LUCAS BURAHEM MARTINS

ALGORITHMS FOR THE TIME WINDOW ASSIGNMENT VEHICLE ROUTING PROBLEM

LAVRAS – MG

2020

LUCAS BURAHEM MARTINS

ALGORITHMS FOR THE TIME WINDOW ASSIGNMENT VEHICLE ROUTING PROBLEM

Manuscript presented as phase of Defense Exam of the Graduate Program in Computer Science, to obtain the Master's degree.

Prof. DSc. Mayron César de Oliveira Moreira
Advisor

Prof. DSc. Manuel Iori
Co-advisor

LAVRAS – MG
2020

Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Martins, Lucas Burahem.
Algorithms for the time window assignment vehicle routing problem / Lucas Burahem Martins. – 2020.
62 p. : il.

Dissertação(mestrado acadêmico) –Universidade Federal de Lavras, 2020.
Orientador: Prof. DSc. Mayron César de Oliveira Moreira.
Bibliografia.

1. Operational Research. 2. Vehicle Routing Problem. 3. Assignment Time Windows. I. Moreira, Mayron César de Oliveira. II. Iori, Manuel. III. Título.

LUCAS BURAHEM MARTINS

ALGORITHMS FOR THE TIME WINDOW ASSIGNMENT VEHICLE ROUTING PROBLEM

Manuscript presented as phase of Defense Exam of the Graduate Program in Computer Science, to obtain the Master's degree.

APPROVED on August 24, 2020.

Prof. DSc. Dilson Lucas Pereira	UFLA
Prof. DSc. Geraldo Robson Mateus	UFMG
Prof. DSc. Manuel Iori	UNIMORE - Italy

Prof. DSc. Mayron César de Oliveira Moreira
Advisor

Prof. DSc. Manuel Iori
Co-advisor

LAVRAS – MG
2020

ACKNOWLEDGEMENTS

The time has come to conclude this journey. It was never easy, but friendships and guidance always take us far. First of all, I would like to thank God for the opportunity given to me. Thank you for taking care of everyone around me during that time. I am very grateful to my family and my girlfriend for their unconditional support all the time. You're everything to me. I thank my advisor Mayron Moreira for reaching out to me during the most difficult moment of this journey, for guiding me, for teaching me what is right and wrong. Without you, none of this would be possible. I am grateful to my co-advisor Manuel Iori and scientific tutor Giorgio Zucchi for providing me with the greatest opportunity of my life so far. You were responsible for my professional growth through your teachings and guidelines. My great friends at UFLA, what would this journey be like without you? I will never forget the moments of leisure and fun. You taught me the importance of friendships in our lives. Thanks for the support, the joys, and the teachings. I would like to thank Italy and the Italians for having received and welcomed me in such a difficult time for humanity. Italy is beautiful and the Italians are incredible. To DCC staff and members, thank you for always offering us the best of you. If DCC produces quality work, a large part of this is due to the cleaning team that always takes care of the facilities in an impeccable way, the secretariat always supporting us with patience and dedication, to the professors who advise and support us with knowledge, the students who strive for the quality of projects and UFLA that offers us the best campus structures. I would like to thank the members of the master's defense panel Dilson Pereira, Manuel Iori, Geraldo Robson, Franklina Toledo, André Pimenta, and Andreza Beezão Moreira for accepting to be part of this process together with us. Finally, I thank myself for being strong in difficult times and for all the effort to produce quality work.

The authors would like to thank the support of Coopservice S.coop.p.a. (Italy), UNIMORE (Italy) under FAR 2018 grant, and the National Council for Scientific and Technological Development (CNPq, Brazil).

ABSTRACT

We study the Time Window Assignment Vehicle Routing Problem (TWAVRP), which appears in real contexts where we can see, for example, unknown demand, fluctuations per delivery, and multi-periods. We consider two problems, that share as the main characteristics of capacitated vehicles, and exogeneous time windows for each client. The first problem is a stochastic approach, we deal with a set of scenarios, and a multi-period variant. Our goal is to minimize transportation costs and to assign endogeneous time windows overall scenarios, for variant 1, and for all periods, for variant 2. We propose a hybrid algorithm for both problems, that generates a set of routes by requesting an Iterated Local Search (ILS) metaheuristic and then chooses the most appropriate routes through a set-covering based auxiliary formulation. The contributions described here are threefold. First, we improve the best-known solutions reported to the stochastic TWAVRP, proposed in the literature. Then we test an approach for the multi-period TWAVRP by adding heterogeneous vehicles and driver stopping periods assumptions. This variant appears in the pharmaceutical industry. With a database provided by the Coopservice company, TWAVRP has been adjusted to handle real instances. Finally, we test our approach with such an instance. Computational results indicate that the proposed algorithm is accurate in practice, obtained good solutions for both artificial and real instances. For instances that have more than 45 customers, our method outperforms the results found in the literature. In the end, we were able to answers our research question: "What are the algorithms that can optimize costs and respect all constraints of TWAVRP and its variant concerning the Coopservice routing planning?"

Keywords: Vehicle Routing Problem, Time Window Assignment, Pharmaceutical industry, Coopservice Company, Hybrid algorithm.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Vehicle routing is a class of problems that appears in several combinatorial optimization studies due to their practical relevance, mainly in the areas of retail and transport (TOTH; VIGO, 2014). The classical *Vehicle Routing Problem* (VRP) calls for shipping freight to customers located along a distribution network using a fleet of vehicles, to minimize the delivery costs.

Inspired by retail distribution networks, Spliet and Desaulniers (2015) introduced the *Time Window Assignment Vehicle Routing Problem* (TWAVRP). The TWAVRP appears when the quantity of demands of the customers are uncertain, and time windows should be allocated to the customers located at the distribution network, to minimize the expected travel costs. In the TWAVRP, each endogenous time window with a fixed-width, must be associated with the exogenous time window of the client. The exogenous time windows are represented by the arrival and departure limits of a customer. According to Neves-Moreira et al. (2018), the TWAVRP can be defined as a *two-stage stochastic optimization problem*. The first stage decisions are to assign a set of time windows to customers before demand is known. In the second stage, after requests are revealed for each day, delivery schedules respecting the assigned time windows must be designed.

The TWAVRP appears in several segments of the world economy, such as the pharmaceutical industry. The pharmaceutical industry has stood out in the world economy for its great importance to public health. According to the World Health Organization (WHO), the global pharmaceutical market is worth US$ 300 billion a year, and this number is going up to US$ 400 billion in the next three years. The Italy health system's guidelines are for the centralization of drug distribution to hospitals and health structures. Medicines need safety and efficient transport due to the high demand from hospitals and pharmacies. This fact explains the increasing amount of investments in logistics to improve the effectiveness of transportation.

Coopservice is one of the most prominent Italian companies in the design, supply, and management of integrated services to businesses and communities. One of these services is pharmaceutical logistics. The benefits for institutions are the reduction of business costs and staff dedicated to the purchasing function. The company is responsible for delivering medication to several hospitals in order to fulfill their demands. For that, Coopservice has the support of a robust fleet of over 300 vehicles of three different types, a nationwide network of warehouses, and about one thousand coordinated operators (COOPSERVICE, 2018). Each vehicle, with limited cargo capacity, leaves a depot and must visit a set of hospitals. Technicians should be at the delivery place when the vehicle arrives to take care of the shipment. Each hospital has a particular time window that must be respected by the vehicle. Moreover, each hospital must order fewer products than the total capacity of the vehicle. Over a planning horizon (i.e., weekly, monthly), the challenge is to build a schedule subject to technical constraints, minimizing costs and maximizing time window robustness.

One of the fundamental problems underlying the Coopservice context is dealing with a TWAVRP variant with time windows, with policies and routing regulatory limitations for drivers. In this way, the contributions of this study are:

1. Improvement of the best upper bounds reported by Dalmeijer and Spliet (2018) for the stochastic TWAVRP;
2. Formal definition of the Coopservice problem which consists of a variant of the multi-period TWAVRP to support heterogeneous vehicles and stop points to the drivers;
3. Development of a hybrid heuristic to efficiently solve both problems;
4. Use of real data of the company to generate new instances.

This project's general objective is to develop algorithms to efficiently solve variants of a vehicle routing problem inspired by the Coopservice pharmaceutical context. Time windows requirement is essential since hospital staff should be at the delivery place when the vehicle arrives to take care of the medications and these time windows should be allocated to the customers to minimize the travel costs.

This manuscript is structured as follows. Chapter 2 presents a literature review concerning recent works on VRP and their variations related to the studied problem. Chapter 3 describes formally the TWAVRP variants, and presents their mathematical models. Chapter 4 presents the methodology to solve the problems we take. The method proposed is evaluated on Chapter 5 and the conclusions are reported in Chapter 6.

2 LITERATURE REVIEW

Due to academic interest in VRP variations, researchers have focused on more realistic VRPs, named Rich Vehicle Routing Problem (RVRP). The problem we describe in Chapter 3 has characteristics that resemble the Vehicle Routing Problem with Time Windows (VRPTW) (DESROCHERS; DESROSIERS; SOLOMON, 1992), the Vehicle Routing Problem with Flexible Time Windows (VRPFlexTW) (TAŞ; JABALI; Van Woensel, 2014), and the Time Window Assignment Vehicle Routing Problem (DALMEIJER; SPLIET, 2018). Since one of our contributions is to solve a real instance of the Coopservice company that appears in the context of the Pharmaceutical industry, we also present a set of studies related to the Pharmaceutical Vehicle Routing Problem (Pharmaceutical VRP) (CAMPELO et al., 2019).

Most of the problems described in the subsequent sections share the following characteristics:

- Each customer must belong to exactly one route that starts and end on a depot;
- Each vehicle has a known maximum capacity that must not be exceeded;
- Service days and corresponding demands of each customer are known in advance;
- Each customer is associated with a non-negative demand and service duration;

This chapter presents the state-of-art of these problems. The following subsections describe each of the variants mentioned. Table 2.1 and 2.2 summarizes the general VRP attributes and time windows characteristics presented in the papers described in this chapter, respectively.

2.1 Vehicle Routing Problem with Time Windows

The vehicle routing problem with time windows (VRPTW) is a generalization of the VRP involving appropriate time intervals for performing services, called time windows. In these problems, customer service can only be started within a time window defined by the client (DESROCHERS; DESROSIERS; SOLOMON, 1992).

Desrochers, Desrosiers and Solomon (1992) develop a new optimization algorithm to solve the VRPTW. The authors used a column generation and a branch-and-bound scheme to optimize the solution. They also performed computational experiments in 3 sets of benchmarks found in the literature (Solomon's well-known set of instances) to test the effectiveness of the proposed algorithm. The results show an optimality gap of 1.5%, on average, over 27 test problems.

Cordeau, Laporte and Mercier (2001) propose a unified Tabu Search Algorithm to solve the VRPTW with two variations: Periodic and Multi-Depot VRPTW. After the generation of a starting solution, a specialized heuristic for the Traveling Salesman Problem with Time Windows builds several solutions and selects the best one for a post-optimization phase. Computational performance is tested on Solomon's instances and results compared with other heuristics from literature. The experiments show that the gap between the best-known solution for the instances and the result obtained by applying the proposed Tabu Search is below 1.5%.

Ombuki, Ross and Hanshar (2006) study the VRPTW as a multi-objective optimization problem (MOP) to minimize the total cost of routing and the number of vehicles used without violating the capacity of the vehicles and time windows. The authors propose the use of a Genetic Algorithm in which the two dimensions of the problem are considered separated in the multi-objective search space using the Pareto Ranking procedure. The experimental results are obtained through the use of the standard Solomon's VRPTW benchmark problem instances. The results show a better average number of vehicles compared to some of the well known published GA based methods for many instances. The proposed GA was competitive compared to other well-known works of the literature, obtaining better results for some instances.

Azi, Gendreau and Potvin (2010) tackle a VRPTW variant where a vehicle can perform several routes during a day. The authors use a branch-and-price approach using column generation in which lower bounds are computed by solving an LP relaxation in two phases: generating non-dominated feasible routes and selecting some of these routes to form the vehicle workday. The experiments are computed on Solomon instances benchmarks for 100 euclidean customers. The algorithm was only tested on instances of problems with 25 to 50 customers.

Ceschia, Gaspero and Schaerf (2011) formalize a Tabu Search algorithm with a combination of neighborhoods to solve a VRPTW with a mixed fleet. The neighborhood relations are defined by a set of possible moves that depend on the day, the vehicle, and the route's position. The outsourcing of part of the transportation to external carriers is allowed with a complex cost for the usage of the vehicle that is limited by the capacity and the travel distance. Also, priorities for customer demands can be set at a particular value. For this, the orders distinguish between mandatory orders and free ones. The author performs experimental analysis on public benchmarks to compare previous works on similar problems and apply the proposed algorithm on a real-case data set on a technological company in Italy.

Amorim et al. (2014) consider a VRP in the context of food delivery. The authors model the problem as a Rich Vehicle Routing Problem (RVRP), where a heterogeneous fleet of refrigerated trucks must deliver food types classified as dry, cold, and frozen. Each customer has several hard time windows during the day to be serviced. The authors propose an Adaptative Large Neighbourhood Search (ALNS). A food delivery company in Portugal provided the data set used on the computational experiments. The two tested instances have 350 and 336 customers. The results show that the use of the proposed algorithm reduced the company's vehicle costs by almost 20%.

A recent study proposed by Keskin and Çatay (2018) formulated the electric vehicle recharging problem as a VRPTW. The problem is formulated as a mixed-integer linear program. The CPLEX solver obtained the optimal solutions for small instances. A matheuristic based on Adaptive Large Neighborhood Search (ALNS) with an exact method was applied to larger instances.

Ferreira et al. (2018) propose a Variable Neighbourhood Search (VNS) to solve a VRP with Multiple Hard Time Windows. In this problem, each customer has one or more time windows in which they can be visited. The author compares the results of the proposed VNS heuristic to a Hybrid Variable Neighborhood Tabu Search (HVNTS) heuristic over 72 instances, divided into two sets such that the second one presents time window overlap. According to the author, the proposed VNS is competitive since it obtained a relative average deviation of 0.02% on 72 instances proposed by Solomon to the state-of-the-art.

Recently, Fachini and Armentano (2020) developed an approach applying Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. The algorithm reached optimal solutions of instances with up to 100 customers, becoming competitive with other state-of-the-art methods. For a more extensive and complete state-of-the-art concerning this variant, we suggest the book chapter about the VRP with Hard Time Windows (VRPHTW) edited by Toth and Vigo (2014).

2.1.1 Vehicle Routing Problem with Flexible Time Windows

The VRP with Flexible Time Windows (VRPFlexTW) allows a maximum limit of time window violation. A specific penalty is paid for services that start before or after the maximum violation of time windows. This relaxation allows the optimization of the operational cost of procedures since customers can be served before and after the defined interval.

Figliozzi (2010) applies an Iterative Route Construction and Improvement (IRCI) algorithm to the VRP with Soft Time Windows (VRPSTW(). The IRCI uses a bottom-up approach with four components: (*i*) a generalized nearest neighbor heuristic; (*ii*) a sequential constructive algorithm that generates feasible routes with least cost; (*iii*) a route improvement algorithm that can reduce the routing costs; and (*iv*) a start time improvement algorithm that eliminates the violation of some time windows. Computational experiments compare the proposed IRCI to other solution methods reported in the literature. They used 56 Solomon's benchmark problems with 100 customers.

