a dados históricos, o modelo foi capaz de, em média, reduzir a função objetivo global em 7.2% e a distância percorrida em 5.5%. Foram então realizadas simulações para comparar o desempenho dos planos de rotas robustas propostos pelo modelo com os planos usados atualmente pela empresa que foram desenvolvidos através de um sotware de roteamento. No horizonte temporal simulado, ao usar as rotas sugeridas pelo modelo e a otimizar o sequência em que são feitas, foi possível diminuir, em média, a distância percorrida em 17.4% e a duração total das rotas por 8.4%. Estas melhorias têm um impacto nos custos mensais de distribuição da empresa considerados influenciáveis pelo projeto estimado em 12.7%.

Acknowledgements

First and foremost I would like to thank my supervisor, Prof. Pedro Sanches Amorim, for giving me the opportunity to be a part of such an exciting adventure throughout this semester. His uninterrupted encouragement and overwhelming dedication not only put me through the harder times but also inspired me to constantly strive for greatness. I am utterly grateful for his friendship and for becoming a role model for my professional development and future endeavors.

I would like to thank Fábio Moreira for his unceasing support which was definitely the cornerstone to all the work developed during this dissertation. His presence in the team provided never-ending motivation and invaluably enjoyable recreational moments.

I am also extremely gratified by having spent these last month as part of the most supportive, brilliant and inspiring team in CEGI. I could not have wished for a better work family and I hope that my contribution to the team was as big as the influence they had in my personal development.

To all of my family I give my most profound gratefulness, especially my parents and my sister, which made me who I am today and helped me overtake all obstacles I have faced. I can only hope to make you as proud as I am proud of having you guiding me through life.

I would also like to thank all of my friends for all the moments we spent together, and especially Vitor for never letting me forget what is truly important.

Inês, I am deeply overjoyed for sharing these last years with you. You make me a much better person every single day and I sincerely hope to be able to retribute all the care and support you have given me. Thank you for making the present worthwhile and the future worth looking forward to.

"Somewhere, something incredible is waiting to be known"

Carl Sagan

Contents

Acronyms

ALNS Adaptive Large Neighbourhood Search. 6, 8, 48

API Appplication Programming Interface. 35

conVRP Consistent Vehicle Routing Problem. 2, 5–7, 47

conVRP-SLA Consistent Vehicle Routing Problem with Service Level Agreements. 11, 14, 17, 23, 30, 33, 34, 40, 43, 44, 47, 48

FO Fix and Optimize. 17, 18, 22, 41, 42, 47

GA Genetic Algorithm. 7

KPIs Key Performance Indicators. 33, 39, 40, 43, 47

LNS Large Neighbourhood Search. 6

MIP Mixed-Integer Program. 6, 7, 11, 22, 26

PVRP Periodic Vehicle Routing Problem. 6–8

SLA Service Level Agreements. 11

TS Tabu Search. 6–8

TSPTW Traveling Salesman Problem with Time Windows. 8, 39, 44

VNS Variable Neighbourhood Search. 7

VRP Vehicle Routing Problem. 5–9

VRPTW Vehicle Routing Problem with Time Windows. 8, 9

List of Figures

List of Tables

Chapter 1

Introduction

Customer satisfaction is becoming one of the main drivers in competitive markets, and therefore new factors need to be considered during the planning and execution of delivery operations. In some markets, customers try to decrease their overall stock levels and simultaneously improve response time in case of a stock-out. This is pressuring distributors, especially when operating with several competitors, to increase the frequency of delivery and to be more concerned with providing a good service rather than straight up reducing its operating costs. In this setting, the operational efficiency of the resources involved in this whole process is crucial as the border between profit and losses is thin. From an Operations Research point of view, these delivery operations often boil down to vehicle routing problems that arise whenever goods need to be distributed throughout a series of delivery locations by more than one vehicle. While this broad description fits a wide variety of applications, it is common for several specific business characteristics to harden the decision making process on this problem, which in turn leads to the development of increasingly specific models that relate more closely to practical issues faced by companies.

In this research we are concerned with a scenario in which customers have high-standard service level agreements, which may vary significantly in their nature (Table 1.1). These agreements are provided by companies that ensure customized delivery solutions. In the distribution business it is common for this type of agreements to guarantee that between ordering and delivering only a given, small amount of hours may pass, and that multiple deliveries throughout the day may be fulfilled. While the existence of time windows is often the case in vehicle routing problems, having multiple customer-specific order placement deadlines increases planning complexity as it is not possible to aggregate orders from the same customer in the same route if they were placed in different periods of the day. Moreover, some of the orders may be placed just in time for the vehicle loading operations and subsequent departure, which further complicates routing flexibility. An additional challenge occurs when customers with multiple daily order deadlines are to be served alongside customers with less broad agreements that either do not enforce any specific period of delivery or have just a less specific one. In these cases it is possible that, throughout the day, multiple routes are able to deliver these requests. Choosing which route should serve these customers is not trivial since it has implications not only on traveled distance but also on vehicle capacities and may even influence delays throughout the whole day. This is especially common in distribution companies that deliver goods from different companies and different businesses which have specific and non-conformant service level agreements with their own customers. In this case, the distribution company needs to adapt their operations and to develop routes that fulfill all the different specifications required.

Table 1.1: Examples of different service level agreements

Customer	Daily order frequency	Time window	Dispatching period	Ordering deadline
A	Several daily deliveries	Expected arrival time	Afternoon shift	14:15
B	One visit	Between 10:00 and 17:00	Any	Week before

Uncertainty, which renders the planning and execution of these systems harder, may come from several sources such as demand, service times and customers' locations. Moreover, the impact of uncertainty is felt both at the tactical and operational level. On the tactical level, the decisions about the operational model to serve clients and fleet sizing have to be taken simultaneously and must incorporate these uncertainty sources explicitly. Regarding the operational model there are mainly two alternatives of planning the routes to deliver the products: flexible routes that are routed dynamically based on the current orders that have to be delivered or robust routes that have defined the set of clients that are served per route a priori. The first option could theoretically lead to a better dimensioning of the fleet and more optimized routes, but it ignores the fact that, upstream, the orders that were known a couple of hours before have to be prepared and consolidated next to a given delivery vehicle. Furthermore, the important goal of personalization of services, making the driver the contact person whenever the customer needs service is forgotten. Using robust routes one may also increase the driver familiarity with their own routes and territories, which improves driver efficiency. Actually, the increased focus on the customers makes driver recognition an ever more important factor in route planning. Drivers become the face of the company and a very effective medium of customer management as they become acquainted with the downstream decision makers. In addition, robust routes provide a much more stable and easy way to manage delivery schedules and enable the possibility of delivering to customers at approximately the same time every day. The robustness of customer-route assignment may also be motivated by operational level constraints when the customers' order placement deadline is very close to vehicle departure. In these situations, it may be necessary to start picking operations and to load the vehicles before the full list of orders for the day is known. By having robust routes, the assignment of each order to a vehicle is both easier to manage and less error-prone. The tactical challenge of these logistics systems is then how to design the best set of robust routes that will result in the lowest operating costs. This problem is known in the literature as the Consistent Vehicle Routing Problem (conVRP). In recent years, the conVRP has been gathering the attention of the academic community as innovative and better solutions are being requested by companies. In this type of problem, customers are assigned to specific routes before any orders have been placed during a tactical planning phase. On the operational level, every order placed by a customer will be served by the same vehicle at approximately the same time over multiple days. This impedes the flexibility of dynamically molding the routes to the requests to be delivered in order to strictly minimize costs, which is the way vehicle routing problems are usually addressed.

From a practical point of view, the problem being tackled applies to specific sectors that benefit from planning their routes according to the previously defined motivations. Companies that put their focus on customer service level, maintain a close relationship with the customer and operate on a tight delivery lead time have the most benefits to withdraw from using a delivery system with consistent routes. Cases of such companies are fairly common in small package distribution industries facing large competition and highly demanding customers, with pharmaceutical and automobile spare parts distribution being the most well-known examples. Other activities in which consistent routing with a service level focus is relevant include home care services, better established home deliveries, which may be a potential development area for e-commerce operations, and also the transportation of children, elderly or handicapped people. These companies seek for a customer focused environment that leads to more complex and strict service level agreements as a way to increase customer loyalty. However, with stricter service level targets, the number of failures and complaints tends to increase, which often leads to frequent changes to the tactical plan in order to quickly solve these problems. Making solid consistent route plans that will perform well in practice while still being efficient is therefore a very hard task that many companies struggle with. Tools that assist decision-making in these conditions and take the business characteristics in consideration are therefore very useful and desired by practitioners.

Several approaches have been proposed to address variants of the problem tackled in this thesis. Our contribution is two-fold. Firstly, to the best of our knowledge, there is no work that tackles the service level agreement often encountered in these logistics systems, in which not only the delivery window, but also the time between ordering and receiving in the same day is defined. This constraint limits drastically the number of clients to be paired and the flexibility in the departure times of the vehicles. Secondly, although several formulations have been put forward to solve consistent vehicle routing problems, they were never used to address real-world instances. In this work we propose a solution method leveraged by a fix-and-optimize approach that fully utilizes the developed mathematical model and aims to be well suited for application in real-life business situations. In order to test its validity and potential, the solution method is used with historical data to plan the robust routes of a Portuguese pharmaceutical distribution company with over 3000 daily deliveries in an environment with both deterministic and stochastic customers and stochastic demand. The proposed plans are then simulated and the performance of the new plan is compared to the one currently in practice at the company.

The remainder of this thesis is organized as follows. The theoretical background is introduced in Chapter 2 with a literature review of the vehicle routing problem and some of its specific formulations with more relevancy to our model. Chapter 3 presents the formulation of the problem and the respective mathematical model, while Chapter 4 describes the proposed solution approach thoroughly and the motivations for its development. The case study is then introduced in Chapter 5 with a brief description of both the company's operations and all the tools developed and used during the analysis. In Chapter 6, the results of the proposed model on the real case study are shown and analyzed. Finally, conclusions and improvement opportunities are discussed in Chapter 7.

Chapter 2

Literature Review

Since its introduction by Dantzig and Ramser (1959), the research community has been extensively studying different Vehicle Routing Problem (VRP) variants and applications. As expected, the complexity of the addressed challenges also suffered a massive increase as the value of the optimization techniques captured the interest of most competitive logistics operators. The computational power that is available nowadays enables the scientific community to further develop new mathematical models as well as the necessary methods to solve them. This remarkable and logical evolution is described by Laporte (2009).

2.1 Consistent Vehicle Routing Problem

This research mainly focuses on a less studied VRP extension which is the conVRP. This optimization problem demands the definition of vehicle routes for several periods, maintaining a certain level of consistency on pre-selected metrics. For instance, when distribution companies make an agreement for the deliveries to be made always by same driver, they are adding consistency constraints in order to take into account customer satisfaction. Therefore, the objective is to achieve minimum cost routing plans satisfying the classical routing constraints as well as consistency requirements taking into account customer satisfaction. Generally, this type of customer-oriented routing considers two types of consistency for customer satisfaction: driver consistency, and time consistency (Kovacs et al., 2014a). Driver consistency is measured by the number of different drivers that visit a customer whereas time consistency is related to the maximum difference between the earliest visit and the latest arrival at each customer. The conVRP arises in many industries where customer satisfaction is considered as a distinctive factor of competitiveness. Particularly in industries transporting small packages, providing a standard service with a single driver and approximately at the same time of the day enables the customers to prepare themselves for a delivery, strengthening supplier/customer relationships (Kovacs et al., 2014b). Since the conVRP considers several periods, it can be seen as a tactical extension of the classical VRP with customer-focused routes.

Despite the advantages of adopting consistent routes, few papers have addressed the conVRP and most approaches resort to approximation methods. Groer et al. (2009) formulate the conVRP as a Mixed-Integer Program (MIP) and improve the algorithm used by Li et al. (2005) to solve very large VRPs. A real-world data set is used to generate instances with up to 700 customers which are solved by the algorithm. The obtained consistent routes are less than 10% longer on average, compared to inconsistent routes. Recently, Ridder (2014) shows that some optimal solutions provided by Groer et al. (2009) are not feasible because service times were not considered. The author develops an algorithm that improves solutions provided by the latter paper. Tarantilis et al. (2012) propose a Tabu Search (TS) algorithm to iteratively generate template routes and to improve the daily routes that are derived from the template routes. These routes are used as the basis to construct the vehicle routes and service schedules for both frequent and non-frequent customers over multiple days. The best reported cumulative and mean results over all conVRP-benchmark instances is improved. Kovacs et al. (2014b) construct template routes by means of an Adaptive Large Neighbourhood Search (ALNS), which uses several operators in order to destroy and repair a given solution. It is shown that solving daily VRPs may lead to inconsistent routes whereas consistent long-term solutions can be generated by using historic template routes. Kovacs et al. (2014a) state that assigning one driver to each customer and bound the variation in the arrival times over a given planning horizon may be too restrictive in some applications. They propose the generalized conVRP in which a customer is visited by a limited number of drivers and the variation in the arrival times is penalized in the objective function. A Large Neighbourhood Search (LNS) metaheuristic generates solutions without using template routes. The computational results on different variants of the conVRP prove the efficiency of the algorithm, as it outperforms all published algorithms. Sungur et al. (2010) consider a real-world courier delivery problem where customers appear probabilistically. Although the authors do not call it a conVRP, their assumptions are completely in line with this type of problem. The proposed approach generates master plans and daily schedules with the objective of maximizing both the coverage of customers and the similarity between the routes performed in each day. In order to deal with uncertain service times, it is assumed that the master plans serves frequent customers with the worst-case service times found in historical data. Once again, a mathematical formulation is proposed but the real-world problem is tackled by means of a two-phase heuristic based on insertion and TS.

The papers proposing a mathematical formulation to deal with consistency features are still scarce in the literature. These formulations are only able to solve small instances containing less than 12 customers. To the best of our knowledge, no single approach is able to efficiently make use of a mathematical formulation to solve conVRP instances with realistic size.

2.2 Periodic Vehicle Routing Problem

The Periodic Vehicle Routing Problem (PVRP) demands the definition of vehicle routes for several periods without considering consistency constraints. Contrary to the conVRP, the PVRP has been studied extensively for more than forty years and its applications cover numerous contexts. Recently, many applications are considering the classic PVRP as a basis where additional constraints or alternative objectives are added (Campbell and Wilson, 2014). The conVRP is an application of the PVRP considering consistency constraints and the two problems are closely related. Therefore, it is worth to review the literature concerning this topic. The periodicity of VRPs may be considered in three different ways. Either it can be achieved with a predefined set of allowable alternatives (Christofides and Beasley, 1984) or it can be obtained by specifying the space between deliveries to each customer (Cordeau et al., 2001). One can also enforce minimum and maximum required spacing between deliveries (Gaudioso and Paletta, 1992). In this work, we are particularly interested in the first case, as it is more related to our challenge.

The PVRP has been tackled by numerous solution approaches. Cordeau et al. (1997) propose a TS metaheuristic for a PVRP with multiple depots. The algorithm constructs an initial solution that is later improved by the search algorithm. The computational results show that the algorithm outperforms other PVRP heuristics available at the time, solving instances with up to 360 customers and 9 periods. Hemmelmayr et al. (2009) present a Variable Neighbourhood Search (VNS) for a PVRP with time-windows. The initial solution is constructed by solving a VRP for each day using the Clarke and Wright savings algorithm (Clarke and Wright, 1964). Afterwards, the algorithm searches for better solutions by applying the classical shaking, local search and move or not phases that are presented by Hansen and Mladenović (2001). They find new best solutions for sets of instances containing up to 400 customers and up to 10 periods. Vidal et al. (2012) tackle a PVRP by means of an hybrid Genetic Algorithm (GA). The metaheuristic combines the good exploratory features of population-based evolutionary algorithms, the improvement capabilities of neighbourhood-based metaheuristics, as well as advanced procedures to manage the population-diversity. The computational experiments for the PVRP, considering a set of instances with up to 400 customers, showed that the method is able to identify either the best-known solutions or new best solutions for all benchmark instances. Therefore, it outperforms current state-of-the-art metaheuristics.

Note that the aforementioned approaches are approximated methods which do not guarantee optimal solutions. Indeed, few exact approaches are available for the PVRP. Baldacci et al. (2011) present a mathematical formulation that is used in a framework consisting in three phases. The formulation is strengthened with valid inequalities and used to compute a near-optimal dual solution of the LP-relaxation. Then, a reduced integer problem containing all optimal solutions is generated from the dual solution. Finally, the resulting problem is solved with an MIP solver. The effectiveness of the proposed method is shown on benchmark instances and on new sets of test instances as well. Since it presentes an exact method, this paper is also the first evaluation of the best-known solutions for the PVRP instances that had been used in the previous 30 years.

It is quite clear that exact methods are not able to solve large instances. Researchers try to hybridize exact and approximated phases so as to enhance the algorithms for the PVRP. Pirkwieser and Raidl (2009) propose a column generation where a mathematical formulation selects a set of optimal routes (columns) to solve a PVRP. The algorithm starts by considering a small set of routes and additional routes that can potentially lead to better solutions are iteratively generated by solving the a pricing subproblem. Recently, Cacchiani et al. (2014) propose a hybrid optimization algorithm which also relies on a column generation approach. Although there are not many approaches based on this idea, this strategy of restricting the scope to exact parts of algorithms should be better explored. Indeed, there is evidence that approaches such as the Fix-and-Optimize (Helber and Sahling, 2010) can be quite efficient in different optimization problems. The idea is to iteratively find improvements on a given solution by fixing parts and optimizing the smaller subproblems. This enables the possibility to solve larger instances, as we can fix as much variables as the solver needs. It seems reasonable to use a similar approach in large and realistic PVRPs with consistency constraints.

2.3 Vehicle Routing Problem with Time-Windows

Besides the consistency constraints regarding the number of drivers servicing each customer, the problem we aim to tackle in this work also considers time-windows for each request. This feature is usually necessary in rich routing problems as customers need to be prepared to receive shipments. Since this type of constraint is responsible for a huge increase in complexity, most successful approaches are obtained with non-exact methods.

Solution approaches based in the TS metaheuristic have been proven to be quite efficient. Cordeau et al. (2001) propose a unified TS heuristic for the Vehicle Routing Problem with Time Windows (VRPTW) where an initial solution is constructed, without guaranteeing feasibility. Afterwards, the TS algorithm generates a certain number of solutions and chooses the best feasible one. This solution is then post-optimized by applying a specialized heuristic for the Traveling Salesman Problem with Time Windows (TSPTW) to each individual route. The computational experiments show that the proposed algorithm may not be the best available for the VRPTW. Nevertheless, this weakness is compensated by the flexibility, the speed of execution, and memory usage of the approach. The program can run on any computer with minimal resources, solving instances with up to 100 customers. Taillard et al. (1997) propose a TS heuristic for a VRP with soft time-windows. The vehicles are allowed to arrive late to customer locations, though a penalty is incurred in the objective function. Problems with hard time-windows can also be tackled by setting penalties to large values. The algorithm uses a stochastic insertion heuristic to construct different solutions. Then, the TS heuristic is applied to each solution and the resulting routes are stored in an adaptive memory, which will be later used to construct other solutions. Subset routes are combined in order to obtain novel solutions until the stopping criteria are met. In the end, a post optimization procedure is applied to each individual route. This methodology has produced best-known solutions for VRPTW benchmark instances. Pisinger and Ropke (2007) present a general heuristic for several VRPs, including the VRPTW. An ALNS metaheuristic is applied, improving 183 best-known solution out of 486 benchmark tests. In this metaheuristic, a number of simple algorithms compete to modify the current solution. In each iteration, an algorithm destroys and another algorithm repairs the solution. The choice of the algorithms to be used in each iteration is made by an adaptive layer which is biased according to the past performance of each algorithm. The authors describe this methodology as a sequence of fix-and-optimize operations. The fix operation selects a subset of variables that are fixed at their current value whereas the optimize operation seeks to find a near-optimal solution by changing non-fixed variables. This is one of the most successful methods among the VRP community, and therefore we consider that exploring different techniques to fix and optimize parts of routing problems is a promising research direction.

Purely exact methods are not so common in the literature as they are not able to solve instances with a large number of customers. The survey presented by Kallehauge (2008) reviews four different formulations for the VRPTW and describes two main lines of development concerning exact algorithms. One focuses on general decomposition approaches and in the solution to dual problems associated with the VRP. The other is concerned with the analysis of the polyhedral structure of the problem. Despite the advances in the exact methods, tackling instances with realistic size is still an unwise decision. Baldacci et al. (2012) review recent exact methods for the VRPTW and report a comparison between different approaches. It is interesting to observe that state-of-the-art approaches are only able to solve instances with less than 100 customers.

Chapter 3

Mathematical formulation

Consistent vehicle routing is still most likely a sub-optimal plan on the daily basis as it trades off the necessary flexibility to respond to the specific daily demand of each customer in order to allow the use of consistent routes. It is therefore of utmost importance to avoid excessive compromises of solution quality in the solution method to ensure suitable and efficient results, which ultimately led to the development of an exact model of the proposed problem. In this section, the Consistent Vehicle Routing Problem with Service Level Agreements (conVRP-SLA) will be formally stated along with the associated MIP model.

3.1 Problem statement

Let $\mathscr{D}$ be the set of days with service to be fulfilled and $\mathscr{P}$ a set of distinct periods of shipping within the depot. Let $\mathscr{C}$ be the set of customers having to be served throughout the days. Each customer has a set of ordering windows that, according to their Service Level Agreements (SLA), may or may not have an associated delivery time window $[a_i, b_i]$, a previously assigned shipping period p_i and/or an order availability time av_i (Figure 3.1). The time window refers to the time at which the order should be delivered at the customer, the period specifies whether this order should be delivered by a route in a specific shipping period and the order availability time is the earliest time at which that specific order could be ready for shipping if it was to be placed at exactly the ordering deadline. All ordering periods which do not have a specific order deadline during the daily horizon are assumed to be ready at opening hours.

Let $\mathscr{N}$ be the set of n nodes, each representing a pair costumer-ordering window with c_i being the customer of node i. In this way, every different window of delivery of each customer is to be treated as a different node (Figure 3.2). Two additional fictitious nodes 0 and $n+1$ will be added to the set representing the depot as a departure node and a return node, respectively, with $\mathscr{N}_C$ being the subset of all non-depot nodes. Let $\mathscr{N}_{TW}$ be the subset of nodes with a time window, $\mathscr{N}_P$ the subset of nodes with a specific shipping period and $\mathscr{N}_{OD}$ the subset of nodes with an order

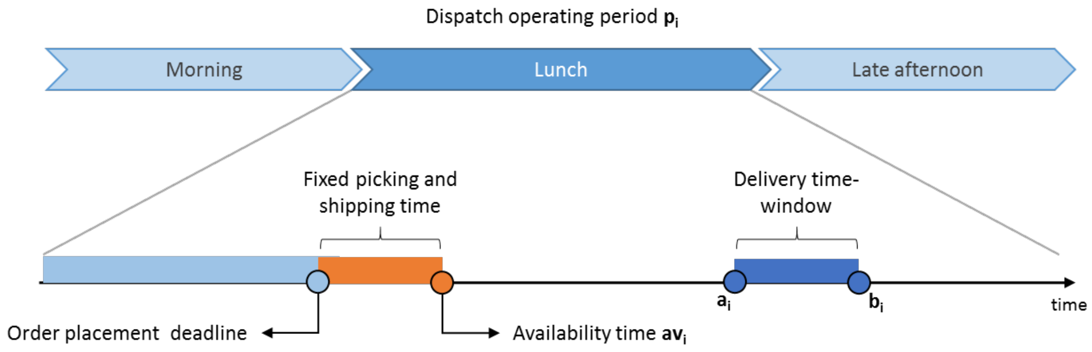

Figure 3.1: Variables defining the service level agreements

placement deadline. On each day, customers may or may not place an order on any of its ordering periods. As such, let $\mathscr{O}$ be the set of orders (i,t) placed by node i on day t, which are defined by their service time st_{it} and ordered quantity q_{it}.

Furthermore, a set $\mathscr{A}$ of arcs connecting nodes i and j on day t is considered. Note that the arc $(0,n+1,t)$ exists and will be used when a route is not to leave the depot on that day. Each arc has an associated travel time tt_{ij} and a travel distance td_{ij}, which are both independent of the day. These arcs are pre-processed in order to eliminate combinations that will never be used and are therefore unnecessary to use in the model. For each day t, in order for an arc to exist connecting node i to node j, both of them need to be either a regular customer node with orders placed on day t or one of the depot nodes, 0 or $n+1$. In case both i and j are customer nodes, they must belong to different customers ($c_i \neq c_j$) and their specific shipping periods, in case they both exist, must be the same.

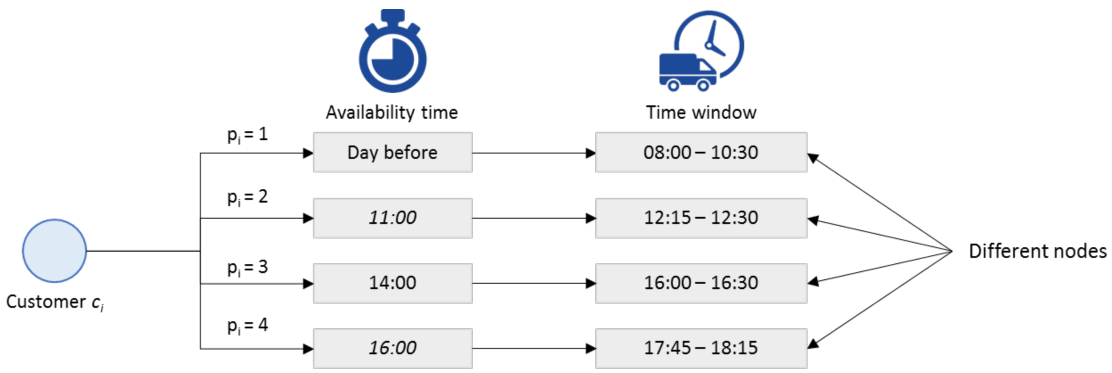

Figure 3.2: Different nodes belonging to the same customer

A set of routes $\mathscr{R}$ is also considered, whose size should be an upper bound estimation of the number of necessary vehicle departures. Each route r has an associated period rp_r, which defines which of the shipping periods it operates on. The vehicles that will perform the routes are considered to be homogeneous so a maximum capacity Q is defined. In order to adjust the model to labor related legal constraints, a maximum single route duration MD is also considered in the model. A depot time window $[aW, bW]$ is also present representing the operational hours of the depot and therefore defining the minimum departure time and maximum return time for all routes.