TaŞ et al. (2013) consider the VRPSTW with stochastic travel times. Such characteristic plays a role in the calculations of transportation and services costs. The authors develop a mathematical formulation aiming at constructing a set of routes minimizing the total cost that involves penalties for early or late servicing, transportation, and services costs. The solution method is developed in three phases. The first phase builds a feasible solution by using an initialization algorithm. Then, this solution is improved by a Tabu Search method concerning the total weighted cost. In the last phase, a post-optimization method is applied to the generated solution. The authors took Solomon's problem instances for 100 customers to show the effectiveness of the algorithm. The tests showed that the proposed method's usage was able to improve the best-known solution by 3% on average.

TaŞ, Jabali and Van Woensel (2014) develop a solution procedure to the VRPFlexTW using an adaptation of the Tabu Search algorithm. A linear programming model was proposed to handle the detailed scheduling of customer visits for given routes. Computational experiments considered adaptations of Solomon's well-known set of instances. Also, the authors highlight the advantages of the VRPFlexTW when compared to the VRPTW.

Beheshti, Hejazi and Alinaghian (2015) purpose a vehicle routing problem in which the customer has multiple soft time windows, and the demand must be served at an appropriate trade-off between the customer satisfaction and the cost distribution. This variant has a set of non-overlapping time windows where the distributor must prioritize one of them. The authors developed a mathematical model and proposed a Co-evolutionary Multi-objective Quantum-Genetic Algorithm (CCMQCA) to solve the problem. This approach decomposes the problem into two modules. First, we compute the customers' sequence. Then, we evaluate the number of customers in each vehicle. The proposed algorithm was compared to the current solution obtained by a Non-dominated Sorting Genetic Algorithm (NSGA-II) and a Multi-objective Quantum-inspired Evolutionary Algorithm (MQEA). The solutions for 110 customers show that the CCMQCA obtained better solutions than the other algorithms.

Fachini and Armentano (2017) propose a dynamic programming algorithm to solve the Traveling Salesman Problem With Flexible Time Windows (TSPFlexTW). In this case, the Flexible time windows correspond to increasing the upper and lower limits of the time windows. Linear penalty costs and a cost per unit time of anticipation and delay are applied to the objective function in cases where the new limits of the time windows are not respected. The authors use a dynamic programming algorithm with a label extension mechanism to solve the problem. Experiments evaluate the proposed algorithm in a set of 135 instances with several customers ranging from 20 to 200. The results show that the method was able to achieve 70% of optimal solutions.

Fachini and Armentano (2018) solve the traveling salesman problem with flexible time windows with a dynamic programming method based on label-correcting strategies to minimize travel costs with the use of flexible time windows. The algorithm was tested in two sets of symmetric and asymmetric instances. For the asymmetric instances, the proposed algorithm solved 198 of 203 instances with a gap of approximately 0.20% compared to the optimal solutions. Evaluating the use of Flexible Time Windows, the author shows that in comparison to TSP with Hard Time Windows, there was an average cost savings of 17.22 %.

2.1.2 Time Windows Assignment Vehicle Routing Problem

When the customer demands is uncertain, time windows should be allocated to the distribution network in such a way that expected travel costs are minimized. This problem is called the Time Window Assignment Vehicle Routing Problem (TWAVRP). In the TWAVRP, a fixed-width endogenous time window is associated with an exogenous time window with arrival and departure limits. According to Neves-Moreira et al. (2018), TWAVRP can be defined as a two-stage stochastic optimization problem. Given a set of customers where each one must be visited within a regular period, the first stage decisions are to assign a set of time windows to each customer, before demand is known. In the second stage, after the demand is revealed for each day, we establish delivery schedules respecting the assigned time windows.

Spliet and Gabor (2015) introduces the TWAVRP. They consider a finite number of scenarios with their probability of occurring where the demand of each client is uncertain. In this case, the uncertaint comes from the fact that each scenario is characterized by different combinations of demands for each client. The proposed model is solved by a Column Generation algorithm incorporated in an exact branch-cut-and-price algorithm. Computational experiments with 40 instances with 10, 15, 20, and 25 customers show that the proposed branch-cut-and-price algorithm solves the TWAVRP above 25 customers in 3 scenarios.

Spliet and Desaulniers (2015) tackle a TWAVRP where where each client has a predefined time window represented by a start and a departure time (exogenous time windows). A fixed width endogenous time window must be associated with each customer's exogenous time windows in order to increase their robustness. They consider a finite number of scenarios with their probability of occurring. To solve the problem, the authors implement a branch-cut-and-price algorithm and a Tabu Search based on column generation. New test instances are generated using a uniform distribution. The authors show that the results obtained considering five scenarios allow an average cost reduction of 3.64% compared to a single-scenario approach.

Dalmeijer and Spliet (2018) addresses the TWAVRP through a Branch-and-Cut algorithm. The author introduces a branching strategy based on a set of valid and precedence inequalities. The algorithm's effectiveness is demonstrated through numerical experiments and comparisons with a proposal of a branch-cut-and-price algorithm (SPLIET; GABOR, 2015).

Inspired by a large European food retailer, Neves-Moreira et al. (2018) apply the TWAVRP to a real food distribution with a fleet composed of around 200 customers, with time windows defined according to the product segments. This problem considers both the traveled distance and fleet requirements costs. Solution method uses three phases: (*i*) Route Generation, (*ii*) Initial Solution Construction and (*iii*) Improvement by a Matheuristic.

Spliet, Dabia and Woensel (2018) propose a mathematical formulation for the TWAVRP, which includes time-dependent travel times. To deal with this new variation, the authors apply a branch-and-price-cut algorithm to minimize transport costs. Computational tests were run on artificial instances that contain four sets of 10 instances with 10, 15, 20, and 25 customers. The best solution found is 0.55% higher than the optimum solution, with a gap of 2.40% regarding LP relaxation.

2.2 Pharmaceutical Vehicle Routing Problem

In recent years, the pharmaceutical industry has gone through various changes, mainly due to the aging of the population and the increase in service costs (MAGALHÃES; SOUSA, 2006). The Pharmaceutical Vehicle Routing Problem addresses the application of the VRP and its variations to optimize medication delivery systems in the pharmaceutical industry. According to Campelo et al. (2019), this context is defined by three characteristics:

- Low margins for distribution players;
- Highly regulated market;
- Well-established competitive environment;

Magalhães and Sousa (2006) deal with a Dynamic Vehicle Routing Problem in a case study of pharmaceutical goods distribution operating in the North and Center of Portugal. A four-phase dynamic algorithm is proposed, where: (*i*) a cluster of the orders is determined; (*ii*) potential routes are constructed; (*iii*) a route is selected for operation, and (*iv*) such route is subject to an improvement process. The algorithm was evaluated the company's operation team during one week, and results showed accurate and fast compared with results obtained by the manual procedure.

Liu et al. (2013) consider a case study in France provided by Home Health Care (HHC) companies, where the customers and the hospitals demand various pickup and delivery services. It concerns the delivery of drugs from companies to patient's homes, drug delivery from hospital to patients, and pickups of waste and biological material from patient's homes to hospitals. A mathematical model and two other approaches are proposed to solve the problem: Genetic Algorithm (GA) and Tabu Search. These algorithms are compared with the mathematical model solution using the CPLEX solver. The database test is generated based on 100 customers of eighteen Solomon's VRPTW instances. For the 132 instances tested, the GA deviated on average by 16.4% compared with the CPLEX result, and the Tabu Search deviated on average by 17.0%. The CPU time of the Tabu Search is, on average, 9.28% seconds longer than the GA.

Similar to the pickup and delivery problem discussed earlier, Liu, Xie and Garaix (2014) extend the variation of the VRP by also treating the periodicity of the consumers. This problem was called Periodic Home Health Care Pickup and Delivery Problem (PHHPDP). To minimize the cost involved, the authors apply a hybridization of the Tabu Search algorithm. The results obtained are compared to state-of-the-art algorithms for the problem. From 5 real-life instances, the solution obtained by the proposed Tabu Search is shorter than the one used by the HHC Company by 10.08%, 16.59%, 15.73%, 8.41%, and 9.31%, respectively.

Ceselli, Righini and Tresoldi (2014) study the distribution of drugs or vaccines in case of emergency with mathematical programming techniques. The authors consider the option of reaching citizens to deliver drugs with a mixed fleet or by establishing distribution centers (DC), where clients must go to receive their treatments or drugs. The objective function aims to maximize the demand served within a deadline. A branch-cut-and-price algorithm was run over 440 instances with 10, 20, and 50 delivery sites with a limit of one hour, obtaining the optimal solution in almost all the instances.

Settanni, Harrington and Srai (2017) investigate how to assess the multifaceted aspects of the reconfiguration of the Pharmaceutical Supply Chain (PSC) due to new technological interventions in drug manufacturing and delivery models. From a survey, the authors demonstrate that the current PSC definitions fail to capture consumers due to the centralization of production and the inadequate conceptualization of the supply chain structure. These failures are because current works aim at maximizing efficiency and effectiveness against other potentially relevant aspects.

Martins et al. (2017) tackle the wholesalers network redesign problem by taking into account both operational costs and customer service level. This process is the so-called supply chain network redesign (SCNR) and aims to optimize the strategic-tactical redesign wholesaler's network decisions to evaluate the solutions obtained operationally. The response time of wholesalers and the availability of products are their competitive edges. The process is divided into two stages. First, the strategic-tactical decisions are obtained by solving a mixed-integer programming (MIP) model. On the second stage, the obtained solution is evaluated by a simulation model to assess the impact of the redesign of the wholesaler's network. The proposed model showed savings by 4%, on average, on a pharmaceutical wholesalers case-study.

Kramer, Cordeau and Iori (2018) present a RVRP based on a practical Italian pharmaceutical distribution problem. The main objective of the paper is to deliver pharmaceutical products to hospitals and health care facilities in Tuscany (Italy). The problem has characteristics of multiple depots, heterogeneous fleet, and flexible time windows. Depots are classified into principal and auxiliary. There is an option to supply auxiliary depots with products from the central depots. Customers are classified into small customers and hospitals. Hospitals accept the anticipation of demands by being equipped with warehouses. The authors developed an Iterated Local Search (ILS) with a constructive algorithm, local search procedures, and a perturbation phase combined with a multi-start (MS) method to further diversify the search. The performance of the proposed ILS algorithm is evaluated by solving real-life instances of pharmaceutical distribution, and by solving artificial instances. The proposed MS-ILS can provide feasible solutions that achieve improvements about 23% on average compared to the best solution found by the greedy heuristic. However, the results show gaps of 2% in relation to the best-known solution.

Campelo et al. (2019) propose a Fix-and-optimize (FO) approach using the mathematical model developed to a problem faced at a pharmaceutical distribution company operating mainly in Portugal. The authors tackle the Consistent Vehicle Routing Problem with multiple daily deliveries, different service level agreements, and release dates. The solution method has a node grouping step to reduce the number of nodes, an initial solution construction stage, and the Fix-and-Optimize algorithm.

Table 2.1 – Extensions of VRP found in the literature.

	Features						
Reference	M.D.	O.D.	Het. Fleet	Hom. Fleet	Periodic	Sol. Method	Case study
Amorim et al. (2014)		✓	✓		✓	LNS	Food distribution company
Azi, Gendreau and Potvin (2010)		✓		✓		Branch-and-Price	-
Beheshti, Hejazi and Alinaghian (2015)		✓		✓		CCMQCA	Juice distribution company
Campelo et al. (2019)		✓		✓	✓	Fix-and-Optimize	Pharmaceutical distribution
Ceselli, Righini and Tresoldi (2014)	✓		✓			Branch-Cut-and-Price	-
Ceschia, Gaspero and Schaerf (2011)		✓	✓		✓	Tabu Search	Technology company (Italy)
Cordeau, Laporte and Mercier (2001)	✓			✓		Tabu Search	-
Dalmeijer and Spliet (2018)		✓		✓	✓	Branch-and-cut	-
Desrochers, Desrosiers and Solomon (1992)		✓		✓		Branch-and-bound	-
Fachini and Armentano (2017)		✓		✓		Dynamic Programming	-
Feillet et al. (2014)		✓		✓	✓	VNS	-
Ferreira et al. (2018)		✓		✓		VNS	-
Figliozzi (2010)		✓		✓		IRCI	-
Groër, Golden and Wasil (2009)		✓		✓	✓	MIP	-
Keskin and Çatay (2018)		✓		✓		ALNS	Electric Vehicle Routing
Kovacs, Parragh and Hartl (2015)		✓		✓	✓	MDLNS	-
Kovacs et al. (2015)		✓		✓	✓	LNS	-
Kovacs, Parragh and Hartl (2014)		✓		✓	✓	TBLNS	-
Kramer, Cordeau and Iori (2018)	✓		✓		✓	ILS	Pharmaceutical distribution
Liu et al. (2013)		✓		✓		GA and Tabu Search	Pharmaceutical pick up and delivery
Liu, Xie and Garaix (2014)		✓		✓	✓	Tabu Search	Pharmaceutical pick up and delivery
Magalhães and Sousa (2006)		✓		✓	✓	-	Pharmaceutical pick up and delivery
Martins et al. (2017)	✓		✓		✓	MIP	Pharmaceutical distribution
Neves-Moreira et al. (2018)		✓	✓			Fix-and-Optimize	Product distribution
Ombuki, Ross and Hanshar (2006)		✓	✓			GA	-
Spliet and Desaulniers (2015)		✓		✓		Branch-cut-and-price	-
Spliet and Gabor (2015)		✓		✓	✓	Branch-cut-and-price	-
Spliet, Dabia and Woensel (2018)		✓		✓		Branch-cut-and-price	-
Stavropoulou, Repoussis and Tarantilis (2019)		✓		✓	✓	Adaptative Tabu Search	-
Tarantilis, Stavropoulou and Repoussis (2012)		✓		✓	✓	Tabu Search	-
TaŞ et al. (2013)		✓		✓		Tabu Search	-
TaŞ, Jabali and Van Woensel (2014)		✓		✓		Tabu Search	-
Our work		✓	✓		✓	ILS	Pharmaceutical distribution

Legend: M.D. - Mult-Depot | O.D. - One depot | Het. Fleet - Heterogeneous Fleet | Hom. Fleet - Homogenenous Fleet

Source: Authors.

Table 2.2 – Extensions of VRPTW variations found in the literature.

	Time Window Type			
Reference	VRPTW	VRPSTW	VRPFlexTW	TWAVRP
Azi, Gendreau and Potvin (2010)	✓			
Amorim et al. (2014)	✓			
Beheshti, Hejazi and Alinaghian (2015)			✓	
Campelo et al. (2019)		✓		✓
Ceschia, Gaspero and Schaerf (2011)	✓			
Cordeau, Laporte and Mercier (2001)	✓			
Dalmeijer and Spliet (2018)				✓
Desrochers, Desrosiers and Solomon (1992)	✓			
Fachini and Armentano (2017)			✓	
Ferreira et al. (2018)	✓			
Figliozzi (2010)		✓		
Keskin and Çatay (2018)	✓			
Kovacs, Parragh and Hartl (2015)	✓			
Kramer, Cordeau and Iori (2018)		✓		
Neves-Moreira et al. (2018)			✓	
Neves-Moreira et al. (2018)			✓	
Ombuki, Ross and Hanshar (2006)	✓			
Spliet and Gabor (2015)				✓
Spliet and Desaulniers (2015)				✓
Spliet, Dabia and Woensel (2018)				✓
Tarantilis, Stavropoulou and Repoussis (2012)	✓			
TaŞ et al. (2013)		✓		
TaŞ, Jabali and Van Woensel (2014)			✓	
Our work	✓			✓

Source: Authors.