Sets and subsets

$\mathscr{A}$	set of arcs
$\mathscr{C}$	set of customers
$\mathscr{D}$	set of days
$\mathscr{N}$	set of nodes
$\mathscr{N}_C$	set of non-depot nodes
$\mathscr{N}_{TW}$	subset of nodes with time windows
$\mathscr{N}_P$	subset of nodes with shipping period
$\mathscr{N}_{OD}$	subset of nodes with order deadlines
$\mathscr{O}$	set of orders
$\mathscr{P}$	set of shipping periods
$\mathscr{R}$	set of routes

Parameters

a_i	earliest limit of node i 's time window
b_i	latest limit of node i 's time window
p_i	mandatory shipping period of node i
av_i	time at which orders from node i are available for shipping
n	number of customer nodes
st_{it}	service time of node i on day t
q_{it}	quantity ordered by node i on day t
td_{ij}	travel distance between node i and node j
tt_{ij}	travel time between node i and node j
rp_r	shipping period of route r
Q	estimated vehicle capacity
MD	maximum allowed route duration
aW	time at which the depot allows vehicles to leave
bW	time until which vehicles may return to the depot
M	big number

Decision variables

There are three main binary decision variables in this model: X^r_{ijt}, which defines each arc that is traveled by route r during day t; Z^r_i, which defines each node's route assignment; Y_r, which defines whether route r is active, i.e. if any customer is assigned to that route. Additionally, three time related decision variables are also considered, with W_{it} representing the daily arrival time at customer i, dW^r the planned departure time of route r and rW_t^r being the time at which route r arrives to the depot on day t. Moreover, since daily request patterns are stochastic, capacity and service level constraints might not be fulfillable at all times. As such, constraint softening decision variables were defined along with their associated objective function weights. Early and late arrivals at node i on day t are represented as ϕ_{it}^- and ϕ_{it}^+, respectively. ω_t^r defines how much route r exceeds maximum route duration L on day t, while θ_t^r is the excess of capacity on that same route and day.

$$X_{ijt}^r = \begin{cases} 1 & \text{if arc } (i,j,t) \text{ is traveled by route } r, \\ 0 & \text{otherwise.} \end{cases}$$

$$Z_i^r = \begin{cases} 1 & \text{if node } i \text{ is part of route } r, \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_r = \begin{cases} 1 & \text{if route } r \text{ is active,} \\ 0 & \text{otherwise.} \end{cases}$$

W_{it} time at which node i is served on day t

dW^r time at which route r is set to departure from the depot

rW_t^r time at which route r arrives at the depot on day t

ϕ_{it}^- earliness at node i on day t

ϕ_{it}^+ lateness at node i on day t

ω_t^r overduration of route r on day t

θ_t^r overcapacity of route r on day t

α weights of objective function components

3.2 Mathematical formulation

The conVRP-SLA's mixed integer linear formulation is as follows.

The objective function (3.1) primarily minimizes the total traveled distance since this is usually the main cost driver of transportation operations. A penalty for the number of routes is also included in case there is a fixed cost associated with their execution. The remainder parts represent the different penalties for breaking constraints regarding customer arrival times, route duration and vehicle overloads, respectively.

$$\min \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} \sum_{t\in\mathcal{D}} \sum_{r\in\mathcal{R}} td_{ij} \cdot X_{ijt}^r + \sum_{r\in\mathcal{R}} \alpha_r \cdot Y_r + \sum_{i\in\mathcal{N}} \sum_{t\in\mathcal{D}} (\alpha_1 \cdot \phi_{it}^- + \alpha_2 \cdot \phi_{it}^+) + \sum_{r\in\mathcal{R}} \sum_{t\in\mathcal{D}} (\alpha_3 \cdot \omega_t^r + \alpha_4 \cdot \theta_t^r) \quad (i,j,t) \in \mathcal{A} \tag{3.1}$$

Firstly, each node must be assigned to only one route (3.2). Also, each node must be visited exactly once, but only when it has placed an order on that particular day and by the route it was assigned to (3.3). To ensure a correct flow, the same reasoning applies to the number of departures from each node (3.4).

$$\sum_{r\in\mathscr{R}} Z_i^r = 1 \quad \forall i \in \mathscr{N}_C \tag{3.2}$$

$$\sum_{i\in\mathscr{N}}\sum_{t\in\mathscr{D}}\sum_{r\in\mathscr{R}} X_{ijt}^r = Z_j^r \quad \forall j \in \mathscr{N}_C, (i,j,t) \in \mathscr{A} \tag{3.3}$$

$$\sum_{j\in\mathscr{N}}\sum_{t\in\mathscr{D}}\sum_{r\in\mathscr{R}} X_{ijt}^r = Z_i^r \quad \forall i \in \mathscr{N}_C, (i,j,t) \in \mathscr{A} \tag{3.4}$$

Each route must also depart from the depot exactly once every day (3.5). In case no orders are placed by the nodes of route r on a given day t, then the existence of the arc $(0,n+1,t)$ allows for the fulfillment of this constraint. The return of the route to the depot is already assured by the basic flow constraints (3.3) and (3.4). If there are no nodes assigned to route r, then it is inactive (3.6). On the other hand, when at least one customer is assigned to a route r, then it is an active route (3.7).

$$\sum_{j\in\mathscr{N}} X_{0jt}^r = 1 \quad \forall t \in \mathscr{D}, r \in \mathscr{R}, (0,j,t) \in \mathscr{A} \tag{3.5}$$

$$\sum_{i\in\mathscr{N}_C} Z_i^r \geq Y_r \quad \forall r \in \mathscr{R} \tag{3.6}$$

$$Z_i^r \leq Y_r \quad \forall i \in \mathscr{N}_C, r \in \mathscr{R} \tag{3.7}$$

As all ordering windows of each customer are treated as different nodes, these have to be forced to be shipped in different routes (3.8). Moreover, when an order has a specific shipping period assigned to it, it can only be delivered by routes operating on that period (3.9).

$$Z_i^r + Z_j^r \leq 1 \quad \forall r \in \mathscr{R}, i,j \in \mathscr{N}_C : c_i = c_j \tag{3.8}$$

$$rp_r + M(1-Z_i^r) \geq p_i \geq rp_r - M(1-Z_i^r) \quad i \in \mathscr{N}_P, r \in \mathscr{R} \tag{3.9}$$

In order to enable the existence of time windows in the model, the arrival time at all customers must be traced. As such, the time at which service starts at a given node j must be greater than or equal to the time at which the previous node i was served plus its service time and the travel time between the two (3.10). Also, for each node, the arrival time at each day is compared to the target time window (3.11).

$$W_{it} + st_{it} + tt_{ij} \leq W_{jt} + M(1-X_{ijt}^r) \quad \forall (i,j,t) \in \mathscr{A}, i,j \in \mathscr{N} \tag{3.10}$$

$$a_i - \phi_{it}^- \leq W_{it} \leq b_i + \phi_{it}^+ \quad \forall (i,t) \in \mathscr{O}, i \in \mathscr{N}_{TW} \tag{3.11}$$

Both the depot departure and return times need to be coordinated with the first (3.12) and last (3.13) deliveries, respectively. Since exact departure and return times are only necessary for planning when the single route duration is over the limit, equality constraints are unnecessary as they would only jeopardize the performance of the model without any benefit.

$$dW^r + tt_{0j} \leq W_{jt} + M(1 - X_{0jt}^r) \quad \forall (0,j,t) \in \mathscr{A}, r \in \mathscr{R} \tag{3.12}$$

$$W_{it} + st_{it} + tt_{i0} \leq rW_t^r + M(1 - X_{in+1t}^r) \quad \forall (i,0,t) \in \mathscr{A}, r \in \mathscr{R} \tag{3.13}$$

In addition, vehicles need to leave and return to the warehouse during operational hours (3.14, 3.15). Also, each route must depart only after all of its orders are ready to be shipped. (3.16).

$$dW^r \geq aW \quad \forall r \in \mathscr{R} \tag{3.14}$$

$$bW \geq rW_t^r \quad \forall t \in \mathscr{D}, r \in \mathscr{R} \tag{3.15}$$

$$dW^r \geq av_i - M(1 - Z_i^r) \quad \forall i \in \mathscr{N}_{OD}, r \in \mathscr{R} \tag{3.16}$$

Furthermore, it is necessary to keep track of each vehicle's total load (3.17) and daily duration (3.18) in order to define the penalties for excesses on any day. These constraints and their associated objective function penalties are extremely dependent, not only on how much the decision maker values these trade-offs, but also on how precise the measure of durations and vehicle capacity is.

$$\sum_{i \in \mathscr{N}_C} q_{it} \cdot Z_i^r \leq Q + \theta_t^r \quad \forall r \in \mathscr{R}, t \in \mathscr{D} \tag{3.17}$$

$$rW_t^r - dW^r \leq MD + \omega_t^r \quad \forall r \in \mathscr{R}, t \in \mathscr{D} \tag{3.18}$$

Finally, integrality, binary and non-negativity conditions are set by constraints (3.19).

$$X_{ijt}^r, Z_i^r, Y_r \in 0,1; \quad W_{it}, dW^r, rW_t^r, \phi_{it}^-, \phi_{it}^+, \omega_t^r, \theta_t^r \geq 0. \tag{3.19}$$

Chapter 4

Solution approach

As is the case with most vehicle routing problems, the proposed model proved to be extremely hard to solve via exact methods as instances increased in size. In fact, using these methods, most possible applications of this problem are prone to need instances of unsolvable size. Therefore, a Fix and Optimize (FO) based approach was designed to tackle this problem by attempting to make use of its natural characteristics in order to solve larger instances. In this chapter, an overview of the methodology is given along with a detailed explanation of the proposed algorithm.

4.1 Methodology overview

In the conVRP-SLA, the main driver for complexity is the number of considered nodes as they increase the number of arcs and, consequently, the number of decision variables and constraints, exponentially. Any change in the route assignment of a particular node leads to differences in many parts of the solution, as routes need to be recalculated along with arrival time and all the other associated variables. However, this assignment is the cornerstone decision that the model needs to make, with everything else being dependent of the main binary variables Z_i^r. While changing a node may have a sort of butterfly effect throughout the remaining routes, distant routes tend to be somewhat independent from each other as a customer from one of them will most likely never join the other, especially under soft time window and capacity constraints. Hence, a solution method was devised with the goal of (1) reducing the number of nodes and (2) iteratively exploring the solution space. This method includes a pre-processing stage in which the dataset is manipulated in order to simplify the problem followed by the actual problem solving stage.

The pre-processing stage aims to decrease the number of nodes in the actual model. The proposed algorithm consists of finding nodes which are very close to one another compared to the overall distances of the instance and to consider them in the model as a single grouped node. These nodes also need to have a similar ordering period and similar time windows in order to be able to be served by the same routes. By doing this, the number of nodes is effectively being reduced with the downside of not allowing some groups of customers to be served by different routes. Such a procedure might be dangerous in areas of high node density as it may end up blocking the formation of good routes, hence why the parameters of the algorithm should be carefully defined for each instance. However, it is highly effective in less dense areas in which there are small groups of nodes very close to one another (e.g. a village), while being very distant from the rest of the nodes. In this case, it is safe to assume that these nodes will never be served by different routes.

The next step of the methodology focuses on actually solving the problem by trying to avoid approaching the whole instance at once. Taking advantage of the problem's natural focus on the binary decision variables Z_i^r and X_{ijt}^r, a FO procedure was designed that fixes most of the nodes and arcs and focuses on optimizing just a smaller part of the problem at each iteration. This allows for the exploration of a high percentage of the solution space without creating boundaries and while maintaining efficiency. This method requires a feasible initial solution, whose finding using the mathematical model is time consuming on larger instances. Therefore, an initial solution construction algorithm was also implemented to allow for fast start up times.

The overall solution approach consists of the described processes (Figure 4.1). The detailed definition of their individual steps is as follows.

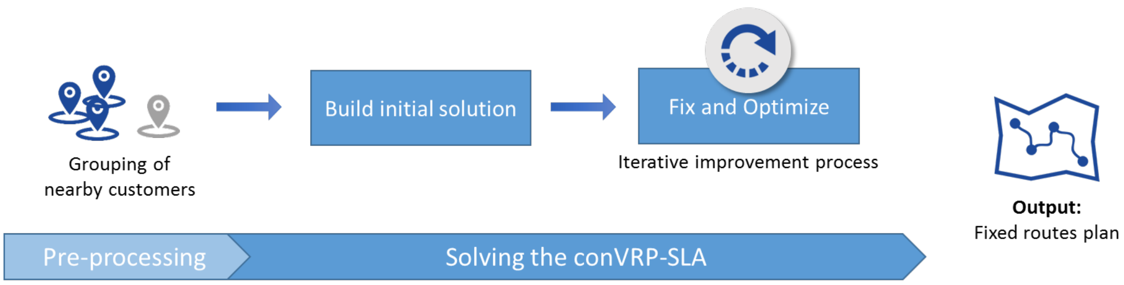

Figure 4.1: Overview of the proposed matheuristic solution process

4.2 Proposed algorithm

Node grouping

The node grouping step is based on the algorithm proposed by Dondo and Cerdá (2007) with the intent to reduce the computation effort of subsequent solution phases. The goal is to aggregate small groups of nodes with compatible time windows and that are geographically very close to one another when compared to the global set (Figure 4.2). This significantly reduces the number of variables and constraints of the model by assuming that those customers will always be served by the same vehicle, which is, most likely, the case. In order not to limit the ability of the model to allocate nodes to different periods, this stage is only performed for nodes with the same shipping period.

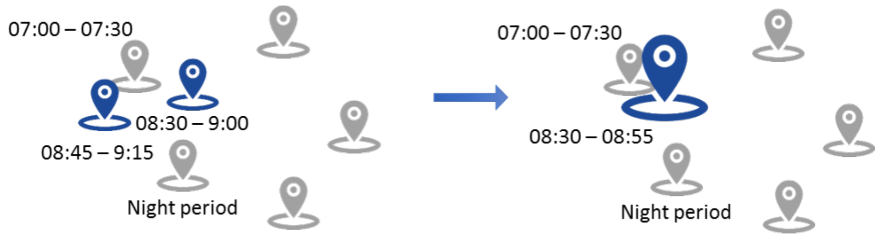

Figure 4.2: Node grouping to reduce the size of the instance

The grouping algorithm (Algorithm 1) starts with a pool of all the nodes from $\mathcal{N}$ ordered first by a_i and then by b_i, while requiring a maximum inter-node distance of $maxD$ and a maximum vehicle waiting time of δ_{wait}. Also, let $\mathcal{N}^*$ be a set of grouped nodes i^* to be populated by the algorithm and $\mathcal{N}'$ a temporary set of unassigned regular nodes. Then, and until the original set $\mathcal{N}_p$ is empty, groups of nodes i^* will be created, populated and then added to the final set of grouped nodes $\mathcal{N}^*$. Every time a new grouped node i^* is initialized, it is populated with the first node of $\mathcal{N}$, which is also removed from the original set. The temporary set $\mathcal{N}'$ is then created as a copy of $\mathcal{N}$ and iterated for each of its nodes i. Node i is added to the group i^* if they both share shipping period, if the distance to its closer node is less than the maximum allowed and if the time window is fulfillable. In order for this last condition to be verifiable, the time to serve all the nodes in the group needs to be traced. Every time a new node joins the group, both its own service time and the travel duration between itself and the node of the group which it is closest to are added to the previous group service time. The latest time window in the node should then be fulfillable assuming service starts at the beginning of the first node's time window and that a maximum delay of δ_{wait} is allowed. In the end, all distances $td^*_{i^*j^*}$ and travel times $tt^*_{i^*j^*}$ are recalculated respectively as $td_{i^*_n,j^*_0}$ and $tt_{i^*_n,j^*_0}$, assuming that the service of the new group of nodes starts at the first node and progresses according to their placement order.

While δ_{wait} should be defined as a relatively small period of time comparing to expected route duration as it represents how late it is acceptable to arrive to a customer in order for it to be grouped with nearby nodes, D_{max} should vary according to the node density of each specific instance. This is done to avoid excessively grouping nodes in high density areas, which would significantly hinder the model's ability to define the routes by acting as a simple routing heuristic.

Initial solution construction

The initial solution construction stage is very straightforward as its only purpose is to allow for a quick start of the next steps when there is no provided initial solution. During this step, it is important to evaluate how close a node is to another, but a simple concept of distance is insufficient when dealing with different periods of delivery and time windows. For this purpose, a compatibility score between nodes τ_{ij} is introduced, attempting to measure the distance between nodes and penalizing it when their time windows are not compatible. This measure is computed as shown in equation 4.1,

Algorithm 1 Pseudo-code for the node grouping stage.

function GROUPNODES($\mathscr{N}, D_{max}, \delta_{wait}$)
  $\mathscr{N}^* \leftarrow \emptyset$
  Order $\mathscr{N}$ by a_i, then by b_i
  repeat
    $i^* \leftarrow$ First $i \in \mathscr{N}$
    $\mathscr{N}' \leftarrow \mathscr{N} \leftarrow \mathscr{N} \setminus \{i^*[0]\}$
    repeat
      $i \leftarrow$ First $i \in \mathscr{N}'$
      if $p_i = p_{i^*}$
       and $min\left\{td_{j \in i^*}, i\right\} \leq D_{max}$
       and $q_{i^*} + q_i \leq Q$
       and $a_{i^*} + st_{i^*} + min\left\{tt_{j \in i^*, i}\right\} \leq max\{b_i, b_{i^*}\} + \delta_{wait}$ **then**
        Add i to i^* after closest node j
        $q_{i^*} \leftarrow q_{i^*} + q_i$
        $st_{i^*} \leftarrow max\left\{st_{i^*} + tt_{j,i} + st_i, a_i + st_i - a_{i^*}\right\}$
        $b_{i^*} \leftarrow max\{b_{i^*}, b_i\}$
        Remove i from $\mathscr{N}'$ and $\mathscr{N}$
      else
        Remove i from $\mathscr{N}'$
      end if
    until $\mathscr{N}' = \emptyset$
    Add i^* to $\mathscr{N}^*$
  until $\mathscr{N} = \emptyset$
  return $\mathscr{N}^*, td^*, tt^*$
end function

$$\tau_{ij} = td_{ij} + \left[\left(\frac{a_j + b_j}{2} - \frac{a_i + b_i}{2} - avg(st_{it}) \right) * \bar{v} - td_{ij} \right]^2 \tag{4.1}$$

with $\bar{v}$ being the average speed of the vehicles to allow for a comparison between time and distance. This speed can be computed from an average of the ratio between each of the travel distances and travel durations in the instance. In the special case in which some nodes are to be placed in specific routes, a subset of nodes $\mathcal{N}'$ and the respective list of node-route assignment variables Z' should also be provided. This can be especially useful if the algorithm is run on a previously found solution as an improvement mechanism.

Algorithm 2 Initial solution building.

---

function BUILDINITIALSOLUTION(Optional N', Optional Z')
  for all $i \in \mathcal{N}', r \in \mathcal{R} : Z_i'^r = 1$ **do**
    $i_{fix} \leftarrow r$
  end for
  for all $i \in \mathcal{N} \setminus \mathcal{N}'$ **do**
    $minCompat \leftarrow 0$
    $foundNode \leftarrow false$
    for all $r \in \mathcal{R} : rp_r = p_i, j \in \mathcal{N} : j_{fix} = r$ **do**
      if $minCompat = 0$ or $\tau_{ij} \leq minCompat$ **then**
        $minCompat \leftarrow \tau_{ij}$
        $i^* \leftarrow j$
        $foundNode \leftarrow true$
      end if
    end for
    if foundNode **then**
      $i_{fix} \leftarrow i^*_{fix}$
    else
      $i_{fix} \leftarrow random\, r \in \mathcal{R} : p_i = rp_r$
    end if
  end for
  for all $t \in \mathcal{D}, r \in \mathcal{R}$ **do**
    $\mathcal{O}_t^r \leftarrow (i, t') \in \mathcal{O} : i_{fix} = r$ **and** $t' = t$
    Order $\mathcal{O}_t^r$ by a_i, then by b_i
    for $k = 0$ to $size(\mathcal{O}_t^r) - 2$ **do**
      $X^r_{\mathcal{O}_t^r[k]\mathcal{O}_t^r[k+1]} \leftarrow 1$
    end for
  end for
end function

The construction algorithm (Algorithm 2) is performed in several steps. Firstly, routes are populated with the provided nodes $\mathcal{N}'$ in case there are any. Then, each of the remaining nodes i is assigned a fixed route i_{fix} by searching the subset of all the routes of compatible periods for the node which is closest to it. For this, the minimum compatibility score found so far is traced by *minCompat*, while boolean $foundNode$ tracks whether any node was found. The route of the closest node found is then selected for node i. In case no compatible node is found, the node is assigned a random route from all the period compatible ones. The value of variables Z_i^r are then set accordingly as 1 or 0 and an iteration through all the days and routes follows. In this step, orders are ordered by their time-windows and variables X_{ijt}^r are set assuming the route is performed in this sequence. As time windows are not hard constraints, as long as no period compatibility constraint is broken, every partial solution constructed in this way is feasible and by solving the model with the injected fixed variables, one is able to almost instantaneously fill out the remaining decision variables.

Fix and optimize

The last and main stage of the solution method is a neighborhood search improvement matheuristic based on a FO approach. The reasoning for this approach is grounded on the high computational burden of having a very large number of integer variables and on how easy it is to define a neighborhood as a set of nodes which are very close to one another. By defining a subset $\mathcal{N}_R$ as the set of all nodes which are going to be able to change their values from the current solution and subsequently limiting a very large portion of the integer decision variables, the MIP becomes substantially easier to solve. First, let a node be labeled as *fixed* if the variables Z_i^r are to maintain their values for every route r, and *released* if these same variables will be able to change values. Furthermore, let a route r be labeled as *fixed* if, for all nodes $i \in \mathcal{N}$, Z_i^r is to remain equal to the incumbent solution and *released* if the nodes are allowed to both join and leave this route. In practice, fixing a binary variable means imposing constraints on it, such that both their lower bound and upper bound are equal to the current value in the incumbent solution.

The overall FO algorithm (Algorithm 3) starts by setting a counter n_{noimp} to keep track of the number of iterations with no improvement to the incumbent solution and a time tracker ω to account for the time already expended in the algorithm. Then, a loop with three main steps runs until the time counter is over the limit ω_{max}. Firstly, the list of nodes to release $\mathcal{N}_R$ is computed following the current neighborhood size n_{nodes}. Then, both the node assignment variables Z_i^r and the arc usage variables X_{ijt}^r need to be either fixed to their current values or released. This is done according to the previously defined list of nodes to release and the relevant arc releasing parameters.

After these pre-processing stages, the model is solved and the next procedure depends on the found solution. Let sol represent the found solution, with sol_{obj} being its objective value and $sol_{\mathcal{V}}$ the full set of its variable values. If the incumbent solution is not improved, this is, if the found objective value is not better than the overall best obj_{min}, the counter of no improvement iterations is updated. Then, when the counter reaches a pre-established limit n_{max}, the neighborhood's size n_{nodes} is increased by $step$. This process raises the number of nodes that will be released in the next iteration, which allows for a larger search space that helps escape local optima. On the other hand, if the best overall objective value is improved, the new solution is saved as the new overall best by updating obj_{min} and setting the overall best set of variables $\mathcal{V}$ to current variable values. These variable values will be the ones used to fix variables in next iterations until a new best solution is found. In addition, if the current neighborhood size is larger than the initial one, n_0, it is reset to tighten the exploration space and therefore refocus the search on the newly found solution. This overall procedure is then repeated until the time limit is exceeded.

Algorithm 3 Overall Fix and Optimize algorithm.

$n_{noimp}, \omega \leftarrow 0$
repeat
  $n_{noimp} \leftarrow n_{noimp} + 1$
  $\mathcal{N}_R \leftarrow$ SelectNodesToRelease($\mathcal{N}, n_{nodes}$)
  FixModelVariables($\mathcal{V}, \mathcal{N}_R, \delta$)
  $sol \leftarrow$ solve conVRP-SLA
  if $sol_{obj} \leq obj_{min}$ **then**
    $obj_{min} \leftarrow sol_{obj}$
    $n_{noimp} \leftarrow 0$
    $\mathcal{V} \leftarrow sol_{\mathcal{V}}$
    if $n_{nodes} > n_0$ **then**
      $n_{nodes} \leftarrow n_0$
    end if
  else
    if $n_{imp} = n_{max}$ **then**
      $n_{noimp} \leftarrow 0$
      $n_{nodes} \leftarrow n_{nodes} + step$
    end if
  end if
  update ω
until $\omega \geq \omega_{max}$

The first neighborhood definition stage deals with the selection of which nodes to release (Algorithm 4). The first node to be released, i_0, is selected randomly between all the nodes in the set. Then, and until enough nodes have been selected, the node i^* with the highest compatibility score with the first random node is also selected to be released. By using the compatibility score (4.1), not only distance is being considered but also the compatibility of the time windows. Nodes are only considered for selection when they share the period of delivery with the first node or when they do not have a specific period for delivery. In the end, all the routes containing the selected nodes are released as well.

Algorithm 4 Node selection algorithm.

function SELECTNODESTORELEASE($\mathcal{N}, n_{nodes}$)
  $i_0 \leftarrow \mathcal{N}[random]$
  add i_0 to $\mathcal{N}_R$
  while $sizeof(\mathcal{N}_R) < n_{nodes}$ **do**
    $i^* \leftarrow i : \tau_{ii_0} = max\{\tau\}$
    add i^* to $\mathcal{N}_R$
  end while
  return $\mathcal{N}_R$
end function

After selecting the routes and nodes to be fixed and released, this information needs to be converted to set the actual binary variables of the model (Algorithm 5). Note that, since the time the model will take to explore the solution space is extremely dependent on the number of binary variables it considers, this decision is of utmost importance to assure the algorithm performs efficiently.

A new concept of a connection node is introduced representing the nodes that, on any of the days of the incumbent solution, are at a maximum of δ positions apart from any of the released nodes in their route. Each connection node will then have all the arcs connecting it to released nodes or other connection nodes of its route released as well. By doing this, released nodes will be able to move freely between connection nodes, while the connection nodes will remain in their current route. This position buffer δ will therefore define to what extent each released route will be freed for new nodes to be able to join it. This buffer allows for a more controlled release of arcs, as, in a very long route, if the neighborhood being optimized corresponds to just a small geographical part of it, it is not necessary to allow for the whole released route to be changed. A representation of fixed (1), released (2) and connection nodes (3) shows how these last nodes are defined based on the released ones around them (Figure 4.3).

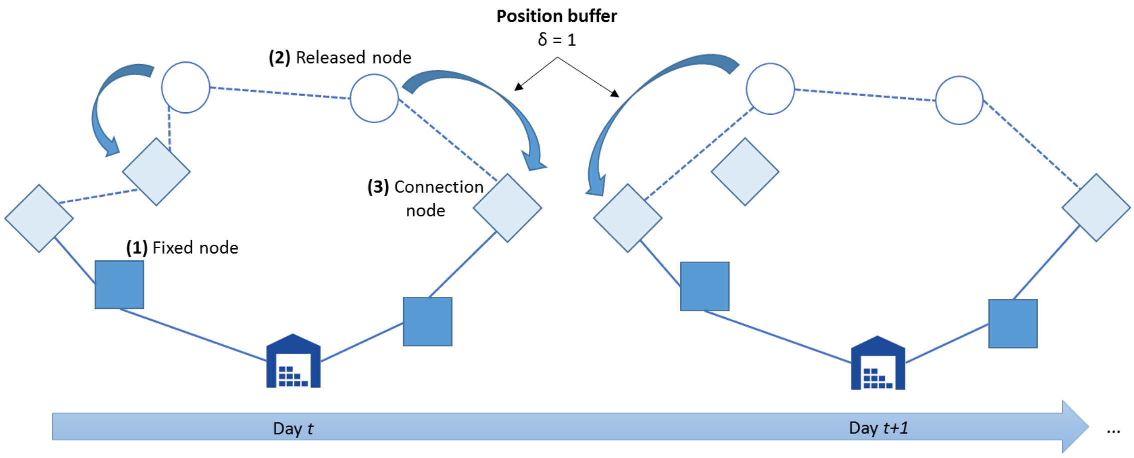

Figure 4.3: Different types of nodes in a route

Let $\mathcal{N}_F$ be the subset of fixed nodes, $\mathcal{N}_C$ the subset of connection nodes and $\mathcal{N}_R$ the subset of nodes to be released. Also, let similar subsets be created for routes, with $\mathcal{R}_F$ being the subset of fixed routes and $\mathcal{R}_R$ the subset of released routes. The first step is to define which nodes are going to be connection nodes. In order to achieve this, each order (i,t) is given an attribute, pos_{it}, defining in which position it is served. Then, for each route r on day t, if a node i is within δ positions of a released node, it is transferred from the subset of fixed nodes $\mathcal{N}_F$ to the connection nodes subset $\mathcal{N}_C$.