3 MATHEMATICAL FORMULATIONS

This chapter presents notation (Section 3.1) inspired in Dalmeijer and Spliet (2018), and formally describes in Section 3.2 the stochastic *Time Windows Assignment Vehicle Routing Problem* (TWAVRP) according to Dalmeijer and Spliet (2018). For the variant of TWAVRP faced by the Coopservice company, we show a mathematical model on Section 3.3 based on the modeling proposed by Dalmeijer and Spliet (2018). In this way, we replicate the idea of scenarios by converting it into a planning horizon composed of a set of periods. For the Coopservice extension, we also consider two additional characteristics: heterogeneous fleet and regulatory routing limitations for drivers.

3.1 Notations

Consider a set of clients denoted by $H = \{1, 2, \ldots, n\}$. A complete and directed graph $G = (N, A)$ models the network of this problem, where $N = H \cup \{0, n+1\}$ is the overall set of nodes and 0 and $n+1$ represent, respectively, the departure and arrival depot nodes of all routes (we split the depot on departure and arrival in order to improve the understanding of the mathematical formulation). A set $A_H \subset A$ of arcs indicates the connections between any pair of different customer nodes $i, j \in H$. Similar to set N, we define $A = A_H \cup \{(0, j) \cup (j, n+1), \forall j \in H\}$, as the overall set of arcs connecting customers and depot nodes. Each arc $(i, j) \in A$ has an associated travel time t_{ij} and a travel cost c_{ij}. We assume that the travel times are non-negative and satisfy the triangle inequality.

Each client $j \in H$ has to be assigned to an endogenous time window of width w_j, which must be selected in a fixed exogenous time window $[e_j, l_j]$, where $l_j - e_j \geq w_j$. A time window $[e_0, l_0]$ represents the opening hours of the departure depot. Similarly, a time window $[e_{n+1}, l_{n+1}]$ represents the opening hours of the arrival depot. The objective function consists in minimizing the expected traveled cost overall scenarios.

3.2 TWAVRP mathematical formulation

Consider an unlimited set of homogeneous vehicles with capacity Q is available at the departure depot. To model demand uncertainty, we consider a set Ω of demand scenarios, each having probability of occurrence p_ω, for $\omega \in \Omega$, in such a way that $\sum_{\omega \in \Omega} p_\omega = 1$. Each customer $j \in H$ has a demand in scenario $\omega \in \Omega$ given by $0 \leq d_j^\omega \leq Q$. Thus, for each scenario customers have a combination of different demands.

Define y_j as a variable which measures the starting times of the endogenous time windows on the client $j \in H$. Thus, the endogenous time windows of client j is given by $[y_j, y_j + w_j]$ and $y_j \in [e_j, l_j - w_j]$. Let $x_{ij}^\omega \in \{0,1\}$ be a binary variable equal to 1 if arc $(i,j) \in A$ is traversed in scenario ω. Define f_j^ω as the time of client $j \in H$ receives delivery in scenario $\omega \in \Omega$.

The flow variable z_{ij}^ω for all $(i,j) \in A$, $\omega \in \Omega$, depends on the direction the arc is traversed (given by variable x). If $x_{ij}^\omega = 1$, variable z_{ij}^ω represents the total vehicle load when it traverses arc (i,j). If $x_{ji}^\omega = 1$ variable z_{ij}^ω represents the leftover capacity on the vehicle when traversed the arc (j,i). If $x_{ij}^\omega = x_{ji}^\omega = 0$, thus z_{ij}^ω is zero.

The mixed-integer linear programming proposed by Dalmeijer and Spliet (2018) reads:

$$\text{minimize} \sum_{\omega \in \Omega} p_\omega \sum_{(i,j) \in A} c_{ij} x_{ij}^\omega \tag{3.1}$$

subject to

$$\sum_{j \in H \cup \{n+1\}} x_{ij}^\omega = 1 \qquad \forall i \in H, \omega \in \Omega \tag{3.2}$$

$$\sum_{i \in H \cup \{0\}} x_{ij}^\omega = 1 \qquad \forall j \in H, \omega \in \Omega \tag{3.3}$$

$$z_{ij}^\omega + z_{ji}^\omega = (x_{ij}^\omega + x_{ji}^\omega) Q \qquad \forall (i,j) \in A, i < j, \omega \in \Omega \tag{3.4}$$

$$\sum_{j \in N} (z_{ij}^\omega - z_{ji}^\omega) = 2 d_i^\omega \qquad \forall i \in H, \omega \in \Omega \tag{3.5}$$

$$\sum_{j \in \mathcal{N}} z_{0j}^\omega = \sum_{i \in H} d_i^\omega \qquad \forall \omega \in \Omega \tag{3.6}$$

$$\sum_{j \in H} z_{n+1,j}^\omega = \left(\sum_{j \in H} x_{0j}^\omega \right) Q \qquad \forall \omega \in \Omega \tag{3.7}$$

$$\sum_{j \in H} z_{j0}^\omega = \left(\sum_{j \in H} x_{0j}^\omega \right) Q - \sum_{i \in H} d_i^\omega \qquad \forall \omega \in \Omega \tag{3.8}$$

$$f_j^\omega - f_i^\omega \geq t_{ij} x_{ij}^\omega + (l_j - e_i)(1 - x_{ij}^\omega) \qquad \forall i,j \in H, \omega \in \Omega \tag{3.9}$$

$$s_0 + t_{0j} \leq f_j^\omega \qquad \forall j \in H, \omega \in \Omega \tag{3.10}$$

$$f_i^{\omega} + t_{i,n+1} \leq l_{n+1} \qquad \forall i \in H, \omega \in \Omega \tag{3.11}$$

$$f_i^{\omega} \geq y_i \qquad \forall i \in H, \omega \in \Omega \tag{3.12}$$

$$f_i^{\omega} \leq y_i + w_i \qquad \forall i \in H, \omega \in \Omega \tag{3.13}$$

$$y_i \in [e_i, l_i - w_i] \qquad \forall i \in H \tag{3.14}$$

$$x_{ij}^{\omega} \in \{0,1\} \qquad \forall (i,j) \in A, \forall \omega \in \Omega \tag{3.15}$$

$$z_{ij}^{\omega} \geq 0 \qquad \forall (i,j) \in A, \forall \omega \in \Omega \tag{3.16}$$

The objective function (3.1) minimizes the traveled costs for all scenarios. Constraints (3.2) and (3.3) establish the flow conservation. Each customer must have an entry arch traversed by a vehicle and an exit arc traversed by the same vehicle. Constraints (3.4) indicate that when opposing arcs (i,j) and (j,i) are used, the sum of z_{ij}^{ω} and z_{ji}^{ω} must be equal to the vehicle capacity. Constraints (3.5) establish the flow conservation for the z-variables. The d variable works as follows: before visiting the customer i, the vehicle load is d_j^{ω} units more than after. After visiting the customer i, the empty space is d_i^{ω} more units than before. The total difference in both flows is equal to $2d_i^{\omega}$. When opposing arcs (i,j) and (j,i) are used, the sum of the differences of z_{ij}^{ω} and z_{ji}^{ω} for each customer $j \in N$ must be equal to $2d_i^{\omega}$. Constraints (3.6) indicate that the total load of a vehicle recorded in the departure deposit must equal the total demand of all clients visited by him. Constraints (3.7) the total capacity considering all vehicles used for routing must be equal to the number of vehicles used multiplied by the homogeneous capacity Q. Constraints (3.8) set the total excess capacity of all used vehicles. Constraints (3.9) are the MTZ-inequalities that model the service time. If $x_{ij} = 1$, the vehicle travels from i to j in scenario ω, then $f_j^{\omega} - f_i^{\omega} \geq t_{ij}$. Otherwise, if $x_{ij} = 0$, the arc (i,j) is not traversed on scenario ω, then $f_j^{\omega} - f_i^{\omega} \geq l_j - e_i$. Constraints (3.10) establish that the vehicles only can leave the depot after it opens. Constraints (3.11) establish that the vehicles must arrive on the depot before it closes. Constraints (3.12) and (3.13) indicate that the endogenous time windows of the clients must be respected. Constraint (3.14) enforce that these endogenous time windows are within the exogenous time windows. Domain variables are presented by Constraints (3.15)-(3.16).

3.3 TWAVRP in Coopservice context

Products such as medicines must be delivered to a set of hospitals, denoted by H. Each hospital $j \in H$ is visited by a single-vehicle. We consider a set of periods P representing a planning horizon. A demand q_j^p of a hospital j on period p must be met. The company has a heterogeneous fleet defined by set V of vehicles. Each vehicle $v \in V$ has capacity q^v that must be respected.

Each arc $(i, j) \in A$ has associated travel time t_{ij}^v, for each $v \in V$, and a travel cost c_{ij}. Note that the time spent going from i to j depends on the vehicle that operates the route. It is necessary to consider the regulatory limitations for truck drivers in Italy. The daily driving time must not exceed α hours. After a driving period of β hours, the driver must make an interruption of γ minutes unless he starts a rest period. A hospital j can be attended by a set of vehicles V_j, taking into account its location.

An arc $(0, |H|+1)$ indicates a vehicle does not leave the depot on that day. Technicians should be at the delivery place when the vehicle arrives to take care of the medicines. Thus, node $j \in H$ must be delivered at time window $[e_j, l_j]$. Service time is defined by s_j^v and varies according to hospital and vehicle. A time window $[e_0, l_0]$ represents the operating time of a depot.

Each customer j indicates a set of days to be visited. We assign to each customer $j \in H$ an endogenous time windows of width w_j which we will use to attend the customer in all required periods. Let be y_j the starting time of the endogenous time window at each node $j \in H$ and z_{jp}^v a binary variable equal to one if node j is visited by vehicle $v \in V$ on period $p \in P$. Concerning work hours regulations for drivers, we create artificial node $\tilde{j}$ for each hospital to represent a driver stopping point and define extended sets $\tilde{N} = H \cup \{\tilde{j}\}$ and $\tilde{A} = A \cup \{\ (i,\tilde{j}) \cup (\tilde{j},k) : i,k \in H\ \}$. Moreover, we extend the set of edges of the graph G to $A = \{(i, j, p) : i, j \in H, p \in P\}$ and $\tilde{A} = \{(i, j, p) : i, j \in \tilde{N}, p \in P\}$, where the arc (i, j, p) represents the possibility to visit nodes (i, j) on day p. We also set $s_{\tilde{j}}^v = \gamma$, and $t_{i\tilde{j}}^v = t_{\tilde{j}i}^v = 0$, $\forall i \in H$.

Let $x_{ijp}^v \in \{0, 1\}$ be a variable equal to 1 if arc $(i, j) \in A$ is traveled by vehicle $v \in V$ on scenario p. We denote z_{jp}^v as a binary variable equal to one if node j is visited by vehicle $v \in V$ on period p. Define f_{ijp}^v as the starting time to travel from node i to j by vehicle v on the period p. Our goal is to minimize the total cost of the routes and the quantity of vehicles. We present the Coopservice context model as follows.

$$\text{minimize} \sum_{(i,j,p)\in A} c_{ij} x_{ijp}^{v} + \sum_{j\in H|(0,j,p)\in A} \sum_{v\in V} x_{0jp}^{v} \tag{3.17}$$

subject to

$$\sum_{v\in V} z_{jp}^{v} = 1 \qquad \forall j \in H, \forall p \in P \tag{3.18}$$

$$\sum_{i\in \tilde{N}|(i,j,p)\in \tilde{A}} x_{ijp}^{v} = z_{jp}^{v} \qquad \forall j \in H \cup \{\tilde{j}\}, \forall v \in V, \forall p \in P \tag{3.19}$$

$$\sum_{j\in \tilde{N}|(i,j,p)\in \tilde{A}} x_{ijp}^{v} = z_{ip}^{v} \qquad \forall i \in H \cup \{\tilde{j}\}, \forall v \in V, \forall p \in P \tag{3.20}$$

$$\sum_{j\in H|(0,j,p)\in A} x_{0jp}^{v} \le 1 \qquad \forall v \in V, \forall p \in P \tag{3.21}$$

$$\sum_{j\in \tilde{N}|(i,j,p)\in \tilde{A}} f_{ijp}^{v} \ge \sum_{k\in \tilde{N}\setminus\{|H|+1\}|(k,i,p)\in \tilde{A}} [f_{kip}^{v} + (t_{ki}^{v} + s_{ip}^{v}) x_{kip}^{v}] \qquad \forall i \in \tilde{N}, \forall v \in V, \forall p \in P \tag{3.22}$$

$$e_0 x_{0jp}^{v} \le f_{0jp}^{v} \qquad \forall (0,j,p) \in A, \forall v \in V \tag{3.23}$$

$$\sum_{j\in H} q_j^p z_{jp}^{v} \le q^v \qquad \forall v \in V, \forall p \in P \tag{3.24}$$

$$\sum_{k\in \tilde{N}\setminus\{|H|+1\}} (f_{k(|H|+1)p}^{v} + t_{k(|H|+1)}^{v} x_{k(|H|+1)p}^{v}) - \sum_{j\in H} f_{0jp}^{v} \le \alpha \qquad \forall v \in V, \forall p \in P \tag{3.25}$$

$$f_{ijp}^{v} \le \alpha x_{ijp}^{v} \qquad \forall (i,j,p) \in \tilde{A}, \forall v \in V \tag{3.26}$$

$$\sum_{v\in V} z_{\tilde{j}p}^{v} = \sum_{j\in H|(0,j,p)\in A} x_{0jp}^{v} \qquad \forall j \in H, \forall p \in P \tag{3.27}$$

$$\sum_{i\in H} f_{i\tilde{j}p}^{v} \le \beta \sum_{i\in H} x_{i\tilde{j}p}^{v} \qquad \forall v \in V, \forall p \in P \tag{3.28}$$

$$\sum_{k\in H\cup\{\tilde{k}\}} f_{kjp}^{v} \ge y_j, \qquad \forall j \in H, \forall v \in V \tag{3.29}$$

$$\sum_{k\in H\cup\{\tilde{k}\}} f_{kjp}^{v} \le y_j + w_j, \qquad \forall j \in H, \forall v \in V \tag{3.30}$$

$$z_{jp}^{v} = 0 \qquad \forall v \in V, \forall v \notin V_j, \forall p \in P \tag{3.31}$$

$$x_{ijp}^{v} \in \{0,1\} \qquad \forall (i,j,p) \in A, \forall v \in V \tag{3.32}$$

$$z_{jp}^{v} \in \{0,1\} \qquad \forall j \in \tilde{v} \backslash \{0\}, \forall v \in V, \forall p \in P \tag{3.33}$$

$$f_{ijp}^{v} \in \mathbb{R}_{+} \qquad \forall (i,j,p) \in \tilde{A}, \forall v \in V \tag{3.34}$$

$$y_j \in [e_j, l_j - w_j] \qquad \forall j \in H. \tag{3.35}$$

The objective function (3.17) minimizes the total distance covered in all periods.. Constraints (3.18) state that each hospital must be visited. Constraints (3.19) and (3.20) establish that a hospital is visited if it is included in a route. Each hospital must have an entry arch traversed by a vehicle and an exit arc traversed by the same vehicle. Constraints (3.21) indicate that a vehicle that performs a route must depart from a depot at most once. Constraints (3.22) assure the starting time to attend hospital j must be greater or equal than the starting time of its previous visited hospital i, added by its service time and the time to travel to j. The time to leave the depot by vehicles is established by Constraints (3.23). Route capacity constraints are defined by inequalities (3.24). Each route must have a demand that respects the total capacity of the vehicle. Constraints (3.25) and (3.26) state daily duration. The total daily driving time of a route must not exceed α hours. For each arc (i, j), the model checks that the crossing's start time respect α. Constraints (3.27) indicate that the number of vehicles visiting $\tilde{j}$ must equal the number of vehicles leaving the depot, thus establishing that each route respects the stopping points. Constraints (3.28) establish that the driver of the vehicle reaches a β amount of driving hours he must start the resting time. Constraint (3.29) and (3.30) ensures that the endogenous time window assigned to a customer respects the customer's exogenous time window limits. Constraints (3.31) establish the set of vehicles that can visit a node j on scenario p. Domain variables are presented by Constraints (3.32)-(3.35).