The next step is to set the binary variables of the model. Regarding node-route assignment variables, Z_i^r is fixed to its current value if $i \in \mathcal{N}_F \cup \mathcal{N}_C$, and it is released if $i \in \mathcal{N}_R$. As seen in Figure 4.4, an arc X_{ijt}^r is released in one of three situations. Firstly, if both $i \in \mathcal{N}_R$ and $j \in \mathcal{N}_R$, the arc is released for all $t \in \mathcal{D}, r \in \mathcal{R}_R$ (1), therefore allowing these nodes to freely join any of the

---

Algorithm 5 Model variables fixation algorithm.

---

function FIXMODELVARIABLES($\mathscr{V}, \mathscr{N}_R, \delta$)
    $\mathscr{N}_F \leftarrow \mathscr{N} \setminus \mathscr{N}_R$
    $\mathscr{N}_C \leftarrow \emptyset$
    for all $r \in \mathscr{R}, t \in \mathscr{D}$ **do**
        $(i, j, t, r) \leftarrow (i', j', t', r') : X_{i'j't'}^{r'} = 1 \wedge i = 0 \wedge t' = t \wedge r' = r$
        $pos \leftarrow 0$
        while $j \neq n+1$ **do**
            $pos \leftarrow pos + 1$
            $pos_{jt} \leftarrow pos$
            $(i, j, t, r) \leftarrow (i', j', t', r') : X_{i'j't'}^{r'} = 1 \wedge i' = j \wedge t' = t \wedge r' = r$
        end while
    end for
    for all $(i, t) \in \mathscr{O} : i \notin \mathscr{N}_R$ **do**
        for all $(j, t) \in \mathscr{O} : Z_i^r = Z_j^r = 1$ **do**
            if $|pos_{it} - pos_{jt}| \leq \delta \wedge j \in \mathscr{N}_R$ **then**
                Add i to $\mathscr{N}_C$
                Remove i from $\mathscr{N}_F$
            end if
        end for
    end for
    for all $i \in \mathscr{N}, r \in \mathscr{R}$ **do**
        if $i \in \mathscr{N}_R \wedge r \in \mathscr{R}_R$ **then**
            Release Z_i^r
        else
            $Z_i^r \leftarrow \mathscr{V}[Z_i^r]$
        end if
    end for
    for all $(i, j, t, r) \in \mathscr{A}$ **do**
        if $i \in \mathscr{N}_R \wedge j \in \mathscr{N}_R \wedge r \in \mathscr{R}_R$ **then**
            Release X_{ijt}^r
        else if $i \in \mathscr{N}_R \wedge j \in \mathscr{N}_C$ **or** $i \in \mathscr{N}_C \wedge j \in \mathscr{N}_R$ **then**
            Release X_{ijt}^r
        else if $i \in \mathscr{N}_C \wedge j \in \mathscr{N}_C \wedge Z_i^r = Z_j^r = 1$ **then**
            Release X_{ijt}^r
        else
            $X_{ijt}^r \leftarrow \mathscr{V}[X_{ijt}^r]$
        end if
    end for
end function

---

released routes in between other released nodes. In this stage, released routes may be from many different periods, which allows for nodes that do not have a specific one to freely change period of delivery. However, when just releasing these arcs, a node can only join another route if there are released nodes ordering in every day of the horizon in which it has also placed orders. This makes it hard for nodes to change routes unless the number of released nodes is very large, hence the connection nodes were introduced.

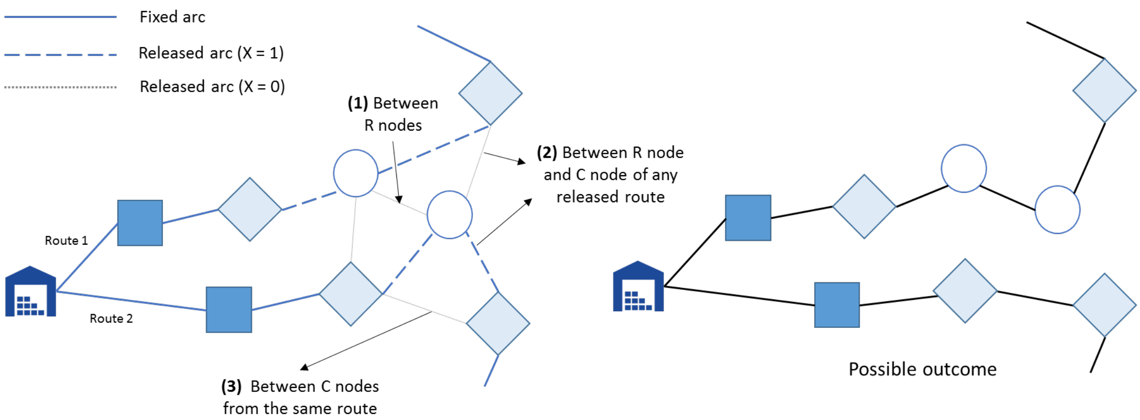

Figure 4.4: Different types of released arcs to allow for changes in customer assignment

In order to allow for every released node to join routes between connection nodes, all arcs X^r_{ijt} are released if either $i \in \mathcal{N}_R \wedge j \in \mathcal{N}_C$ or $j \in \mathcal{N}_R \wedge i \in \mathcal{N}_C$ for the current route of the connection node and for any $t \in \mathcal{D}$ (2). Finally, the arcs between connection nodes are also released for every $t \in \mathcal{D}$ if $i \in \mathcal{N}_C \wedge j \in \mathcal{N}_C$ (3). This is only done for route $r : Z^r_i = Z^r_j = 1$, as connection nodes will never change routes, so their arcs are only possible if they already belong to the same route. These arcs are released to allow the route to change configuration between the connection nodes if any released node either joins or leaves the route.

All the remaining arcs are fixed to their current values, effectively reducing the size of the problem by decreasing not only the number of binary variables but also the number of constraints. This makes the MIP much faster to solve, and is therefore suitable to apply in an iterative method in which many significantly smaller problems are solved. The size of the released neighborhood, which is highly dependent on both the number of released nodes and the position buffer, is the main driver for how the model will behave. When very few nodes and arcs are released, the model very quickly finds the best solution for each iteration. However, doing so hinders its ability to change the incumbent solution and may therefore not be able to escape from local optima. On the other hand, if many nodes and arcs are released, the model will solve very slowly and it will be hard to explore most of the solution space in acceptable amounts of time. As these performances vary according to the characteristics of the instance, the parameters should be fine tuned regarding the number of nodes and the amount of service level agreements. The settings used to solve the case study instances will be provided and can be used as benchmarks for new instances.

4.3 Approach for larger instances

When the number of days in the instance starts to increase, the proposed algorithm may still take too long to reach a good solution. This is because the workload of the matheuristic per released node is proportional to the number of considered days. When this is the case, the whole solution approach should be integrated within an increasing time horizon framework (Algorithm 6). This consists of starting the algorithm with only a small number of days and adding the rest of the planning horizon along the way. Having a smaller set of days enables the model to quickly explore the solution space regarding the basic route structure. The decision maker should decide on the minimum number of days *minDays* which ought to be sufficient to define a reasonable route skeleton which, in stabler operations, might be just a couple of days. The goal is to then enlarge the sample size to enable a more robust optimization, which acts as a stress test to the routes defined in the previous iterations. The step in number of days is defined as $dayStep$ and is applied at the end of every iteration.

Algorithm 6 Increasing horizon algorithm.

$nDays \leftarrow minDays$
$sol \leftarrow \emptyset$
while $nDays \leq sizeof(\mathscr{D})$ **do**
  $\mathscr{D}' \leftarrow \mathscr{D} : t \leq nDays$
  $\mathscr{O}' \leftarrow \mathscr{O} : t \leq nDays$
  $\mathscr{N}' \leftarrow i : (i,t) \in \mathscr{O}'$
  $\mathscr{N}', td, tt \leftarrow$ GroupNodes($\mathscr{N}', maxD, \delta_{wait}$)
  if nDays = minDays **then**
    BuildInitialsolution()
  else
    BuildInitialsolution($sol_Z, sol_{\mathscr{N}}$)
  end if
  $sol \leftarrow$FixAndOptimize
  $nDays \leftarrow nDays + dayStep$
end while

In every iteration, the set of considered days $\mathscr{D}'$ is defined by *nDays*, with the set of orders $\mathscr{O}'$ being the subset of all orders which were placed during the considered horizon, and with the set of nodes being reduced to only contain the ones that placed those orders. The matheuristic is then applied to the reduced instance, with the only difference being that the initial solution construction method is supplied with the current solution node-route assignment variables Z' in all but the first iteration.

Chapter 5

Case study in a pharmaceutical distribution company

The developed model and solution method were applied in a Portuguese pharmaceutical distribution company in order to plan their consistent routes for their operations in Portugal. This section will present an overview of the case study, how the necessary data was collected, the modifications made to the model in order to adapt it to the company's business model and all the additional tools that were developed in parallel to support the decision making process.

5.1 Case study overview

5.1.1 Company description

The company targeted by the case study is a distributor that is part of a pharmaceutical wholesaler that operates mainly in Portugal. The pharmaceutical industry business is characterized by its low margins for the distribution sector of the supply chain, for being a highly regulated market, and by belonging to a well-established competitive environment. The company mainly provides delivery services to a limited number of merchandise suppliers by specializing on the transportation of pharmaceutical items. These suppliers consist of other businesses of its parent-company, which need deliveries to pharmacies, supermarkets and perfume shops and also other large wholesalers.

The parent-company's final customers are well defined in the information system, place frequent orders and their packages are sent in uniform containers retrieved automatically from the company's warehouse. They may or may not have several deliveries in the different daily operational periods and they may place orders until a deadline, which is normally around 30 minutes before the assigned vehicle departure. On the other hand, the deliveries made for other external companies have fairly stochastic customers, non uniform containers and arrive at the warehouse in the beginning of the day. As there is no unique identifier for the recipient of these orders, each package comes with its own delivery address and is assigned to a route according to its postal code prefix. Moreover, the types of service level agreements in place are different from customer to customer, with some of them having a strict time windows, a specific period to be shipped in and an order deadline just in time of the route departure, while others have no restrictions at all (Table 5.1). These different delivery characteristics with distinct service level agreements complicate the route planning process significantly, especially when developing consistent routes, which makes it suitable as an application of the developed conVRP-SLA model.

Table 5.1: Main characteristics of the company's different customers

Customer origin	Container	Frequency	Time window	Order deadline	Route allocation
Internal	Uniform	Several daily deliveries	Strict	Before route departure	Robust routes plan
External	Unpredictable	Single delivery	Site-dependent	Beginning of the week	Postal code prefix

The quantity demanded by each customer is also stochastic, as well as the possibility of not ordering at all. There is a very subtle seasonality effect over the course of the year peaking in the cold season that is not very relevant for the distribution. However, the main concern during the planning stage is the beginning of each month in which the customers tend to increase the ordered amount up to 10% mostly due to increased budget availability and commercial target resets, which increases upstream pressure in the warehouse and challenges vehicle capacity.

Distribution paradigm

The company currently operates from five distinct warehouses, hereinafter named A, B, C, D and E. Each warehouse has a different driver cost structure, with some having a completely outsourced fleet, other having their own drivers and even a mixture of both situations. When the drivers are outsourced, the cost of the routes is roughly proportional to the traveled distance, while the company's own drivers have a fixed salary and are paid extra if overtime is needed.

All the warehouses have a robust routes system with a schedule being provided to the drivers in the beginning of each week. Orders are then automatically sent to their pre-established route, but the drivers are allowed to manually trade packages during the loading of the vehicles if they believe they have been wrongly assigned, which usually only occurs with external customers whose route was set according to the postal code prefix. Each route has a previously set departure time, but the short interval between the customer's order placement deadline and the planned departure is often not sufficient for the warehouse to dispatch the products in time for them to be loaded into the truck. Usually, the drivers depart whenever they have finished loading the vehicles.

In order to reduce the traveled distance and duration of some of the biggest routes, the company started implementing transshipments, in which a vehicle is loaded in the warehouse with the cargo of several routes and a transshipment point is set in order to perform the transfer. This is usually done in the very beginning of the supplier route in order not to further delay the deliveries of the receiving routes.

Current route planning procedures

The company currently defines its routes in a two step method. First, an instance with all the internal customers is created and is manually separated in geographical areas and in period of operation. The requests from the external customers, which are not period specific, are not considered during this phase because it is currently not possible to assign a route to each one of them on an operational level, even though they represent almost half of the total deliveries in one of the depots. Then, the nodes are used as input in a commercial route planner to define the robust routes for that period of operation. These routes are then used operationally until a new redefinition of the routes is performed, which does not have a defined frequency and is done very sporadically. All the customers that are not considered during the planning stage and, as such, do not have a specified route to be assigned to, are assigned daily on an operational level. A first rough assignment is made by having a manually created table that assigns every national postal code prefix to a route. However, some postal code prefixes correspond to a large area and some even have multiple routes going through them. In these cases, the drivers are given the task of reassigning some of the packages when they deem them to be in the wrong route. This procedure is time consuming and leads to mistakes and to deliveries that could be made in a better route. Moreover, the interchangeability of some deliveries, which may be delivered in any period of the day, is completely lost and every load is dispatched in the morning period unless it does not fit in the truck. This places a heavy burden on the morning routes, which often operate on tight schedules, and any delay in these early routes tends to spread to the remaining periods of the day as the drivers may be late for their next shift.

5.1.2 Objectives of the case study

The objective of this study is to provide the company with an overall assessment of their operations and to identify possible improvements by applying changes to the robust routes currently in place. These changes will be identified by using the developed mathematical model and solution method with historical data from the distributor. The main purpose of applying the model in this case study is to allow for a more cohesive plan that accounts for all the different types of customers served while taking into account all the different periods of delivery and the fact that external customers can be shipped in any of those periods.

The methodology used to evaluate the robust routes plan should not only be able to test how this plan would perform under real operational conditions but also to provide an analysis of whether the transshipments in place are correctly defined. Moreover, the methodology should allow for testing what impact would different conditions, such as focusing the distribution on just serving the internal customers, have on the total cost of operations.

5.2 Data analysis and and key performance indicators

The first step of the case study was to gather the necessary data in order to achieved the proposed objectives. As the distribution company operates within a broader pharmaceutical logistics enterprise, it mainly operates with its information systems. However, only a portion of the delivered goods are from customers of their own parent-company, with approximately 25% of their business originating from other external companies. The quality and reliability of the data, especially concerning delivery locations and the service level agreements, varies according to which company the final destination ordered the goods from. The only source of data which allowed for an analysis of the whole distribution network was a spreadsheet containing delivery information. Each of these deliveries has the following information specified in the spreadsheet:

- Warehouse and date of dispatch;
- Route and driver which performed it;
- Vehicle departure and return times;
- Customer and destination address;
- Number of boards and boxes delivered and their total combined weight;
- Time window and actual arrival time.

As an important part of the data was either missing or wrong, substantial work was done in order to assure its quality. Firstly, in order to find out where each delivery was made, some addresses had to be corrected and an algorithm was developed to retrieve its coordinates and to flag the ones with low certainty, which were then manually corrected. Then, whenever route durations or total distance was either missing or invalid, they were replaced by the average of that same route. Finally, the delivered quantity is measured both in number of boards and boxes, with the boards having a fixed volume while the boxes may be anything from a small package to a full pallet. This made it extremely hard to estimate vehicle capacity, which motivated a study of the weight/boxes in order to provide a unified measure of volume. Hence, orders were divided into two groups over whether they exceeded a $70\,kg/unit$ limit, in order to try and separate pallets from regular boxes. This limit was defined visually as there were clearly two distinct groups when the whole dataset was plotted together. Then, a regression was made for each group to estimate the marginal weight of a pallet and a box (Figure 5.1).

The weight of a pallet is therefore estimated as $166.7Kg$ and the weight of a box as $4.1kg$ according to the regression. The volume taken by each order, V, was then globally estimated from the number of boards n_{board}, the number of boxes n_{box} and the total weight W_{order} in kilograms (Equation 5.1). Each smaller box was visually estimated to be comparable in volume to a board and a pallet to be approximately the same size as 30 boards. Not only the volume taken by each order but also the capacity of the vehicles will therefore be measured in number of boards.

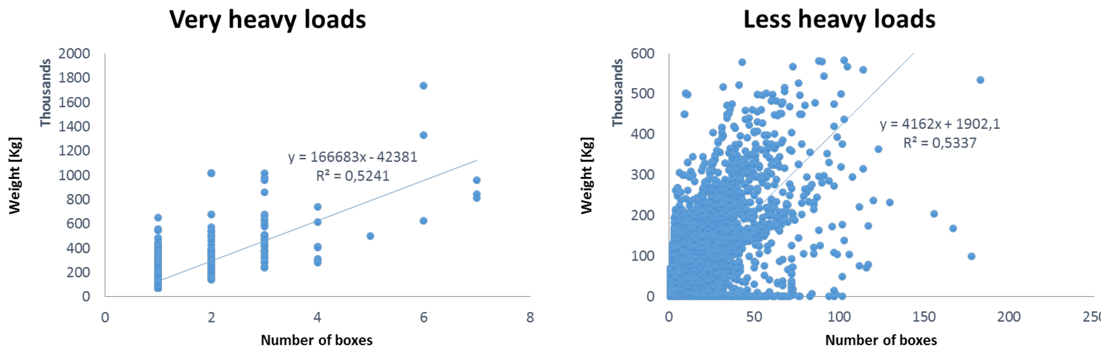

Figure 5.1: Regressions of the weight of the carried load as a function of the number of boxes

$$V = n_{board} + 30 * \left\lfloor \frac{W}{166.7} \right\rfloor + \frac{W - 166.7 \left\lfloor \frac{W}{166.7} \right\rfloor}{4.1} \quad [\text{number of boards}] \tag{5.1}$$

In order to facilitate the analysis of the data to enable the definition of meaningful and reliable Key Performance Indicators (KPIs), a complete database was created from scratch with Microsoft SQL Server and the data retrieved from the distinct company spreadsheets consolidated and inserted into it. The relevant tables, unique identifiers, attributes and relationships had to be defined according to both the business structure and the available data. Furthermore, the cost structure of the company was analyzed in order to identify the elements which would be influenceable by the project. After discussing the objectives with the decision makers, the relevant costs were defined as the overtime costs of all the warehouses, the costs of outsourced distribution, the fuel and maintenance costs of the company's fleet and the vehicle rental cost from one of the warehouses in which the company considers reducing the fleet size. Methods were then added to the database in order to automatically calculate the cost elements, taking into consideration all the relevant taxes and other fiscal and legal considerations.

The new and consolidated data structure allowed for a much more practical and insightful data analysis. As there were no specific KPIs defined from the start, a dashboard was developed with Microsoft Excel 2013 to provide a fully interactive environment to analyze the performance of the company. This report proved to be extremely useful to the company as there was currently no business intelligence software for the decision makers to look at their consolidated data and to execute an in-depth analysis of operational performance. After discussion with the company's decision makers, the project focused on three main total indicators, the total traveled distance, total route duration and percentage of late deliveries.

5.3 Methodology used to solve the case study

In order to achieve the proposed objectives, an approach (Figure 5.2) was developed to tackle the specific planning problem of this company by making use of the developed conVRP-SLA model and solution method with the necessary adjustments and auxiliary tools.

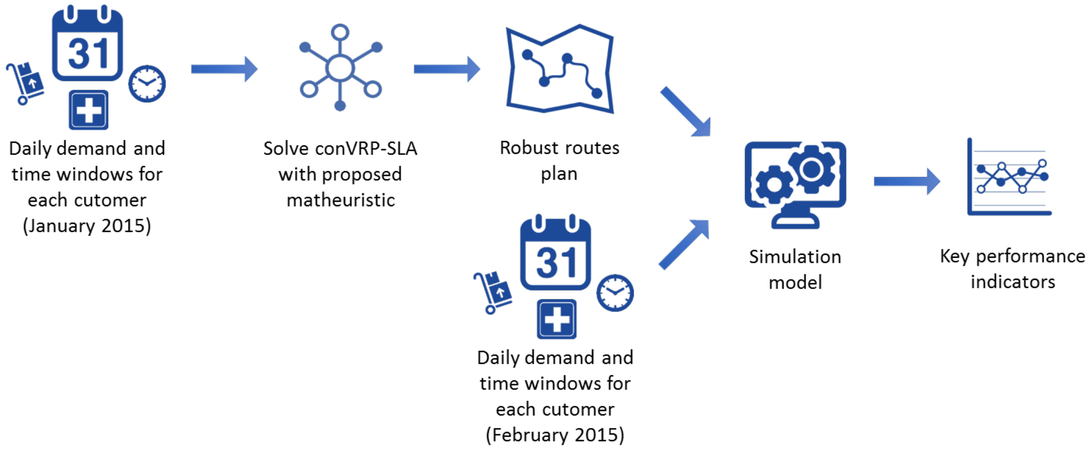

Figure 5.2: Methodology used in the case study

The development of the robust routes was made by using the model with real data from a previous month, looking for which set of routes would have responded best to that particular demand pattern. After the routes have been defined, an algorithm was developed to propose different possible transshipments. These are not included in the model directly as they are subject to different operational constraints that may suddenly force the stoppage of the transshipment, which means that both the supplier and the receiving routes should be good routes independently. Finally, in order to validate the proposed routes and to compare them with current operations, a simulation model was developed in order to evaluate how the new plan would perform under real requests. The simulation was done with real historical data from all the weekdays of February in order for the results to be comparable with the real performance reports from that month.

5.3.1 Route planning

Firstly, the route planning stage is done by directly applying the conVRP-SLA model and solution method with real data from the month of January. Since seasonality is often only a factor within each month and not so much over the year, with the peak happening on its first week, the first five weekdays of January were used as the planning horizon. This period is considered to be enough by the decision makers at the company since almost every consistent customer is bound to order at least once during a full week period. Also, by using the peak period, the resulting routes should be more robust and allow for better service levels, which is also one of the main goals of the company in this project. As there is no current foreseeable major alteration to the monthly demand pattern, a set of routes that would perform solidly in this month is considered to also be solid for the remainder of the year. In case any big customer changes occur meanwhile, the company should adjust to them by making a new plan. Each warehouse has different customer characteristics regarding both geographical dispersion and ordering habits. Moreover, the driver payment method and, therefore, the weights of the different cost components on the total cost, also differ from depot to depot. For these reasons, the planning of the routes will be made separately for each warehouse in an attempt to account for as many specific characteristics as possible.

In order to speed up the solution procedure, the current routes of the customers, when a specific assignment existed, were used during the initial solution building stage instead of starting from scratch. Changing the route assignment of a customer usually implies a different average arrival time, which will start a process of redefinition of service level agreements between the commercial department of the company and the customer. As in the current situation every internal customer already has an expected arrival time and these were set according to the robust routes in place, there was a need to define whether each time window should be kept or discarded, or else it would be impossible to redefine the routes maintaining all arrival times. After discussing the issue with the decision makers, it was agreed that the first 40% of clients in each route should be considered as more important and therefore maintain their time window.

Another important factor in the planning stage is the delivery frequency. Orders from the main customer have very consistent destinations, with most of them ordering at least once every single day, whereas destinations from other external customers may have as few as one single delivery on record. These non recurring destinations have very little impact on the routes performed daily and therefore the benefit of including them in the robust routes is not worth the additional computational effort required. Thus, only the destinations with at least one delivery per week will be considered, while every other destination will be assigned to the route with the node closest to it in the simulation stage.

An important piece of data which is not available in the company's information system is the service time for each destination, which varies significantly from customer to customer, going from as low as one minute to as much as one hour in busier places such as big delivery hubs or hospitals. As no clear distinction between the type of destination can be made from the available data, the service time in minutes (Equation 5.2) was approximated according to a statistical study recently performed by the planning team in which actual service times at many different destinations were measured.

$$st_{it} \approx 2.5 + 1.5 * \left\lfloor \frac{q_{it}}{6} \right\rfloor \tag{5.2}$$

There is a fixed component present at all deliveries which accounts for parking, picking, walking time and paperwork and also a variable component which accounts for the need of additional trips back and forth to the vehicle when the driver cannot carry the whole load at once. The quantity considered is the measure approximated from the number of boards and boxes actually delivered. Finally, both the travel distance and travel duration matrices need to be defined, and these values need to be reliable since they are the main drivers for all objective function components. In order to achieve this, all of the destinations involved in the case study were georeferenced using the Google Maps Appplication Programming Interface (API), and then the matrices were constructed unsymmetrically for each warehouse by using the same software to retrieve both travel distance and duration for each given pair of destinations. Note that the values retrieved are deterministic and chosen by the software, which in most cases will correspond to the actual performed path but may also be very different in both aspects.

5.3.2 Transshipment evaluation

With distance and working hours being two of the main cost drivers in the distribution business, reducing their impact is one of the main goals in route planning. For this reasons, transshipment are already performed by the company. An assessment of the current practice and suggestions of possible new transshipment points were within the objectives of the case study. Hence, an algorithm was developed that, given a set of robust routes, ranks each pair of routes *supplier-receiver* in order to suggest which are the best transshipment opportunities. The evaluation will only be done for the first three periods of the day since in the remaining periods additional delays may cause vehicles to miss pharmacies' opening hours. There are logistical factors involved in this tactical stage that stretch beyond the route planning scope, such as finding a driver who lives close to the transshipment point and judging whether the caused delays in both routes are acceptable. Yet, it is still possible to ensure there is a vehicle capable of carrying both routes' load. The final decision of whether transshipments are doable or not will still be left to the decision maker.

There are two main factors that indicate whether a transshipment is good or not. Firstly, the furthest away the receiving route starts from the depot, the more distance saving potential there is, making it a good candidate for a transshipment. However, just starting far away from the depot is not enough as the supplier route needs to be taken into account as well. The closest a different route is from that potential receiving route, the better candidate it is for supplying it in a transshipment. Also, when evaluating transshipments, it is important to take into account that they should be done as early in the route as possible, as any deliveries made prior to it provoke delays on both routes. However, it might be the case that a specific route is very good for receiving the transshipment but it is currently serving a couple of customers close to the depot before traveling to the farthest destinations. These closer customers might be able to change route without major losses, and, therefore, the first three customers of each route are considered during this evaluation.

Algorithm 7 Transshipment evaluation algorithm.