The model presented was implemented in *Python 3.7.4*, using the MILP solver *Gurobi 8.1.1* running in the same computer specified on Chapter 5. We tested the algorithm in the generated instance by the process that will be presented on Chapter 4 by limiting the time to 3600 seconds, not obtaining feasible solutions. Due to the computational effort for the exact resolution of the model by means of solvers, we will approach the use of heuristic methods to solve the exposed problem in this work.

4 HYBRID ALGORITHM

This chapter presents the methodology that will be applied to solve both variants of the TWAVRP described in Chapter 3. Section 4.1 describe the proposed method adopted to find good-quality solutions. Section 4.1.1 approach the constructive method to generate the initial feasible solution for the problem. Section 4.1.2 indicate the heuristic approaches to be used based on their performance in other routing problems. Finally, Section 4.2 define a mathematical model that will assist in solving the problem.

4.1 Proposed heuristic

The heuristic method proposed in this study is outlined in Algorithm 1. It has two successive phases. The first one generates routes, while the second one selects a subset of routes having minimum cost. A similar idea was adopted by Moreira and Costa (2013), who efficiently solved a quite different combinatorial optimization problem involving job rotation schedules in assembly lines with heterogeneous workers. Our method is composed of two parts. First, we generate a pool of feasible routes, minimizing travel costs overall scenarios (stochastic problem), or the total costs overall periods (Coopservice problem) (Lines $4-7$), subject to vehicle capacity constraints, and exogenous time windows. Concerning Coopservice's problem, we also include driver stopping period constraints. Then, we call an auxiliary *Mixed Integer Linear Programming* (MILP) formulation to select the most appropriate routes of the set, to optimize the total cost overall periods (Line 8) by respecting the generated endogenous time windows. This process is performed repeatedly until a certain number of iterations n_{run} is reached. At each iteration, the MILP starts the search from the best solution found in the previous iteration.

The reference framework of Phase 1 is the ILS introduced by Lourenço, Martin and Stützle (2003). Such an algorithm has four components: (i) initial solution generator; (ii) local search procedure; (iii) perturbation; and (iv) acceptance criterion. This metaheuristic choice is derived from the fact that it has been successfully applied in several combinatorial optimization problems (AVCI; TOPALOGLU, 2017; GUNAWAN; LAU; LU, 2015; NOGUEIRA; PINHEIRO; SUBRAMANIAN, 2018), including a number of VRP variants (HADDADENE; LABADIE; PRODHON, 2016; SUBRAMANIAN; CABRAL, 2008; KRAMER; CORDEAU; IORI, 2018). Moreover, it contains fewer parameters to be fine-tuned concerning other metaheuristics. In Line 7, we represent the ILS by function *ILS*(I, α, ζ), which returns the set of routes obtained by the execution of the metaheuristic after receiving the input I, which is the data from scenario $\omega \in \Omega$ (stochastic TWAVRP) or period $p \in P$ (multi-period TWAVRP). Note that parameter α corresponds to the perturbation factor, whereas ζ gives the number of iterations without improvements. Next, we explain each component of the ILS and the subsequent mathematical formulation used to select the final routes.

Algorithm 1: Hybrid algorithm

Input: I (instance), n_{run}
Output: $(s, f(s))$ (solution, and its objective function)
$Y = \Omega$ (stochastic TWAVRP), or $Y = P$ (multi-period TWAVRP)
$s \leftarrow \emptyset$
while $n_{run} \neq 0$ **do**
  $R \leftarrow s$; ▷ Empty pool of routes
  foreach $y \in Y$ **do**
    $R \leftarrow R \cup ILS(I_y, \alpha, \zeta)$; ▷ Generating the set of routes for each period
  $s \leftarrow RSM(s, R, I)$; ▷ Route Selector Model (RSM)
  $n_{run} \leftarrow n_{run} - 1$
return $(s, f(s))$;

4.1.1 Constructive method

Algorithm 2 gives the heuristic invoked to create the initial solution for the proposed ILS. The algorithm is inspired by the greedy strategy presented by Zhigalov (2018). Let $Y = \Omega$ (stochastic TWAVRP), or $Y = P$ (multi-period TWAVRP). Let $\tilde{C}_y$ be the set of all customers $y \in Y$. First, $\tilde{C}_y$ is sorted according to the earliest start time of the exogenous time window (i.e., e_i, for $i \in \tilde{C}_y$) of the customers (Line 4). The main loop consists of Lines 5–13 and terminates when all customers have been assigned. In each iteration, an empty route is opened (Line 6), and the highest priority customers (according to the sorting in Line 4) are added to the route, one at a time if such assignment respects vehicle capacity and time window constraints (Lines 7–10). Feasibility is checked by invoking the *infeasible*($\mathscr{R}$) procedure. The method verifies the viability of a route through a procedure that checks the time spent to visit each customer, following the sequence of visits stipulated by the route. If the arrival time of the vehicle at a customer respects the exogenous time window and the capacity of the is not exceeded, the route is feasible. During this process, when the vehicle reaches a certain amount of time (multi-period TWAVRP), the stopping point is added to the route indicating that it is the driver's resting time (the stopping point is not considered for the stochastic TWAVRP variant). If the current route is feasible, the customer j is included at the end of the route ($\mathscr{R}$) under construction and then removed from $\tilde{C}_y$ (Line 13). The new cost of the route is given by the total distance traveled following the trajectory of visits stipulated by the route, considering the new customer inserted.

Algorithm 2: Constructive Heuristic (CH)

1 **Input:** I (data set), C_y (set of all available clients for a data set I for $y \in Y$)

2 **Output:** s (feasible solution)

3 $s \leftarrow \emptyset$;

4 $\tilde{\mathscr{C}} \leftarrow sort(C_y)$; ▷ sort clients in non-descending order of earliest exogenous time window

5 **while** $\tilde{\mathscr{C}} \neq \emptyset$ **do**

6 $\quad \mathscr{R} \leftarrow \emptyset$;

7 $\quad$ **foreach** $j \in \tilde{\mathscr{C}}$ **do**

8 $\quad\quad \mathscr{R} \leftarrow \mathscr{R} \cup \{j\}$;

9 $\quad\quad$ **if** *infeasible*($\mathscr{R}$) = ***true*** **then**

10 $\quad\quad\quad \mathscr{R} \leftarrow \mathscr{R} \setminus \{j\}$;

11 $\quad\quad$ **else**

12 $\quad\quad\quad \tilde{\mathscr{C}} \leftarrow \mathscr{H} \setminus \{j\}$;

13 $\quad s \leftarrow s \cup \mathscr{R}$;

14 **return** s

4.1.2 Iterated Local Search (ILS)

ILS algorithm is a metaheuristic that generates a sequence of solutions to a problem through iterative applications of improvement methods in each solution (STÜTZLE; RUIZ, 2018). These improvement methods are:

- Local Search (LS): The Local search method consists of investigating the search space through changes applied internally in a solution. In our approach, we consider that only solutions that improve the current one are accepted.
- Perturbation: Introduces modifications to a candidate solution by modifying the local optimal solution, generating possible improvements in the solution. The modification introduced for the perturbation method must be sufficiently larger than the modifications done in the local search phase (STÜTZLE; RUIZ, 2018).

The Local Search (LS) method is composed of six elementary neighborhoods:

N1 *Relocate intra-route*: change position of a customer in a route;

N2 *Swap intra-route*: swap two customer positions in a route;

N3 *2-opt*: invert a sequence of customers allocated to the same route;

N4 *Relocate inter-route*: relocate a customer to a different route in the same period;

N5 *Swap inter-route*: exchange two customers allocated in different routes, in the same period;

N6 *Cross inter-route*: split two routes at given points and exchange their remaining parts.

The LS method invokes the neighborhoods according to the procedure shown in Algorithm 3. Given a solution s, a list $NL(s)$ of neighborhoods is initialized according to the inter-route neighborhoods (N4, N5, and N6). If $s^{'}$ is feasible and the distance performed, represented by function $f(s^{'})$, decreases compared to the current solution (Line 7), an intra-route search procedure (N1, N2, and N3) is performed over s' (Lines 9–13). If the intra-route procedure improves s', the current solution $\tilde{s}$ is used to replace s^* (Line 13). The process terminates when no inter-neighborhood can return an improvement.

Algorithm 3: Local Search method (LS)

1 **Input:** s (feasible solution)
2 **Output:** s^* (best feasible solution found)
3 $s^* \leftarrow s$;
4 **foreach** $N \in NL(s)$ $\triangleright$ *$NL(s)$: list of inter-neighborhoods of solution s*
5 **do**
6   **foreach** $s' \in N$ **do**
7   **if** $f(s') < f(s^*)$ ***and*** *infeasible(s')* = ***false*** **then**
8             $s^* \leftarrow s'$;
9   **foreach** $N \in NI(s^*)$ $\triangleright$ *$NI(s^*)$: list of intra-neighborhoods of solution s'*
10   **do**
11   **foreach** $\tilde{s} \in N$ **do**
12   **if** $f(\tilde{s}) < f(s^*)$ ***and*** *infeasible($\tilde{s}$)* = ***false*** **then**
13                         $s^* \leftarrow \tilde{s}$;
14 **return** s^*

Starting from a solution s^*, the Perturbation method invokes a list of $NL(s^*)$ of possible neighborhood moves according to all neighborhood moves (N1, N2, N3, N4, N5, and N6). A percentage α of neighborhoods in $NI(s^*)$ is randomly chosen and applied to s^*.

Algorithm 4 summarizes the ILS that applied to each period of Phase 1.

Algorithm 4: Iterated Local Search (ILS)

Input: C_y (data set), α (perturbation factor), ζ (number of iterations without improvements)
Output: $\mathscr{R}$ (set of feasible solutions found)
$s^* \leftarrow \emptyset$; ▷ Best solution found so far (take $f(s^*) = +\infty$)
$s \leftarrow$ *Constructive Heuristic*(C, C_y); ▷ C_y: set of available customers of data set C
$s_{ls} \leftarrow LS(s)$;
$\mathscr{R} \leftarrow s_{ls} \cup s$; ▷ Initializing the set of feasible solutions
$s^* \leftarrow s_{ls}$;
$count \leftarrow 0$
while $count \neq \zeta$ **do**
  $s' \leftarrow Perturbation(s^*, \alpha)$;
  $s_{ls} \leftarrow LS(s')$;
  $\mathscr{R} \leftarrow \mathscr{R} \cup s' \cup s_{ls}$;
  if $f(s') < f(s^*)$ **then**
    $s^* \leftarrow s'$;
    $count \leftarrow 0$;
  else
    $count \leftarrow count + 1$;
return $\mathscr{R}$;

4.2 Route Selector Model

To approach the TWAVRP presented by Spliet and Gabor (2015), consider a set Ω of demand scenarios model demand uncertainty, each having probability of occurrence p_ω, for $\omega \in \Omega$, in such a way that $\sum_{\omega \in \Omega} p_\omega = 1$. We can use the ILS algorithm to generates a set R_ω of feasible routes for each scenario $\omega \in \Omega$ (see Algorithm 1). Note that all routes in R_ω respect the capacity and time-windows constraints for the TWAVRP, and the driver stopping periods for its variant found at Coopservice. We built a MILP formulation, called *Route Selector Model* (RSM), whose aim is to choose the most appropriate subset of routes from R_ω, assigning an endogenous time window to each client, overall scenarios.

To present the RSM, we take from R_ω: (i) f^ω_{jr} as the starting time of service on client j on the route r in scenario ω; (ii) c^ω_r as the cost to choose a route $r \in R_\omega$ in scenario ω; and (iii) z^ω_{jr} as a binary parameter equal to one if client j belongs to route $r \in R_\omega$ in scenario ω, 0 otherwise. Consider u^ω_r as a binary variable equal to one if route $r \in R_\omega$ is selected, 0 otherwise, and y_i as a continuous variable that measures the starting time of the endogenous time window of customer $i \in H$. Recall that, as indicated above, w_i gives the time window width of customer i. The RSM is as follows:

$$\min \sum_{\omega \in \Omega} p_\omega c_r^\omega u_r^\omega \tag{4.1}$$

subject to

$$\sum_{r \in R_\omega} z_{jr}^\omega u_r^\omega = 1 \qquad \forall j \in H, \omega \in \Omega \tag{4.2}$$

$$\sum_{r \in R_\omega} f_{jr}^\omega z_{jr}^\omega u_r^\omega \geq y_j \qquad \forall j \in H, \omega \in \Omega \tag{4.3}$$

$$\sum_{r \in R_\omega} f_{jr}^\omega z_{jr}^\omega u_r^\omega \leq y_j + w_i \qquad \forall j \in H, \omega \in \Omega \tag{4.4}$$

$$y_j \in [e_j, l_j - w_j] \qquad \forall j \in H, \omega \in \Omega \tag{4.5}$$

$$u_r^\omega \in \{0,1\} \qquad \forall \omega \in \Omega, r \in R_\omega. \tag{4.6}$$

The model optimizes the total cost of the selected routes. Constraints (4.2) indicate that each customer has to be served in all scenarios by a single route. Constraints (4.3)–(4.4) establish the endogenous time windows. Domain variables are presented by Constraints (4.5)–(4.6).