$\mathscr{T} \leftarrow \emptyset$
for all $r_r \in \mathscr{R}$ **do**
  $ds_{r_r} \leftarrow max\left\{td_{0i}\ \forall i : \overline{pos_i^r} \leq 3\right\}$
end for
for all $r_s \in \mathscr{R}, r_r \in \mathscr{R} : rp_{r_s} = rp_{r_r}$ **and** $rp_{r_s} \leq 3$ **and** $\overline{q_s} + \overline{q_r} \leq Q$ **do**
  $c_{r_s r_r} \leftarrow avg\left\{td_{ij}\ \forall i,j : \overline{pos_i^{r_s}} \leq 3 \textbf{ and } \overline{pos_j^{r_r}} \leq 3\right\}$
  Compute $\Psi_{r_s r_r}$
  Add (r_s, r_r) to $\mathscr{T}$
end for
Order $\mathscr{T}$ by $\Psi_{r_s r_r}$

In order to evaluate transshipment possibilities (Algorithm 7), let the pair of routes (r_s, r_r), which represent the supplier and receiving routes respectively, define a transshipment, with $\mathscr{T}$

being a set of transshipments. Also, let $\overline{pos_i^r}$ be the average position in which node i is served by route r throughout the days. The maximum capacity is defined as Q, with $\overline{q_s}$ and $\overline{q_r}$ being the average loads of the supplier and receiving routes. Each route r is assigned a distance score ds_r which is the maximum distance to the depot between its first three nodes. Then, each transshipment is given a compatibility score $c_{r_s r_r}$ which is the average distance between pairs of the first three customers of each route. Finally, an overall transshipment score $\Psi_{r_s r_r}$ (Equation 5.3) is computed as the distance score of the receiving route multiplied by a factor which compares the actual distance score with the compatibility measure between both routes.

$$\Psi_{r_s r_r} = \frac{{ds_{r_r}}^2}{c_{r_s r_r}} \tag{5.3}$$

The set of transshipments is thus ordered accordingly, with the ones with the most potential benefits appearing on top of the list.

5.3.3 Simulation model

After the planning stage is finished, with both the robust routes and the transshipments to be made already defined, it is crucial to be able to evaluate how they will perform under real conditions before implementing the proposed changes in the operational level. For this reason, a model was developed in order to simulate the day-to-day distribution paradigm of the company from a given robust routes plan (Figure 5.3).

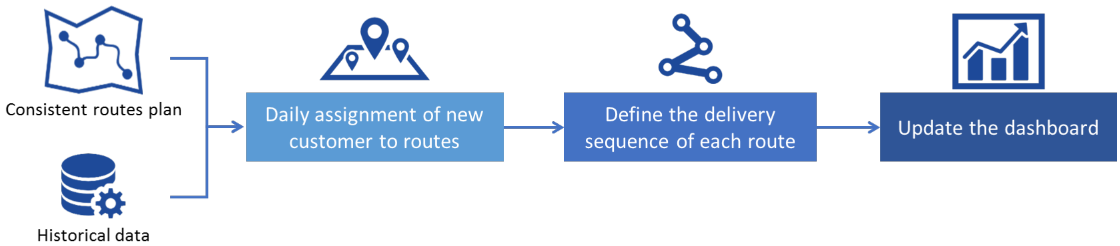

Figure 5.3: Overview of the simulation model

The robust routes plan and the historical data, along with some parameters, are the main inputs of the model. The plan consists of a simple allocation of the main customers considered during the route optimization phase to the routes. The data that will be used during this stage consists of the real demand of the month of February, which was not used during the previous planning steps. By using different days for each stage, the results of the simulation are not influenced by having routes planned specifically for them and they are therefore actually tested for their robustness. The simulation logic is presented in Algorithm 8, whose detailed description follows.

As this set of data considers every delivery made by the company and the routes were planned without customers with less than one weekly delivery, only some nodes, $\mathcal{N}_{Plan}$, will have a pre-established route. Also, since the customers themselves are stochastic, new destinations which have never ordered before are very likely to appear. Hence, the first step of the simulation is to

Algorithm 8 Simulation algorithm.

for all $i \in \mathscr{N} : i \notin \mathscr{N}_{Plan}$ **do**
  Look for closest node $j \in \mathscr{N}_{Plan}$ in a compatible route r
  Assign the route r to i
end for
for all $r \in \mathscr{R}$ **do**
  Define departure as the planned start time
  Define the return depot node's location
end for
for all $r_s \in \mathscr{R}_S$ **do**
  Insert fictious order in the transshipment point
  for all $t \in \mathscr{D}$ **do**
    Run TSPTW model for day t of route r_s
  end for
  for all $r_r \in \mathscr{R}_R, t \in \mathscr{D}$ **do**
    $dW^{r_r} \leftarrow W_{transt} + st_{transt}$
    Run TSPTW model for day t of route r_r
  end for
end for
for all $r \in \mathscr{R}, t \in \mathscr{D} : r \notin \mathscr{R}_S \wedge r \notin \mathscr{R}_R$ **do**
  Run TSPTW model for day t and route r
end for
Upload all data parameters to database

choose which route is going to serve requests which do not have a specific allocation. In practice, this is done automatically by the postal code prefix assignment, with the drivers later deciding whether the package is being shipped on the right route or not. As the postal code prefix data is not reliable, the same procedure cannot be followed by the model. For this reason, an equally simple allocation will be performed by choosing, from all the routes that are able to fulfill its service level agreements, the one with the node which it is most compatible with, using the compatibility measure previously defined (Equation 4.1). While this simple decision tries to mimic the one made by the drivers, it has the benefit of enabling tests of whether they are currently making good decisions or if there would be any benefits in applying a more robust methodology instead of the rough postal code prefix allocation. In order to correctly simulate the current situation, and also to allow for validation of the next steps of the simulation, the model is also able to skip this stage and assign the nodes exactly as they were assigned in the month being analyzed.

After the requests that each route will have to serve on each day are known precisely, the next step is to actually calculate all the parameters needed to define how they are performed for the whole simulation horizon. The main decision to be made in this stage is in what order each request will be served and at what time the vehicle will leave the warehouse, as all the other relevant parameters are easily computed with this information. The distances and durations used during this stage will be the deterministic ones retrieved from the Google Maps API. The model will assume that the vehicles depart from the warehouse at exactly the expected departure time since the data on this aspect is not very reliable either. When testing the current situation, the delivery sequence will be considered the same as the one actually performed by the drivers. On the other hand, whenever a new plan is being tested, these parameters will be calculated through a mathematical model. This model will solve the basic TSPTW problem (Gutin and Punnen, 2002) with the addition of the two-commodity flow subtour elimination constraints as described by Öncan et al. (2009). These constraints are useful as they allow for the optimal solution to be found in less time, because since the time windows are softly defined and breaking them is allowed, additional hard-constraints dealing with subtours improve the lower bound significantly. This particular formulation was chosen for its simplicity and for the good reported results on not only lower bound improvement but also on the time to solve.

An instance is then created for each route and each day of the simulation horizon. The nodes are the destinations to be visited, with the respective time windows and with service time and ordered quantity approximated as previously described. Both the departure and return depot nodes generally correspond to the warehouse being simulated. There are however some situations in which some of these parameters are altered.

Firstly, some of the company's own drivers are allowed to take the vehicles home and therefore do not return to the warehouse after serving the customers. In these situations, the return depot node is altered to be at the driver's approximated home address. Let $\mathscr{R}_S$ be the subset of routes which supply transshipments, r_s, and $\mathscr{R}_R$ the subset of routes that receive transshipments, r_r. When a route that supplies a transshipment is being simulated, an additional fictitious order is inserted at the transshipment location. Also, let nr_{r_s} be the number of routes receiving their cargo from r_s, and Q_{trans} the total number of units being transferred. The transshipment order will have a time window limit equal to its own route's departure time so as to enforce it do be served in the beginning. The service time st_{trans} (Equation 5.4) in minutes was estimated according to indications from the decision makers at the company.

$$st_{trans} \approx 5nr_{r_s} + \frac{30}{100} Q_{trans} \tag{5.4}$$

There is a fixed component of 5 minutes for each route that needs to receive its load from the supplying one, representing the time taken to move the vehicles to a suitable position to perform the transshipment, and a variable component of half an hour for every 100 units that need to be transferred from one vehicle to the other. Lastly, if a receiving route r_r is being simulated, the departure node is placed at the transshipment location and its departure time is equal to the arrival time at the transshipment point, W_{trans}, plus the time it takes to transfer the load.

After generating all the instances, the TSPTW model is run in order to calculate all the relevant parameters. For each route and day, the arrival time at each destination, the total traveled distance and the depot return time are saved and inserted into a database that is built with the same structure as the one used to analyze the current situation and to measure relevant KPIs, which were previously described. By doing this, it is possible to repopulate the dashboard with the simulated data instead of the real one and very quickly compare the results regarding all the defined KPIs.

Also, as the naming of every route is still the same, it is possible to easily see which customers are switching and between what routes. Moreover, it will be easier for the decision makers to validate the results as they will quickly recognize the route for which they already know what customers it serves and whether it has any arrival time or capacity issues.

5.4 Results

In this section, the results obtained from solving the case study are provided. (1) Firstly, the current route plan is simulated and compared to the real results from the company in the same planning horizon. (2) Then, the main characteristics of the solved instances are described, with the relevant parameters for the intermediate steps also being provided. (3) The results of the application of the conVRP-SLA model and matheuristic, followed by the transshipment evaluation algorithm, to plan the routes of the company are then presented. (4) Finally, the simulated results, which evaluate the impact of the planned routes on the actual operational KPIs are shown and discussed for each warehouse and on a general scope as well.

5.4.1 Validation of the simulation model

While the simulation model is quite simple in nature, differences between the expected outcome and the actual results still occur. In order to validate the process when new routes are being tested, the current robust routes plan was provided as the input along with the actual order in which the customers were served, with the results being compared with the real measured KPIs (Table 5.2).

Table 5.2: Comparison between simulated and real results

Depot	Total distance [km]	Total duration [h]	% of late arrivals	Total cost [€]
A	−5.1%	+5.6%	−6.7%	−7.2%
B	+7.9%	+12.5%	+22.1%	+15.8%
C	+12.4%	+11.7%	−3%	+16.1%
D	+3.0%	−17.7%	−8.1%	+2.6%
E	+6.9%	−10.7%	−0.5%	0.0%
Total	+5.3%	−7.6%	−5.7%	+4.2%

As the overall simulated results are different from the real ones, a more detailed, route by route analysis was made to identify the main factors that explain them. Firstly, the differences in the total traveled distance come from three main sources. There are often distance peaks due to errors in the database or isolated events such as a special unaccounted delivery or a trip to the maintenance center. Sometimes deliveries are registered in the wrong route, which makes that route appear longer while the correct one becomes shorter. Also, the drivers might take different paths from the ones considered best by the geographical information system and the fact that some of them take the vehicle home also influences the final results. When these factors do not occur, the simulated distance is very reliable, whereas the arrival times are significantly harder to estimate. Drivers rarely depart at the expected time from the warehouse, as they might leave as soon as they are finished loading the vehicle or they might have to wait for delays in the picking and loading operations. Also, both the travel durations and the service times are the least reliable approximations as their real value depends on many factors for which there is no data available. For example, unpredictable events, such as accidents, vehicle malfunctions or traffic jams might highly influence the travel durations. Late arrivals are even harder to track due to the unpredictable nature of the route durations and the fact that a small deviation in the arrival time may make an otherwise on-time delivery miss its time window.

After discussing these results with the decision makers and with most of the factors causing the differences identified, it was agreed that the simulation as it stood was a valid method to test new robust routes plans and distinct demand patterns.

5.4.2 Instance characteristics

As the model and solution method were tested as a tool to plan the routes of the case study's company, and since the warehouses operate independently from one another regarding distribution, instances were created for each one of them. These instances were generated from real historical data from the company's operations during the month of January and the most relevant parameters are shown in Table 5.3. The larger warehouses had to be analyzed in clusters as the model would take a very long time to load all the variables with the full instance, which would make the fix and optimize algorithm perform poorly. These clusters were defined by calculating the centroids of the routes, which were then used to perform a *k-means* clustering algorithm in RStudio.

The number of clusters was kept to a minimum value which allowed for the FO algorithm to load the model in a matter of seconds during each iteration. The node grouping parameters were then set for each depot, with the maximum inter-group node distance D_{max} being defined according to the geographical density of the customers, whereas the maximum deviation from the time window δ_{wait} was set to the delay value considered to be acceptable by the company. The depots situated in less dense areas have higher distances as, generally, there are no different routes serving areas within a few kilometers from each other. The different objective function weights were set based on discussions with the decision makers. The weights of the delays are higher in areas in which customers are more likely to complain due to late deliveries, and the weight for number of routes only exists in depots whose fleet is partially owned by the company and in which drivers are company's employees.

The instances were then generated for each cluster of every warehouse. The number of considered nodes is effectively being reduced from the real number of customers by not considering non-recurring destinations in the planning stage and by applying the node grouping algorithm. All the characteristics that indicate the size of each of the solved instances, specifying the results of each node reduction step, are shown on Table 5.4.

Table 5.3: Parameters used to define the instances

Depot	**Number of clusters**	$D_{max}[km]$	$\delta_{wait}[min]$	α_r	α_{ϕ^-}	α_{ϕ^+}	α_ω	α_θ
A	1	5	15	0	0.5	1	2	1
B	1	3	15	0	0.5	1	2	1
C	2	1.5	15	0	0.5	2	2	1
D	5	0.5	15	100	0.5	2	2	1
E	2	1	15	250	0.5	1	2	1

These instance sizes, along with the neighborhood exploration parameters that will be discussed subsequently, should be used as benchmarks for other implementations of the method, while considering that larger instances might still be efficiently solved in case more computational power is available.

Also, with the number of variables and constraints in the model increasing exponentially with the number of nodes, the reduction algorithm is able to significantly reduce the size of the instances, achieving a reduction of over 20% in most cases. The biggest warehouse whose routes were planned during this case study has over 1900 different customers, makes more than 5000 weekly deliveries and operates with 101 daily routes.

5.4.3 conVRP-SLA results

The proposed model and matheuristic were developed in IBM ILOG Concert Technology with a C# implementation and were run on a personal computer with 12GB of random access memory and a i7-4710HQ 64-bit processor with 8 threads and maximum frequency of $3.5GHz$.