4.2.1 Route Selector Model on Coopservices context

For the Coopservices context, we relax Constraints (4.4) since delays are common due to climatic reasons, traffic flow, vehicle problems, among others. On the other hand, we penalize with an additional cost in the objective function. The use of the penalty can be beneficial for real routing situations since, some routes made unfeasible by hard constraints, can be viable and useful in real-world use cases that tend to present more relaxed constraints. The penalty will be normalized by he sum of the size of the planning horizon ($|P|$), number of customers ($|H|$) and the departure depot time ($l_{|H|+1}$). In addition, it is important to emphasize that the costs for the use of vehicles will not be considered here, since the tests performed with these data affected the results in order to generate inconsistencies. Considering the variables previously defined in this section, the RSM for the Coopservices context reads:

$$\min \sum_{p \in P} \sum_{r \in R_p} c_r^p u_r^p + \left(\frac{1}{l_{|H|+1} * |P| * |H|} \right) * \left(\sum_{p \in P} \sum_{j \in H} \sum_{r \in R_p} f_{jr}^p x_{jr}^p u_r^p - y_j - w_j \right) \tag{4.7}$$

subject to

$$\sum_{r\in R_p} z_{jr}^p u_r^p = 1 \qquad \forall j \in H, p \in P \tag{4.8}$$

$$\sum_{r\in R_p} f_{jr}^p z_{jr}^p u_r^p \geq y_j \qquad \forall j \in H, p \in P \tag{4.9}$$

$$y_j \in [e_j, l_j - w_j] \qquad \forall j \in H \tag{4.10}$$

$$u_r^p \in \{0,1\} \qquad \forall p \in P, r \in R_p. \tag{4.11}$$

The model optimizes the total cost of the selected routes that are penalized by the normalization of the violation total time of the upper limit of the endogenous time windows in all periods. Constraints (4.8) indicate that each customer has to be served in all periods by a single route. Constraint (4.9) establish the endogenous time windows. Domain variables are presented by Constraints (4.10) and (4.11).

5 COMPUTATIONAL EXPERIMENTS

We performed a set of computational experiments to assess the performance of the ILS-based algorithm that we developed for the TWAVRP. The algorithms were implemented in *Python 3.7.4*, using the MILP solver *Gurobi 8.1.1* for the development of the RSM (Section 4.2.1), running a single thread for a time limit of 3600 seconds on each instance. All experiments were performed on a PC Intel i7, 3.5 GHz with 16 GB RAM, similar to the computer used by Dalmeijer and Spliet (2018).

To generate the pool of routes, Algorithm 4 was executed five times on each instance. This number was tuned through preliminary tests in which we obtained a good trade-off between quality and computational effort. Furthermore, this value allowed the algorithm to make good use of its stochastic components. The number of iterations without improvements (ζ) and the perturbation percentage (α) were fine-tuned through the *Irace* package (LÓPEZ-IBÁÑEZ et al., 2016). For that purpose, we generated 200 training instances by using the instance generator proposed by Dalmeijer and Spliet (2018). The values returned by the *Irace* package at the end of this test were $\zeta = 100$ and $\alpha = 0.35$.

5.1 TWAVRP Instances

We use the set of TWAVRP instances proposed by Spliet, Dabia and Woensel (2018). Each instance considers a different combination of the number of customers, vehicle capacity, demand for each scenario, probability of each scenario, size of exogenous and endogenous time windows, travel costs, and travel times. The data set comprises ninety instances divided into two classes: small instances and large ones. Small instances contain four sets of ten problems, each with 10, 15, 20, and 25 customers, while large instances contain five sets of ten instances each, with 30, 35, 40, 45, and 50 customers. The customer's coordinates were generated as uniformly distributed over a square with sides of length five. The depot is located in the center of the square. Each instance includes demands for each customer in three scenarios with equal probability of occurrence. Exogenous time windows are distributed as follows: a time window $[10,16]$ is given to 10% of the customers; $[7,21]$ to 30% of the customers; and $[8,18]$ to the remaining 60%. The width of the endogenous time window is set to $w_i = 2$ for all customers. The costs and the travel times between the nodes were obtained by calculating the Euclidean distances between their coordinates.

5.2 Coopservice instance

We need to define the following data to evaluate Coopservice's instance:

- Service time: time spent by each customer to execute the service;
- Exogenous time windows: time window of the customer;
- Demand: quantity required by each customer;
- Distance between customers: the real distance between each customer;
- Travel time: real travel time between each customer.

We have the availability of a real-world database provided by Coopservice with the following information:

- Customers_EGAS: list all the customers that must be visited by vehicles. Names and coordinates of each customer are presented.
- Myway_AVEN_EGAS: present the routes executed by Coopservice. The database shows a list of GPS tracks with the exact coordinates of a given vehicle at a given time from the data latitude, longitude, event date, and vehicle identification. From this information, it is possible to estimate the service time of each customer.
- Friulli_Venezia_instance: Given each customer, the exogenous time windows and demand for each scenario on the Coopservice context. In addition, vehicle information (capacity and identification) is made available.

Service time for each customer in the context of Coopservice needs to be estimated from the data available. For this purpose, we use the customers' location and the GPS tracks given by the Myway_AVEN_EGAS database.

5.2.1 Data Preparation

To start the process of estimate service times for each customer, the databases described in Section 5.2 need to be prepared. This phase is important in order to remove noises or problems that may interfere with the process.

5.2.1.1 GPS tracks data preparation

First, failures and duplicated data on the available GPS tracks in the database "Myway_AVEN-_EGAS" must be removed. Columns that do not have a fundamental role to the estimation process were also excluded. After this process, the database has the following attributes:

Table 5.1 – Database "Myway_AVEN_EGAS" after remove unused columns.

Datetime	Latitude	Longitude	Vehicle

Source: Authors.

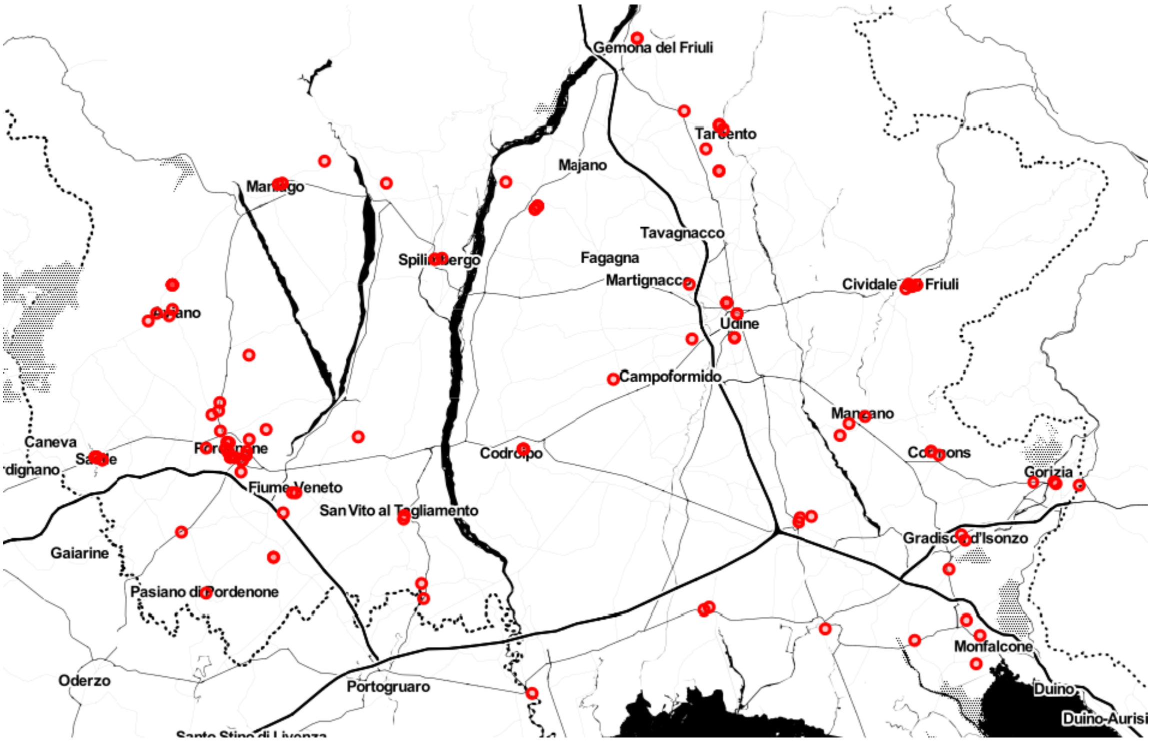
The columns on Table 5.1 indicates:

1. Latitude: latitude of the GPS track.
2. Longitude: longitude of the GPS track.
3. Datetime: exact date and time that the GPS track is traveled.
4. Vehicle: identification of the vehicle traveling along the GPS track.

5.2.1.2 Customers data preparation

The database "Customers_EGAS" lists all the customers that must be visited by the vehicles. In Figure 5.1, we can see all the customers illustrated on a map.

Figure 5.1 – Customers that must be visited.



Source: Authors with the library *Folium*.

When each customer is plotted on a map with a determined radius in meters around their coordinates, it is possible to identify that some customer's radius is in the intersection. This problem occurs because some customers are served at the same location or have near locations. Figure 5.2 shows intersect customers in the Pordenone region (Italy).

Figure 5.2 – *Pordenone* customers.

Source: Authors with the library *Folium*.

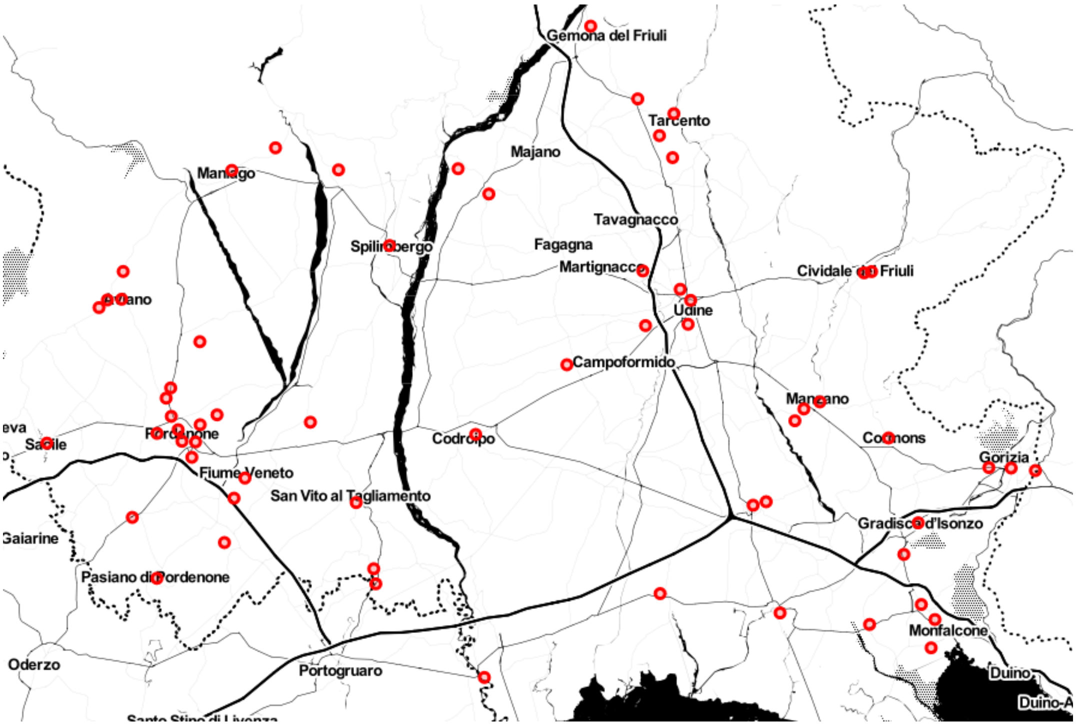
To address this problem, customers who have intersections of the radius will be grouped into a single cluster. These clusters will be called "Macro Customers." The new position of the Macro Customer is given by the average coordinates of all the customers aggregated. Figure 5.3 shows the Macro Customers found in the Pordenone region.

Figure 5.3 – Macro customers founded on *Pordenone* region.

Source: Authors with the library *Folium*.

After this process, it is possible to observe by Figure 5.4 that customers' radius becomes better defined since all intersections between them are eliminated.

Figure 5.4 – Customers plotted after the process of eliminate intersections.



Source: Authors with the library *Folium*.

5.2.2 Service time estimation

The main idea of the algorithm for estimating service times is to check the GPS tracks located within the radius of a given customer. Thus, through the GPS track timestamp, it is possible to estimate the service time for each customer that is visited on a given day. The estimated service times for Macro Customers are shared with all customers aggregated by him. If a customer is visited more than once, his final service time will be estimated by the average service time of all visits.

To illustrate the algorithm steps, we will give an application example comprising four customers who must have estimated their service times. For this purpose, a database with 50 GPS tracks covered in a single day is made available.

Table 5.2 shows the customers who are expected to have their service times estimated. Each customer has a name and their coordinates.

Table 5.2 – Customers.

Customer	Latitude	Longitude
Customer 1	45.811363220214844	13.443992614746094
Customer 2	45.81528854370117	13.518794059753418
Customer 3	45.93513870239258	13.606978416442871
Customer 4	45.936641693115234	13.604848861694336

Source: Authors.

The GPS tracks database is sorted and grouped by days, as shown in Table 5.3. Each GPS track is represented by a coordinate, the date time stamp when the point was visited, and the vehicle responsible for the service.

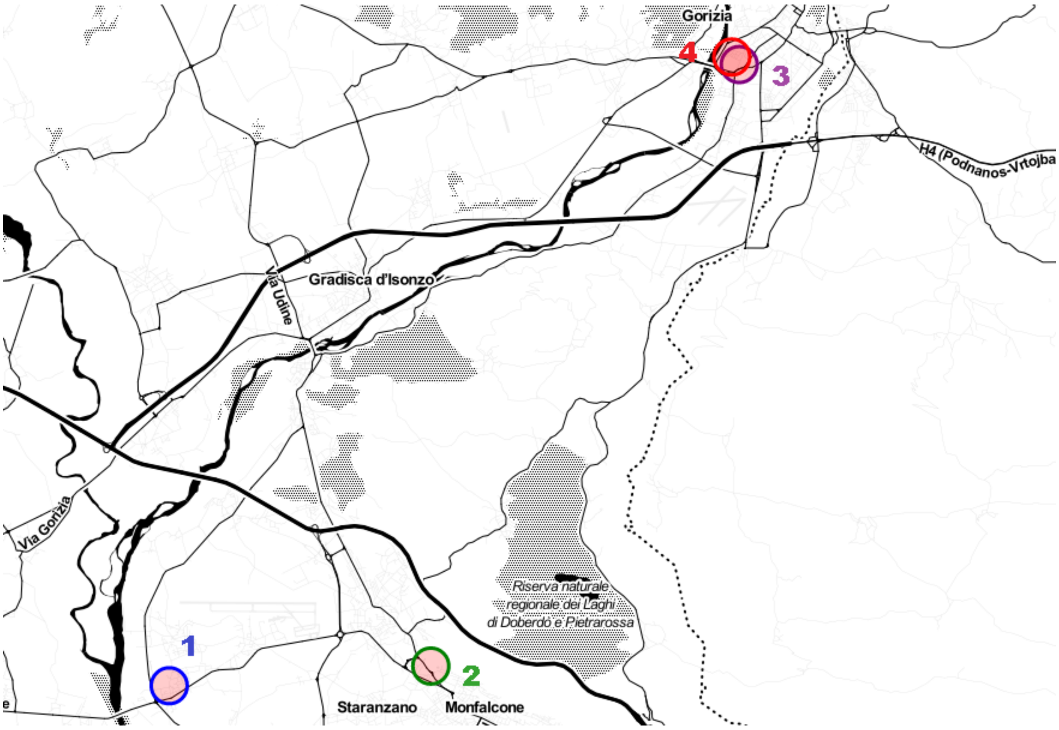
Table 5.3 – GPS tracks order by time.

Datetime	Latitude	Longitude	Vehicle
2019-12-24 07:02:06	45.8116836547852	13.4440231323242	Truck 1
2019-12-24 07:03:06	45.8116874694824	13.4440250396729	Truck 1
2019-12-24 07:03:56	45.8116874694824	13.4440269470215	Truck 1
2019-12-24 07:04:06	45.8116874694824	13.4440269470215	Truck 1
2019-12-24 07:05:06	45.8129196166992	13.4443969726562	Truck 1
2019-12-24 07:06:06	45.8097763061523	13.4441337585449	Truck 1
2019-12-24 07:07:06	45.8109970092773	13.449257850647	Truck 1
2019-12-24 07:08:06	45.8130378723145	13.453784942627	Truck 1
2019-12-24 07:09:06	45.8147773742676	13.4638738632202	Truck 1
2019-12-24 07:10:08	45.8195877075195	13.4769468307495	Truck 1
2019-12-24 07:11:08	45.8210525512695	13.4888553619385	Truck 1
2019-12-24 07:12:08	45.8222541809082	13.498927116394	Truck 1
2019-12-24 07:13:08	45.816162109375	13.5084781646729	Truck 1
2019-12-24 07:13:56	45.816520690918	13.5128288269043	Truck 1
2019-12-24 07:14:08	45.816837310791	13.5145568847656	Truck 1
2019-12-24 07:15:08	45.8165969848633	13.5186452865601	Truck 1
2019-12-24 07:16:08	45.8157005310059	13.5188283920288	Truck 1
2019-12-24 07:17:08	45.8156089782715	13.5188436508179	Truck 1
2019-12-24 07:18:08	45.8154754638672	13.5190830230713	Truck 1
2019-12-24 07:41:14	45.8176689147949	13.5064611434937	Truck 1
2019-12-24 07:42:14	45.8223075866699	13.4993801116943	Truck 1
2019-12-24 07:43:14	45.8218154907227	13.4947547912598	Truck 1
2019-12-24 07:44:12	45.832160949707	13.4924631118774	Truck 1
2019-12-24 07:45:12	45.845100402832	13.4775266647339	Truck 1
2019-12-24 07:46:12	45.8516426086426	13.4613952636719	Truck 1
2019-12-24 07:47:10	45.8545455932617	13.4437046051025	Truck 1
2019-12-24 07:47:12	45.8585624694824	13.4286031723022	Truck 1
2019-12-24 07:48:12	45.8727416992188	13.4501352310181	Truck 1
2019-12-24 07:50:12	45.8922500610352	13.4722166061401	Truck 1
2019-12-24 07:51:12	45.8982238769531	13.4871912002563	Truck 1
2019-12-24 07:53:12	45.9010848999023	13.5223779678345	Truck 1
2019-12-24 07:54:12	45.9032554626465	13.5401086807251	Truck 1
2019-12-24 07:55:14	45.9032173156738	13.540472984314	Truck 1
2019-12-24 07:56:14	45.9032173156738	13.5404720306396	Truck 1
2019-12-24 07:57:14	45.9032173156738	13.5404682159424	Truck 1
2019-12-24 07:58:14	45.9032173156738	13.5404653549194	Truck 1
2019-12-24 07:59:14	45.9032211303711	13.5404663085938	Truck 1
2019-12-24 08:04:14	45.9033737182617	13.5450382232666	Truck 1
2019-12-24 08:09:14	45.9039726257324	13.5631465911865	Truck 1
2019-12-24 08:11:14	45.9045715332031	13.5823211669922	Truck 1
2019-12-24 08:12:14	45.912166595459	13.5976161956787	Truck 1
2019-12-24 08:14:14	45.9231300354004	13.6133480072021	Truck 1
2019-12-24 08:15:14	45.9320220947266	13.6131134033203	Truck 1
2019-12-24 08:16:14	45.9346733093262	13.6110515594482	Truck 1
2019-12-24 08:17:14	45.9355621337891	13.607138633728	Truck 1
2019-12-24 08:18:14	45.9355850219727	13.6037902832031	Truck 1
2019-12-24 08:19:14	45.9357032775879	13.6049213409424	Truck 1
2019-12-24 08:20:14	45.9357948303223	13.6050386428833	Truck 1
2019-12-24 08:21:16	45.9357948303223	13.6050367355347	Truck 1

Source: Authors.

When the customers coordinates given on Table 5.2 are plotted on a map, we can identify that customers 3 and 4 have an intersection between their radius. Figure 5.5 shows the intersections.

Figure 5.5 – Customers plotted before the process of eliminate intersections.



Source: Authors with the library *Folium*.

As described in Subsection 5.2.1.2, a Macro Customer should aggregate customers 3 and 4. The average will give the location of this Macro Customer between the coordinates of the customers replaced. The Macro Customer attributes are shown in Table 5.4.

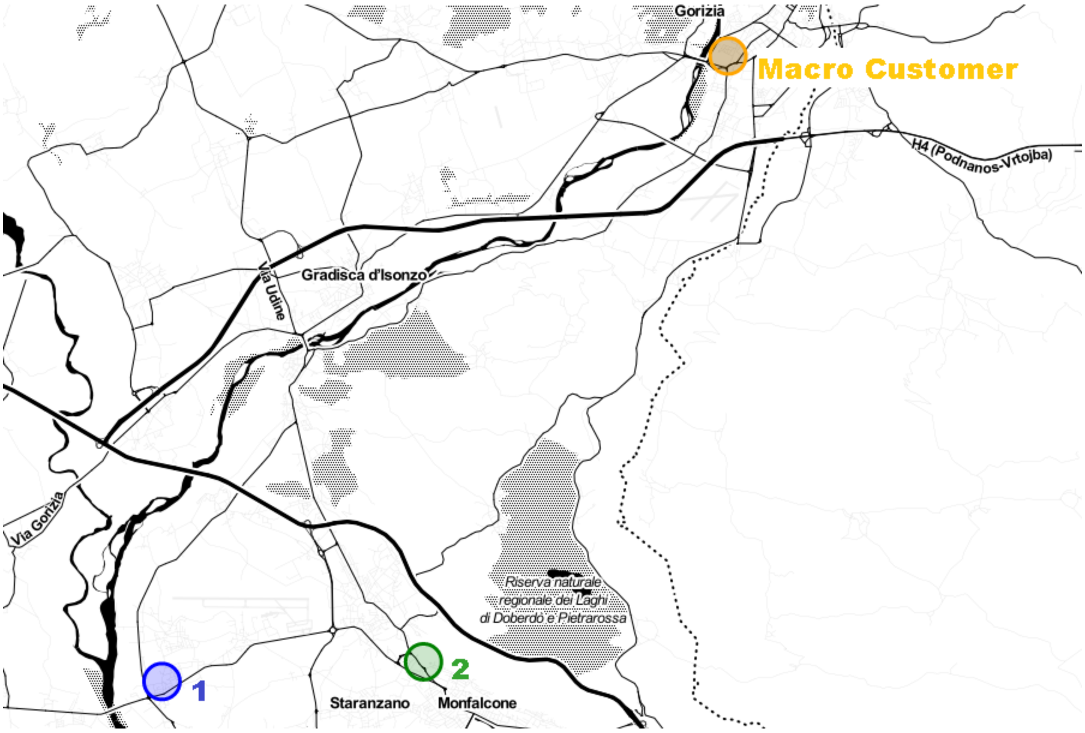
Table 5.4 – Customers after the process of eliminate intersections.

Index	Name	Latitude	Longitude
1	Customer 1	45.811363220214844	13.443992614746094
2	Customer 2	45.81528854370117	13.518794059753418
3	Macro Customer	45.813325881958	13.4813933372498

Source: Authors.

The map represented by Figure 5.6 shows the customer's coordinates plotted after the Macro Customer creation process.

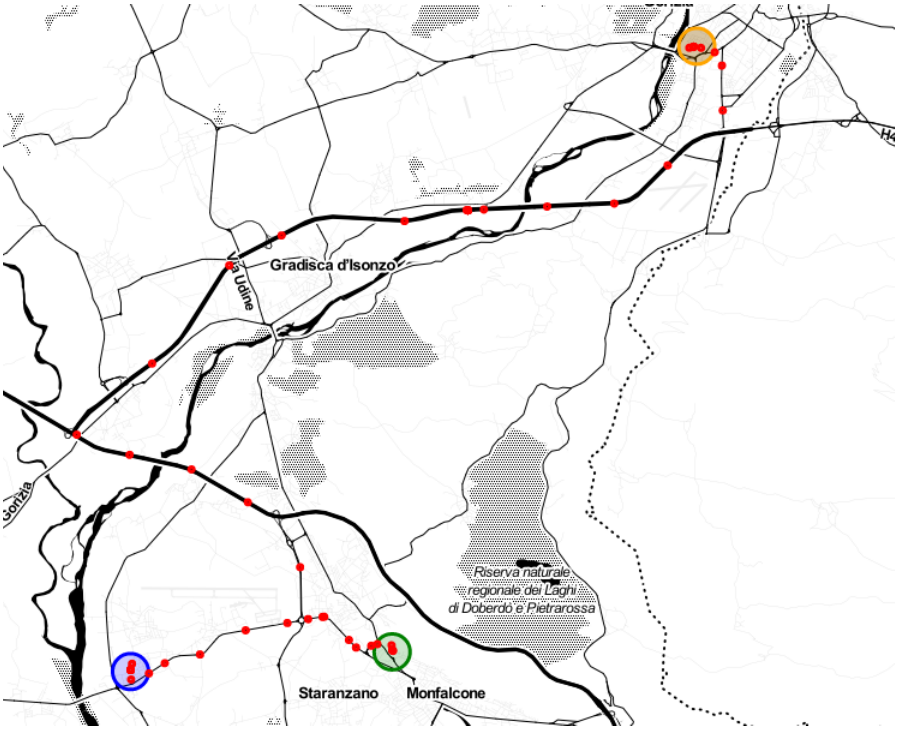
Figure 5.6 – Customers plotted after the process of eliminate intersections.



Source: Authors with the library *Folium*.

By plotting the GPS tracks represented in the Table 5.3 on the map, it is possible to identify the points that are located within the radius limits of each client, as can be seen in the Figure B.5. This process can be verified by measuring the Euclidean distance between the coordinates of the GPS track and the customer. If this distance is less than the defined radius, it is proven that the vehicle is within the radius limits of the customer.

Figure 5.7 – Customers plotted after the process of eliminate intersections.



Source: Authors with the library *Folium*.

Table 5.5 was built based on the Figure B.5. Column "Customer" represents the customer to which the GPS track is associated. If the point is not associated with any customer, it is defined that the vehicle issued the point during the journey traveled.

Table 5.5 – GPS tracks associated with customers.

Datetime	Latitude	Longitude	Vehicle	Customer
2019-12-24 07:02:06	45.8116836547852	13.4440231323242	Truck 1	Customer 1
2019-12-24 07:03:06	45.8116874694824	13.4440250396729	Truck 1	Customer 1
2019-12-24 07:03:56	45.8116874694824	13.4440269470215	Truck 1	Customer 1
2019-12-24 07:04:06	45.8116874694824	13.4440269470215	Truck 1	Customer 1
2019-12-24 07:05:06	45.8129196166992	13.4443969726562	Truck 1	Customer 1
2019-12-24 07:06:06	45.8097763061523	13.4441337585449	Truck 1	Customer 1
2019-12-24 07:07:06	45.8109970092773	13.449257850647	Truck 1	traveling
2019-12-24 07:08:06	45.8130378723145	13.453784942627	Truck 1	traveling
2019-12-24 07:09:06	45.8147773742676	13.4638738632202	Truck 1	traveling
2019-12-24 07:10:08	45.8195877075195	13.4769468307495	Truck 1	traveling
2019-12-24 07:11:08	45.8210525512695	13.4888553619385	Truck 1	traveling
2019-12-24 07:12:08	45.8222541809082	13.498927116394	Truck 1	traveling
2019-12-24 07:13:08	45.816162109375	13.5084781646729	Truck 1	traveling
2019-12-24 07:13:56	45.816520690918	13.5128288269043	Truck 1	traveling
2019-12-24 07:14:08	45.816837310791	13.5145568847656	Truck 1	Customer 2
2019-12-24 07:15:08	45.8165969848633	13.5186452865601	Truck 1	Customer 2
2019-12-24 07:16:08	45.8157005310059	13.5188283920288	Truck 1	Customer 2
2019-12-24 07:17:08	45.8156089782715	13.5188436508179	Truck 1	Customer 2
2019-12-24 07:18:08	45.8154754638672	13.5190830230713	Truck 1	Customer 2
2019-12-24 07:41:14	45.8176689147949	13.5064611434937	Truck 1	traveling
2019-12-24 07:42:14	45.8223075866699	13.4993801116943	Truck 1	traveling
2019-12-24 07:43:14	45.8218154907227	13.4947547912598	Truck 1	traveling
2019-12-24 07:44:12	45.832160949707	13.4924631118774	Truck 1	traveling
2019-12-24 07:45:12	45.845100402832	13.4775266647339	Truck 1	traveling
2019-12-24 07:46:12	45.8516426086426	13.4613952636719	Truck 1	traveling
2019-12-24 07:47:10	45.8545455932617	13.4437046051025	Truck 1	traveling
2019-12-24 07:47:12	45.8585624694824	13.4286031723022	Truck 1	traveling
2019-12-24 07:48:12	45.8727416992188	13.4501352310181	Truck 1	traveling
2019-12-24 07:50:12	45.8922500610352	13.4722166061401	Truck 1	traveling
2019-12-24 07:51:12	45.8982238769531	13.4871912002563	Truck 1	traveling
2019-12-24 07:53:12	45.9010848999023	13.5223779678345	Truck 1	traveling
2019-12-24 07:54:12	45.9032554626465	13.5401086807251	Truck 1	traveling
2019-12-24 07:55:14	45.9032173156738	13.540472984314	Truck 1	traveling
2019-12-24 07:56:14	45.9032173156738	13.5404720306396	Truck 1	traveling
2019-12-24 07:57:14	45.9032173156738	13.5404682159424	Truck 1	traveling
2019-12-24 07:58:14	45.9032173156738	13.5404653549194	Truck 1	traveling
2019-12-24 07:59:14	45.9032211303711	13.5404663085938	Truck 1	traveling
2019-12-24 08:04:14	45.9033737182617	13.5450382232666	Truck 1	traveling
2019-12-24 08:09:14	45.9039726257324	13.5631465911865	Truck 1	traveling
2019-12-24 08:11:14	45.9045715332031	13.5823211669922	Truck 1	traveling
2019-12-24 08:12:14	45.912166595459	13.5976161956787	Truck 1	traveling
2019-12-24 08:14:14	45.9231300354004	13.6133480072021	Truck 1	traveling
2019-12-24 08:15:14	45.9320220947266	13.6131134033203	Truck 1	traveling
2019-12-24 08:16:14	45.9346733093262	13.6110515594482	Truck 1	traveling
2019-12-24 08:17:14	45.9355621337891	13.607138633728	Truck 1	Macro Customer
2019-12-24 08:18:14	45.9355850219727	13.6037902832031	Truck 1	Macro Customer
2019-12-24 08:19:14	45.9357032775879	13.6049213409424	Truck 1	Macro Customer
2019-12-24 08:20:14	45.9357948303223	13.6050386428833	Truck 1	Macro Customer
2019-12-24 08:21:16	45.9357948303223	13.6050367355347	Truck 1	Macro Customer

Source: Authors.