In order to run the matheuristic, the FO parameters had to be defined in such a way that the algorithm would be able to release as many nodes as possible while still having time to perform enough iterations to explore the whole instance. This definition was not trivial and several test runs had to be made in order to adjust their values to the instances being optimized, as the results

Table 5.4: Main characteristics of the solved instances

Instance	**Total customers**	**Recurring customers**	**Grouped Nodes**	**Grouping % reduction**	**Days**	**Routes**	**Orders**
A	197	152	109	-28.3%	5	12	439
B	441	232	166	-28.4%	5	18	623
C1	319	268	181	-32.5%	5	16	762
C2	445	321	253	-21.2%	5	19	905
D1	367	338	304	-10.1%	3	29	817
D2	257	220	166	-24.5%	3	12	446
D3	457	384	283	-26.3%	3	21	774
D4	453	376	274	-27.1%	3	23	707
D5	397	320	220	-31.3%	3	16	578
E1	609	435	340	-21.8%	3	21	683
E2	445	295	259	-12.2%	3	14	521

are highly dependent on how these parameters are set. The values that were used in the instances solved in this case study should be used as a benchmark for other applications. The position buffer δ was kept with a value of 2 since nodes that are more positions apart are already quite far from each other in most situations. The initial number of nodes in the released neighborhood, n_0, was set as 15 to allow for the selection of a few nodes from the whole daily horizon in a small geographical area. The maximum number of iterations without improvement before increasing the neighborhood size, n_{max}, was set as 20, with the neighborhood size increasing by 10 nodes at a time. Finally, the time limit for each instance ω_{max} was set to 3 hours.

The total number of iterations, the objective value of the initial solution, that was defined using the routes currently made by the company, and the best found objective value are shown on Table 5.5. Note that both objective values represent the sum of all clusters for a given depot. The total variation between these two objective values represents how much the model would be able to improve the results if no stochastic element was present in daily operations.

Table 5.5: Results of the conVRP-SLA applied with the developed matheuristic

Depot	n_{iter}	**Initial objective value**	**Best objective value**	**Var**	**Initial distance [km]**	**Best distance [km]**	**Var**	**New trans-shipments**
A	398	10306.75	10038.62	**-2.6%**	10143	9838	**-3.0%**	1
B	303	16209.25	15258.5	**-5.9%**	14741	14020	**-4.9%**	0
C	695	33555.25	31654.25	**-5.7%**	29382	28110	**-4.3%**	0
D	935	53712.0	51112.95	**-4.8%**	35999	34452	**-4.3%**	0
E	699	27469.0	23006.75	**-16.2%**	14818	12882	**-13.1%**	0
Total (**avg**)	3030	141252.3	131071.1	**-7.2%**	105083	99302	**-5.5%**	1

Overall, the model is able to improve the objective value by 7.2%, but with significant differences among the warehouses. In warehouse A, customers have a very low geographical density, hence the improvement is the lowest both in overall objective value and in traveled distance. On the other hand, depot E has the biggest share of stochastic customers being assigned to routes by their postal code prefix, which explains the much larger margin for improvement by the model. On most warehouses, the proposed transshipments were already implemented, and, in some cases, the suggestions had already been tested in practice and removed due to operational difficulties. The only new transshipment proposed was in depot A.

As this optimization stage served the purpose of defining the new route plan for the company, the final results come from the simulation of the proposed plans under real historical data.

5.4.4 Simulation results

The routes defined by the conVRP-SLA were tested on how they would perform under real operational conditions. The different KPIs were then evaluated using the same dashboard developed during the data analysis stage of the case study. Note that, from the routes previously defined, only the internal customers are kept in the route plans during this simulation stage as assigning external customers to a fixed route is currently operationally impossible.

Results are simulated for three different configurations in each warehouse. Firstly, the routes of the current plan are simulated with the same order that was actually performed by the company. Then, the same plan is used but, for each route and day, a TSPTW instance is created and solved to assure the optimum path is used. Finally, the routes proposed by the conVRP-SLA model are simulated using the TSPTW model as well. In this way, it is possible to report two distinct results (Table 5.6), namely the ones obtained by optimizing the sequence in which routes are performed (Δ*sequenced*) and the ones that result from changing the assignment of customer to routes while also optimizing their sequencing (Δ*assignment*).

Table 5.6: Simulation results for the different company's warehouses

Depot	**Simulation type**	**Assignment changes**	**Routes**	**Total duration [h]**	**Total distance [km]**	**Estimated cost [€]**
A	Δ*sequenced*	n/a	n/a	-9.4%	-1.2%	-1.1%
	Δ*assignment*	11	0	-2.3%	-4.0%	-5.8%
B	Δ*sequenced*	n/a	n/a	-8.0%	-9.4%	-11.2%
	Δ*assignment*	9	0	-4.8%	-17.1%	-22.9%
C	Δ*sequenced*	n/a	n/a	-8.8%	-7.9%	-8.0%
	Δ*assignment*	100	-1	-5.9%	-17.5%	-14.5%
D	Δ*sequenced*	n/a	n/a	-4.4%	-9.5%	-7.9%
	Δ*assignment*	100	-2	-9.1%	-16.0%	-11.8%
E	Δ*sequenced*	n/a	n/a	-16.7%	-19.1%	-9.9%
	Δ*assignment*	48	0	-13.2%	-25.6%	-12.7%
Total	Δ*sequenced*	**n/a**	**n/a**	**-8.5%**	**-10.5%**	**-9.9%**
	Δ*assignment*	**268**	**-3**	**-8.4%**	**-17.4%**	**-12.7%**

The simulation showed improvements on all the warehouses. As it was expected, these results come from different sources, and the different configurations help in identifying the major factors explaining them. In depot A, the major improvements mainly come from the proposed transshipment as there were few assignment changes. This transshipment reduces the traveled distance significantly despite increasing the total duration when comparing to the sequenced simulation, as the transfer takes approximately half an hour to be performed. Warehouse B also has very few nodes being reassigned, but has the biggest estimated cost reduction due to having many external customers being delivered by better suited routes. Both depots C and D had many nodes changing routes, and this is the main cause for the improvements along with the reduction of the total number of routes. Warehouse E is very different from the others due to its large share of external customers. As it was claimed in the route optimization stage, there is a large margin for improvement in reassigning these customers that are not present in the robust routes plan, and the simulation stage further reiterates this statement. This depot also has by far the biggest improvements due to having a better route sequencing, which was expectable since many customers are stochastic, making the drivers' decision of which sequence to follow much harder.

Overall, there was an 8.5% improvement in route duration almost exclusively from performing a better route sequencing. On the other hand, the total reduction of traveled distance was evaluated in 17.4%, with 10.5% coming from better sequencing and the remainder from the proposed transshipment and customer assignment changes. All the proposed changes have an estimated impact on the influenceable costs of operations of 12.7%, with a very large share of 9.9% coming just from route sequencing and the remaining cost reductions coming from a better robust routes plan and operational assignment of external customers. The route sequencing was one of the factors being analyzed, but it was not expected to represent such a large portion of the total improvements. While this stresses the potential benefits of having the route sequence optimized before the route departure, there may be cases in which a non-optimal route is performed in order to serve more important customers earlier. Nevertheless, the large savings margin identified should prompt the company to look into a way to operationally optimize the sequencing. Also, despite the potential benefits from redefining the company's route plan, the current robust plan appears to be suitable to the overall business conditions the company operates in. However, the stochastic nature of warehouse E's customers indicates that there would be major benefits in applying a more dynamic route planning process in which routes would be defined after these external customers have placed their orders. As these orders usually arrive to the depot in the beginning of the day, it appears to be possible to successfully implement such a system, which would have the disadvantage of losing the driver consistency and the ability to have a pre-defined driver schedule.

Chapter 6

Conclusions and future work

This dissertation focused on further exploring the conVRP, which is an increasingly popular problem both in literature and in practice as many businesses are shifting to a more service level focused planning. An extension to this problem, the conVRP-SLA, was proposed, which considers different service level agreements that are very common on the pharmaceutical or spare parts distribution industry. Furthermore, in order for the model to be applicable to larger instances, and thus suitable for real situations, a matheuristic based on a FO approach was developed to tackle this problem.

In order to test the model, a real case study was studied in a Portuguese pharmaceutical distribution company that serves pharmacies several times a day and also other external customers with different service level agreements. Firstly, the current company situation was diagnosed by defining suitable KPIs and creating a dashboard that allowed for interactively exploring the operational costs and performance of the past results. Then, the developed mathematical model and matheuristic were used to plan the robust routes of the company by using historical data. Finally, in order to test the proposed routes and to analyze how different scenarios might impact the operations of the company, a simulation model was developed that, given a robust routes plan, reports the estimated KPIs in a similar interactive dashboard.

The route optimization stage, which used the proposed matheuristic, found a new set of routes that improved the overall objective value by 7.2% and reduced the total traveled distance by 5.5% by assigning both types of customers to new routes. However, the operational conditions require a robust pre-defined schedule, so, in order to validate the proposed robust routes plan, the simulation model was used to test their performance under different historical data. In addition, simulations were run with the current routes so as to optimize the sequencing and thus evaluate whether the drivers are performing them in an efficient way.

The simulation stage showed that a better sequencing of the routes could reduce the total duration of the routes by 8.5% and the traveled distance by 10.5%, which in the studied company had an estimated impact on the considered costs of 9.9%. The routes proposed by the optimization stage, which consisted of several assignment changes, one new transshipment and the elimination of 3 routes, had little impact on the total duration after the optimized sequencing, but further improved the traveled distance reduction to a total of 17.4%. Overall, all the proposed changes had an estimated impact of 12.7% on the cost considered to be influenceable by implementing better planning processes.

The results of the several simulations showed different possible areas of improvement for the case study company. Firstly, optimizing the route sequencing has large potential savings, even if some detours may currently be done to serve priority customers first. Furthermore, the model was able to find a robust routes plan that outperformed the current one developed with commercial routing software. The operational assignment of external customers, which cannot currently be included in the route plan, to the route with the most compatible node also outperformed the current postal code prefix assignment, especially in the warehouse in which these customers represent a larger share of the total customer pool. In fact, while robust routes seem suitable for the remaining company's warehouses, the large share of stochastic customers in this depot indicates that flexible route planning could be more suitable here if the company is willing to sacrifice the driver consistency and pre-defined schedules benefits. Regarding the developed conVRP-SLA model and matheuristic, these results showed that they are suitable to be applied as a planning tool in distribution companies with matching operational conditions. Furthermore, as the optimization stage also showed improvements on the considered historical data, this solution method is applicable not only to long term robust planning, but also in more dynamic settings in which consistent routes are still desired.

In order to further develop the proposed methodology, a better initial solution construction procedure could be developed in order for the model to be able to plan robust routes from scratch in a shorter period of time. Furthermore, an interesting additional analysis would be to compare the performance of the proposed matheuristic to different solution methods, namely metaheuristics which sacrifice the use of the mathematical model and hence the flexibility to easily integrate additional operational constraints and local optimality to provide in order to allow for a faster exploration of the solution space. Several approaches have been proposed in the literature such as a template based ALNS (Kovacs et al., 2014b), in which the different service level agreements considered in the developed conVRP-SLA model could be implemented.

Bibliography

Baldacci, R., Bartolini, E., Mingozzi, a., and Valletta, a. (2011). An Exact Algorithm for the Period Routing Problem. *Operations Research*, 59(1):228–241.

Baldacci, R., Mingozzi, A., and Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6.

Cacchiani, V., Hemmelmayr, V. C., and Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, 163(PART 1):53–64.

Campbell, A. M. and Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, 63(1):2–15.

Christofides, N. and Beasley, J. E. (1984). The period routing problem. *Networks*, 14(2):237–256.

Clarke, G. and Wright, J. W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points.

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

Cordeau, J. F., Laporte, G., and Mercier, a. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *The Journal of the Operational Research Society*, 52(8):928–936.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

Dondo, R. and Cerdá, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 176:1478–1507.

Gaudioso, M. and Paletta, G. (1992). A Heuristic for the Periodic Vehicle Routing Problem. *Transportation Science*, 26(2):86–92.

Groer, C., Golden, B., and Wasil, E. (2009). The Consistent Vehicle Routing Problem. *Manufacturing & Service Operations Management*, 11(4):630–643.

Gutin, G. and Punnen, A. P. (2002). *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467.

Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, pages 1–22.

Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, 195(3):791–802.

Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers and Operations Research*, 35(7):2307–2330.

Kovacs, A. A., Golden, B. L., Hartl, R. F., and Parragh, S. N. (2014a). The Generalized Consistent Vehicle Routing Problem. *Transportation Science*, 0(0):–.

Kovacs, A. A., Parragh, S. N., and Hartl, R. F. (2014b). A template-based adaptive large neighborhood search for the consistent vehicle routing problem. *Networks*, 63(1):60–81.

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

Li, F., Golden, B., and Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165 – 1179.

Pirkwieser, S. and Raidl, G. R. (2009). A column generation approach for the periodic vehicle routing problem with time windows. *Proceedings of the International Network Optimization Conference*, 2009:1–6.

Pisinger, D. and Ropke, S. (2007). A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34(8):2403–2435.

Ridder, R. D. (2014). ConRTR , a Consistent Vehicle Routing Problem algorithm.

Sungur, I., Ren, Y., Ordonez, F., Dessouky, M., and Zhong, H. (2010). A Model and Algorithm for the Courier Delivery Problem with Uncertainty. *Transportation Science*, 44(2):193–205.

Taillard, E., Badeau, P., Gendreau, M., Guertin, F., and Potvin, J.-Y. (1997). A Tabu Search Heuristic for the Vehicle Routing Problem with Soft time Windows.

Tarantilis, C. D., Stavropoulou, F., and Repoussis, P. P. (2012). A template-based Tabu Search algorithm for the Consistent Vehicle Routing Problem. *Expert Systems with Applications*, 39(4):4233–4239.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., and Rei, W. (2012). A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3):611–624.

Öncan, T., Altınel, İ. K., and Laporte, G. (2009). A comparative analysis of several asymmetric traveling salesman problem formulations. *Computers & Operations Research*, 36(3):637–654.