Using the data issued in Table 5.5, the column "Datetime" can be used to estimate the services time of the customers (and Macro Customers) visited. If a customer (or Macro Customers) is visited more than once, his/her service time is measured by average of all his/her services. Table 5.6 shows the estimated service times for each of the customers visited during the route.

Table 5.6 – Instance generated with Macro Customer.

Name	Latitude	Longitude	Start	End	Service time
Customer_1	45.811684	13.444023	2019-12-24 07:02:06	2019-12-24 07:06:06	4
Customer_2	45.816837	13.514557	2019-12-24 07:14:08	2019-12-24 07:18:08	4
Macro Customer	45.935139	13.606978	2019-12-24 08:17:14	2019-12-24 08:21:16	4

Source: Authors.

Finally, the service time estimated for the Macro Customer is expanded to all customers aggregated by him.

Table 5.7 – Instance generated with customers aggregated by the Macro Customer.

Name	Latitude	Longitude	Start	End	Service time
Customer_1	45.811684	13.444023	2019-12-24 07:02:06	2019-12-24 07:06:06	4
Customer_2	45.816837	13.514557	2019-12-24 07:14:08	2019-12-24 07:18:08	4
Customer_3	45.935139	13.606978	2019-12-24 08:17:14	2019-12-24 08:21:16	4
Customer_4	45.935139	13.606978	2019-12-24 08:17:14	2019-12-24 08:21:16	4

Source: Authors.

5.3 Experiment 1: comparison with the literature

The experiments compare our Hybrid algorithm (Algorithm 1) with *Branch-and-Cut* (B&C) proposed by Dalmeijer and Spliet (2018), the state-of-art method for the solution of the TWAVRP. We consider two ways to generate the pool of solutions: (i) the first one, the algorithm executes with number of iterations (n_{run}) equal to one, $\zeta = 100$ and $\alpha = 0.35$. We referred to this approach simply as HA-SR (Hybrid algorithm with a single run of ILS); (ii) in the second version (HA), the algorithm executes with $n_{run} = 10$, $\zeta = 10$ and $\alpha = 0.35$. This ensures that at HA, the RSM can be executed several times (generating a different pool for the RSM at each execution) and only once at the HA-SR. The results that we obtained are summarized in Table 5.8 and Table 5.9.

Table 5.8 presents the CPU time for instances aggregated by the number of customers. Columns B&C, HA-SR, and HA, under the group CPU time (seconds), give the computational time spent by, respectively, the algorithm by Dalmeijer and Spliet (2018), the HA with a single execution of ILS and the HA with ten executions of ILS. Regarding instances that have between 10 to 35 clients, we can observe that the average CPU time of the five executions of both HA-SR and HA are higher than the B&C execution time for the small instances (10-25 customers). For larger instances (45 to 50 clients), the average CPU time of the proposed methods is less than the execution of B&C, on average 75,16% for the HA-SR and 48,66% for the HA. Detailed results are given in Appendix A.

Table 5.8 – Average CPU time aggregated by number of customers (10 instances per line, 5 executions per instance)

Instance	CPU Time (seconds)		
N. customers	B&C	HA-SR	HA
10	0.10	10.97 ± 0.58	22.54 ± 0.65
15	4.50	36.12 ± 3.57	65.36 ± 1.02
20	2.20	80.14 ± 8.31	135.87 ± 2.45
25	12.40	142.31 ± 10.21	269.94 ± 8.36
30	544.00	222.68 ± 16.87	502.68 ± 7.02
35	1531.70	341.96 ± 40.22	829.31 ± 22.65
40	3252.00	479.20 ± 39.41	1176.10 ± 10.39
45	3600.00	682.21 ± 65.50	1180.29 ± 12.63
50	3600.00	980.69 ± 180.30	1482.21 ± 406.46

Source: Authors.

Table 5.9 summarized the average best solution found for instances aggregated by the number of customers. The columns named Deviation(%) and Deviation*(%) indicate the gap and the standard deviation of the solution value found overall repetitions concerning the best solution value, and for the best-known values obtained by the B&C method, respectively. The values of the Deviation(%) and Deviation*(%) columns were computed by means of $\frac{z_{method} - z_{B\&C}}{z_{method}} \times 100$, where "method" represent the HA-SR or the HA.

Table 5.9 – Average results aggregated by number of customers (10 instances per line, 5 executions per instance)

Instance	HA-SR		HA	
N. customers	Deviation(%)	Deviation*(%)	Deviation(%)	Deviation*(%)
10	0.74 ± 2.45	0.86 ± 2.45	0.69 ± 2.51	0.81 ± 2.49
15	0.32 ± 3.36	0.61 ± 3.36	0.39 ± 3.44	0.68 ± 3.48
20	0.25 ± 2.09	0.38 ± 2.09	0.30 ± 2.03	0.40 ± 2.03
25	0.45 ± 2.16	0.75 ± 2.16	0.45 ± 2.20	0.62 ± 2.23
30	0.51 ± 1.98	0.66 ± 1.98	0.46 ± 1.97	0.64 ± 1.96
35	0.45 ± 2.07	0.62 ± 2.07	0.40 ± 2.09	0.55 ± 2.10
40	0.45 ± 1.78	0.69 ± 1.78	0.48 ± 1.84	0.65 ± 1.85
45	-0.38 ± 3.43	-0.02 ± 3.43	-0.14 ± 3.16	0.11 ± 3.18
50	-1.59 ± 2.83	-1.39 ± 2.83	-1.23 ± 2.83	-0.94 ± 2.83

Source: Authors.

Regarding instances that have between 10 to 35 customers, we can observe relative average deviations from 2.47% to 3.47% found by the HA-SR and 2.43% to 4.16% found by the HA, compared with the B&C solutions in the worst case. In the group of larger instances (45-50 clients), the ILS outperforms the results found in literature concerning both best-found and average solution values of the five performed tests.

Table 5.10 highlights the behavior of our method on the 20 larger instances having 45 and 50 customers. We report the lower bound and upper bound obtained in Dalmeijer and Spliet (2018) (columns LB and UB, respectively), and the best (column Best) and average (column Avg) solution values found by our HA-SR and HA algorithms. In the problems with 45 clients, both methods were competitive, each finding the best results for about half of the cases. Our methods improved the solution cost obtained by the B&C for all instances with 50 clients, both considering columns Best and Avg. We estimate that the diversity of routes caused by different local search operators was beneficial for the performance of the proposed methods for these most difficult instances. The reason for the better performance of HA-SR compared to HA in relation to the value found is due to the fact that the parameter n_iter (iterations without improvement) of the ILS influences the generation of varied routes more than the execution of the entire process iteratively for a longer time. The gain from the inclusion of several ILS iterations in HA does not generate much difference in the quality of the solution in proportion to the growth of computational time. Overall, we can conclude that the HA-SR and the HA are appropriate approaches for moderate and large size instances of the TWAVRP.

Table 5.10 – Results for instances with 45-50 customers (best UB values appear in bold)

Instance		B&C		HA		HA-SR	
#	N. customers	LB	UB	Best UB	Avg UB	Best UB	Avg UB
71	45	49.52	51.78	**51.34**	**51.57**	51.43	51.59
72	45	50.73	52.13	**51.93**	**52.09**	52.04	52.13
73	45	41.5	**41.7**	42.09	42.18	42.17	42.25
74	45	47.25	**47.84**	47.95	48.21	48.22	48.39
75	45	48.77	49.86	49.82	49.97	**49.81**	**49.96**
76	45	48.38	52.09	**50.25**	**50.46**	50.35	50.47
77	45	50.09	**51.18**	51.53	51.61	51.63	51.76
78	45	52.02	53.95	**53.66**	**53.74**	53.72	53.84
79	45	47.45	**48.21**	48.32	48.47	48.40	48.50
80	45	49.57	50.57	**50.4**	**50.77**	50.69	50.87
81	50	56.81	58.85	**58.31**	**58.40**	58.50	58.72
82	50	51.5	53.2	**53.02**	**53.14**	53.20	53.28
83	50	57.45	60.67	**58.86**	**58.92**	58.97	59.09
84	50	52.31	56.38	**54.3**	**54.39**	54.33	54.54
85	50	53.74	56.07	**55.22**	**55.42**	55.51	55.68
86	50	51.68	54.76	**53.37**	**53.53**	53.48	53.70
87	50	52.47	54.14	**53.88**	**53.97**	53.98	54.13
88	50	54.82	56.91	**56.44**	**56.53**	56.63	56.83
89	50	59.23	61.51	**60.53**	**60.68**	61.22	61.35
90	50	57.68	59.55	**59.17**	**59.24**	59.29	59.41

Source: Authors.

5.4 Experiment 2: Coopservice instance

Subsection 5.2.2 shows that a relevant parameter to estimate service times is the radius size chosen for each customer. The radius size is fixed for all customers and can directly interfere in the results since it can improve the accuracy of the algorithm and eliminate noisy associations between customers and GPS tracks. Thus, a study was carried out, aiming to verify which is the best value to be chosen for the parameter.

Using 3-month GPS tracks for 2019, we test 16 different radius sizes in the range of 100 to 800 meters. For each radius size tested, the following data were collected: number of customers found, the total number of Macro Customers identified, and the number of aggregated customers by Macro Customers. Table 5.11 shows the values obtained.

Table 5.11 – Comparative between different radius for 3 months GPS tracks.

Radius	Estimated customers	Estimated customers (%)	Aggregated customers	Total Macro Customers
100	46	37.70%	38	25
150	74	60.66%	41	27
200	77	63.11%	48	26
250	91	74.59%	51	27
300	96	78.69%	55	27
350	100	81.97%	61	28
400	105	**86.07%**	63	29
450	112	**91.80%**	64	**30**
500	112	**91.80%**	64	**30**
550	114	**93.44%**	66	**30**
600	115	**94.26%**	68	29
650	115	**94.26%**	69	29
700	115	**94.26%**	72	28
750	118	**96.72%**	72	28
800	118	**96.72%**	74	28

Source: Authors.

For comparison purposes, it was defined, according to the company, that will be accepted only results where the rate of customers found was greater than 85%. It is possible to observe that only a radius greater than 450 meters can maintain this rate. The number of customers found increases proportionally to the number of Macro Customers. In this way, the radius that has the largest number of Macro Customers identified is 450, 500 and 550 meters. We opted for 450 meters because it is the size that most closely matches the real average of the customer's radius.

To estimate the service time for customers, the first database "Myway_AVEN_EGAS" is grouped for months. We could identify one month for 2017, nine months for 2019, and one month for 2020. For each month, the customers found on the GPS tracks have their estimated service times. Finally, averages of estimated service times for customers found in all months are measured.

The database "Friulli_Venezia_instance" gives the exogenous time windows, demand for each period, and vehicle capacities. Database "Customers_EGAS" is used to place customers on the map and calculate the distances and travel times among them. For that purpose, we use the open-source library *Open Source Routing Machine* available in python language.

5.4.1 Computational results

The experiments compare our HA with a current solution used on Coopservice to solve the generated real instance, which has a cost of 1,172.41, based on the objective function , and uses eight vehicles to serve 122 customers. To run the HA, we adopt the same parameters reported in Section 4.2.1. For the real context instance, we relax Constraints (4.4) described in Subsection 4.2.1 since delays are common due to climatic reasons, traffic flow, vehicle problems, among others. In this case, we need to use the model described on 4.2.1.

Table 5.12 summarized the aggregated results for the real context instance. Columns Cost, Time(s), Vehicles, and Deviation(%), under the group Avg. HA, gives the average cost, computational time, average number of vehicles used by the found solutions, and the deviation compared with the current solution used to cover all the customers. Columns Cost, Time(s), Vehicles and Gap(%), under the group Best HA, gives the cost, the computational time, average number of vehicles used by the found solutions, and the deviation between the current solution and the best solution found by HA used to cover all the customers.

Table 5.12 – Results for real context instance with 122 customers

Instance			Current solution		Avg. HA				Best HA			
$\lvert H \rvert$	$\lvert P \rvert$	Vehicles	Cost	Vehicles	Cost	Time(s)	Vehicles	Deviation(%)	Cost	Time(s)	Vehicles	Deviation(%)
122	5	8	1172.42	8	1057.33	9700.36	7.52	-10.89	1052.87	9963.57	7.4	-11.35

Source: Authors.

We can observe that the ILS generates results that improve the current solution used by Coopservice. The HA approach found solutions at an average cost of 1,057.33, representing a deviation0 of -10.89% compared to the current solution cost. Regarding the best solution, the HA approach found a cost of 1,052.87, which represents a deviation of -11.35% compared to the current solution cost.

The HA-SR results, using the set of parameters described in Section 5.4.1, are reported on Table 5.13. Columns Cost, Time(s), Vehicles and Gap(%) have the same meanings as those of Table 5.12.

Table 5.13 – Results for real context instance with 122 customers

Instance			Current solution		Average HA-SR				Best HA-SR			
|H|	|P|	Vehicles	Cost	Vehicles	Cost	Time(s)	Vehicles	Deviation(%)	Cost	Time(s)	Vehicles	Cost(%)
122	5	8	1172.42	8	1043.68	7030.61	7.4	-12.34	1035.96	7027.77	7.2	-13.17

Source: Authors.

The results show that the ILS generate solutions that improve the current one used by Coopservice. The ILS approach found solutions at an average cost of 1043.68, representing a deviation of -12.34% compared to the current solution cost. Regarding the best solution, the ILS approach found a cost of 1035.96, which represents a deviation of -13.17% compared to the current solution cost.

Compared to the average results, HA-SR generates better results than the HA with a cost deviation of -1.31% and a time deviation of -37.97%. Thus, the HA-SR proves to be an appropriate approach also in instances of real context. The illustrations of the routes generated for each period of the best solution found are made available on Appendix B. Detailed results for the best solution found by HA-SR are reported in Table 5.14.

Table 5.14 – Best results for real context instance with 122 customers

Instance			HA-SR	
Period	|H|	Vehicles	Cost	Vehicles
1	56	8	972.41	7
2	55	8	1059.97	7
3	79	8	1148.91	8
4	55	8	948.4	7
5	53	8	1049.61	7

Source: Authors.

We can observe that for four periods, the solutions found used seven vehicles. Considering the cost of drivers and gasoline in the European Union, the current transport by truck for health logistics is around 0.90 euro/km. In the solutions found by HA-SR, the average distance covered is 1,035.86 kilometers per period. If this distance were equally distributed among the eight vehicles available for each period, the average distance traveled by each vehicle would be 129.48 kilometers. Thus, each vehicle would spend an average of 116.53 euros, resulting in an average of 932.26 euros per period. If we consider the five periods, the total value of the solution using all vehicles available would be around 4,661.28 euros. The subtraction of four vehicles in the routing plan can generate savings of around 466.12 euros on the week plan.

6 CONCLUSION

We studied the *Time Windows Assignment Vehicle Routing Problem* (TWAVRP), a VRP variant that appears when the volume of customer demands is uncertain and visits over multiple days should be planned. The objective is to create routes that minimize expected travel costs, assigning a time window overall scenarios to each customer, and respecting the vehicle capacity. Our interest in this problem derives from a real-world case study. We decided to begin our research with the development of a hybrid heuristic and test it on the benchmark TWAVRP instances to check if good-quality solutions can be found within reasonable computational efforts.

To this aim, we proposed an *Iterated Local Search* (ILS) algorithm that generates a pool of feasible routes for each scenario, and a mathematical model, called *Route Selector Model* (RSM), that chooses the most appropriate routes, among those created, minimizing total costs and indicate the time windows for the customers. We compared the results of our algorithm (ILS+RSM, called HA) with the Branch-and-Cut (B&C) proposed by Dalmeijer and Spliet (2018). The HA presented competitive results, concerning both solution quality and computational effort. Concerning large-sized instances with 45 and 50 customers, HA-SR emerges as the best approach, finding a solution there is an average deviation of -0.02% for instances with 45 customers and -1.45% for instances with 50 customers. The greatest effectiveness of HA-SR is due to the fact that it has generated a more varied route pool in relation to HA.

In order to evaluate the behavior of the proposed algorithms in a real context of application of Coopservice, we propose an approach able to estimating the service time of each of the 122 customers. From the computational experiments, HA proved be a competitive method, generating better solutions than those used by the company and obtaining savings of around 466.12 euros per week.

To approach the quickly convergence of the RSM to the incumbent solution, we propose a method to find good solution values by invoking ILS + RSM multiple times. The method did not generate good results. Due to the rapid execution of ILS + RSM in each iteration, the diversity of routes in the pool and the solutions found are directly affected.

Future avenues concern: (i) incorporating new realistic constraints deriving from the real-world case study in the metaheuristic, approaching the possibility of more than one stop during the route, respecting the total rest time. (ii) test metaheuristics based on neighborhoods of other natures, such as evolutionary algorithms, as a route generator. In this way, it is possible to analyze how metaheuristics can influence the solution of the problem; (iii) propose strategies to build a more efficient route pool using the HA method. Strategies such as maintaining the best solutions at each iteration, excluding a percentage of the worst solutions, among others, can improve the performance of the algorithm since they are directly associated with the size of the pool of routes.

BIBLIOGRAPHY

AMORIM, P. et al. A rich vehicle routing problem dealing with perishable food: A case study. **TOP**, v. 22, 2014.

AVCI, M.; TOPALOGLU, S. A multi-start iterated local search algorithm for the generalized quadratic multiple knapsack problem. **Computers & Operations Research**, v. 83, p. 54 – 65, 2017. ISSN 0305-0548.

AZI, N.; GENDREAU, M.; POTVIN, J. An exact algorithm for a vehicle routing problem with time windows and multiple use of vehicles. **European Journal of Operational Research**, v. 202, n. 3, p. 756–763, 2010.

BEHESHTI, A. K.; HEJAZI, S. R.; ALINAGHIAN, M. The vehicle routing problem with multiple prioritized time windows: A case study. **Computers & Industrial Engineering**, v. 90, p. 402 – 413, 2015.

CAMPELO, P. et al. Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. **European Journal of Operational Research**, v. 273, n. 1, p. 131–145, 2019.

CESCHIA, S.; GASPERO, L. D.; SCHAERF, A. Tabu search techniques for the heterogeneous vehicle routing problem with time windows and carrier-dependent costs. **Journal of Scheduling**, v. 14, n. 6, p. 601–615, 2011.

CESELLI, A.; RIGHINI, G.; TRESOLDI, E. Combined location and routing problems for drug distribution. **Discrete Applied Mathematics**, v. 165, p. 130–145, 2014.

COOPSERVICE. **Coopservice**. 2018. <https://www.coopservice.it/>. Accessed: 2019-05-10.

CORDEAU, J.-F.; LAPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. **Journal of the Operational Research Society**, Palgrave Macmillan UK, v. 52, n. 8, p. 928–936, 2001.

DALMEIJER, K.; SPLIET, R. A branch-and-cut algorithm for the time window assignment vehicle routing problem. **Computers & Operational Research**, v. 89, p. 140–152, 2018.

DESROCHERS, M.; DESROSIERS, J.; SOLOMON, M. A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. **Operations Research**, v. 40, n. 2, p. 342–354, 1992.

FACHINI, R. F.; ARMENTANO, V. A. Um algoritmo de programação dinâmica para o problema do caixeiro viajante com janelas de tempo flexíveis. Simpósio Brasileiro de Pesquisa Operacional, p. 1739–1750, 2017.

FACHINI, R. F.; ARMENTANO, V. A. Exact and heuristic dynamic programming algorithms for the traveling salesman problem with flexible time windows. **Optimization Letters**, Oct 2018. ISSN 1862-4480.

FACHINI, R. F.; ARMENTANO, V. A. Logic-based benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. **Computers Industrial Engineering**, p. 106641, 2020. ISSN 0360-8352.

FEILLET, D. et al. A new consistent vehicle routing problem for the transportation of people with disabilities. **Networks**, v. 63, n. 3, p. 211–224, 2014.

FERREIRA, H. S. et al. Variable neighborhood search for vehicle routing problem with multiple time windows. **Electronic Notes in Discrete Mathematics**, v. 66, p. 207–214, 2018.

FIGLIOZZI, M. A. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. **Transportation Research Part C: Emerging Technologies**, v. 18, n. 5, p. 668 – 679, 2010.

GROËR, C.; GOLDEN, B. L.; WASIL, E. A. The consistent vehicle routing problem. **Manufacturing & Service Operations Management**, v. 11, n. 4, p. 630–643, 2009.

GUNAWAN, A.; LAU, H. C.; LU, K. An iterated local search algorithm for solving the orienteering problem with time windows. In: OCHOA, G.; CHICANO, F. (Ed.). **Evolutionary Computation in Combinatorial Optimization**. Cham: Springer International Publishing, 2015. p. 61–73.

HADDADENE, S. R. A.; LABADIE, N.; PRODHON, C. A GRASP x ILS for the vehicle routing problem with time windows, synchronization and precedence constraints. **Expert Systems with Applications**, v. 66, p. 274 – 294, 2016. ISSN 0957-4174.

KESKIN, M.; ÇATAY, B. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. **Computers & Operational Research**, v. 100, p. 172–188, 2018.

KOVACS, A. A. et al. The generalized consistent vehicle routing problem. **Transportation Science**, v. 49, n. 4, p. 796–816, 2015.

KOVACS, A. A.; PARRAGH, S. N.; HARTL, R. F. A template-based adaptive large neighborhood search for the consistent vehicle routing problem. **Networks**, v. 63, n. 1, p. 60–81, 2014.

KOVACS, A. A.; PARRAGH, S. N.; HARTL, R. F. The multi-objective generalized consistent vehicle routing problem. **European Journal of Operational Research**, v. 247, p. 441–458, 2015.

KRAMER, R.; CORDEAU, J.-F.; IORI, M. A rich vehicle routing problem for pharmaceutical distribution. **Seventh International Workshop on Freight Transportation and Logistics**, 2018.

LIU, R. et al. Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. **European Journal of Operational Research**, v. 230, n. 3, p. 475–486, 2013.

LIU, R.; XIE, X.; GARAIX, T. Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. **Omega**, v. 47, p. 17 – 32, 2014.

LÓPEZ-IBÁÑEZ, M. et al. The irace package: Iterated racing for automatic algorithm configuration. **Operations Research Perspectives**, Elsevier, v. 3, p. 43–58, 2016.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: **Handbook of metaheuristics**. [S.l.]: Springer, 2003. p. 320–353.

MAGALHÃES, J. M.; SOUSA, J. P. de. Dynamic VRP in pharmaceutical distribution - a case study. **Central European Journal of Operational Research**, v. 14, n. 2, p. 177–192, 2006.

MARTINS, S. et al. An optimization-simulation approach to the network redesign problem of pharmaceutical wholesalers. **Computers & Industrial Engineering**, v. 106, p. 315 – 328, 2017.

MOREIRA, M. C. O.; COSTA, A. M. Hybrid heuristics for planning job rotation schedules in assembly lines with heterogeneous workers. **International Journal of Production Economics**, Elsevier, v. 141, n. 2, p. 552–560, 2013.

NEVES-MOREIRA, F. et al. The time window assignment vehicle routing problem with product dependent deliveries. **Transportation Research Part E: Logistics and Transportation Review**, v. 116, p. 163 – 183, 2018.

NOGUEIRA, B.; PINHEIRO, R. G. S.; SUBRAMANIAN, A. A hybrid iterated local search heuristic for the maximum weight independent set problem. **Optimization Letters**, v. 12, n. 3, p. 567–583, May 2018. ISSN 1862-4480.

OMBUKI, B. M.; ROSS, B.; HANSHAR, F. Multi-objective genetic algorithms for vehicle routing problem with time windows. **Applied Intelligence**, v. 24, n. 1, p. 17–30, 2006.

SETTANNI, E.; HARRINGTON, T. S.; SRAI, J. S. Pharmaceutical supply chain models: A synthesis from a systems view of operations research. **Operations Research Perspectives**, v. 4, p. 74 – 95, 2017. ISSN 2214-7160.

SPLIET, R.; DABIA, S.; WOENSEL, T. V. The time window assignment vehicle routing problem with time-dependent travel times. **Transportation Science**, v. 52, n. 2, p. 261–276, 2018.

SPLIET, R.; DESAULNIERS, G. The discrete time window assignment vehicle routing problem. **European Journal of Operational Research**, v. 244, n. 2, p. 379–391, 2015.

SPLIET, R.; GABOR, A. F. The time window assignment vehicle routing problem. **Transportation Science**, v. 49, n. 4, p. 721–731, 2015.

STAVROPOULOU, F.; REPOUSSIS, P.; TARANTILIS, C. The vehicle routing problem with profits and consistency constraints. **European Journal of Operational Research**, v. 274, n. 1, p. 340 – 356, 2019.

STÜTZLE, T.; RUIZ, R. Iterated local search. In: ____. **Handbook of Heuristics**. [S.l.]: Springer International Publishing, 2018. p. 579–605. ISBN 978-3-319-07124-4.

SUBRAMANIAN, A.; CABRAL, L. dos A. F. An ILS based heuristic for the vehicle routing problem with simultaneous pickup and delivery and time limit. In: HEMERT, J. van; COTTA, C. (Ed.). **Evolutionary Computation in Combinatorial Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 135–146. ISBN 978-3-540-78604-7.

TARANTILIS, C. D.; STAVROPOULOU, F.; REPOUSSIS, P. P. A template-based tabu search algorithm for the consistent vehicle routing problem. **Expert System Application**, v. 39, n. 4, p. 4233–4239, 2012.

TAŞ, D. et al. Vehicle routing problem with stochastic travel times including soft time windows and service costs. **Computers & Operational Research**, v. 40, n. 1, p. 214–224, 2013.

TAŞ, D.; JABALI, O.; Van Woensel, T. A Vehicle Routing Problem with Flexible Time Windows. **Computers & Operations Research**, v. 52, p. 39–54, 2014.

TOTH, P.; VIGO, D. **Vehicle routing : problems, methods, and applications**. 2nd edition. ed. [S.l.]: SIAM, 2014. 463 p. ISBN 9781611973587.

ZHIGALOV, G. **VRPTW**. [S.l.]: GitHub, 2018. GitHub repository: <https://github.com/donfaq/VRPTW>, last commit: 0261f086d1610c2da3152540d0e535a4eaeef76c.

A DETAILED RESULTS FOR THE LITERATURE INSTANCES

In this appendix we make available the detailed results for the instances that were discussed on aggregated form on Section 5 from the tables:

- Table A.1 reports the detailed results for literature instances by Dalmeijer and Spliet (2018) obtained by the HA-SR.
- Table A.2 reports the detailed results for literature instances by Dalmeijer and Spliet (2018) obtained by the HA.

Table A.1 – HA-SR detailed results for literature instances by Dalmeijer and Spliet (2018)

Instance		B & C		HA-SR		
#	N. customers	Opt. cost	Time(s)	Cost	Time(s)	Gap (%)
1	10	17.65	0	17.65	9.08	0.00
2	10	15.56	0.1	15.89	10.53	2.10
3	10	17.42	0	17.42	11.83	0.00
4	10	18.51	0.1	18.51	9.24	0.00
5	10	16.07	0.3	16.5	8.74	2.61
6	10	18	0	18	9.06	0.00
7	10	17.02	0	17.02	9.61	0.00
8	10	23.89	0.1	24.23	9.36	1.39
9	10	20.31	0	20.57	9.8	1.28
10	10	16.31	0	16.31	5.67	0.00
11	15	17.78	0	17.78	29.1	0.00
12	15	27.1	39.1	27.19	38.09	0.32
13	15	29.37	2.6	29.55	28.29	0.61
14	15	23.18	0.2	23.31	33.7	0.54
15	15	24.15	0.6	24.15	26.86	0.00
16	15	21.03	0.3	21.05	30.21	0.09
17	15	22.04	0.1	22.04	34.7	0.00
18	15	22.3	0.8	22.46	55.14	0.71
19	15	26.52	1.3	26.73	30.52	0.77
20	15	22.11	0.4	22.16	25.87	0.23
21	20	28.08	1.2	28.09	103.12	0.05
22	20	29.8	9	29.8	64.72	0.00
23	20	30.3	0.4	30.33	61.07	0.11
24	20	24.16	1.7	24.27	66.9	0.45
25	20	29.84	6.9	29.86	78.36	0.08
26	20	29.72	0.2	29.72	51.69	0.01
27	20	26.48	0.3	26.62	68.64	0.51
28	20	26.14	1.1	26.45	78.77	1.16
29	20	26.61	0.5	26.65	65.64	0.15
30	20	26.36	0.3	26.36	79.62	0.00
31	25	31.43	2.3	31.6	116.21	0.55
32	25	30.71	1.3	30.95	118.2	0.76
33	25	33.71	9.4	33.78	137.18	0.22
34	25	33.34	11.1	33.34	137.88	0.01
35	25	29.05	6.1	29.1	132.1	0.18
36	25	30.5	39.2	30.65	139.96	0.48
37	25	28.68	22.4	28.88	121.49	0.69
38	25	35.69	9.7	36.02	141.07	0.92
39	25	32.55	7.2	32.75	126.46	0.60
40	25	32.14	15.2	32.17	82.66	0.08
41	30	36.38	137	36.39	204.6	0.04
42	30	34.74	3600	34.83	243.78	0.27
43	30	35.48	187.6	35.66	209.66	0.50
44	30	35.88	60.6	36.01	149.1	0.35
45	30	35.55	110.8	35.66	185.59	0.30
46	30	37.47	7.7	37.82	204.8	0.93
47	30	32.54	17.8	32.89	251.69	1.06
48	30	36.32	357.9	36.63	249	0.85
49	30	35.3	930.3	35.4	202.96	0.28
50	30	40.27	30.7	40.47	196.07	0.49
51	35	43.46	18.2	43.67	559	0.49
52	35	41.84	14	41.98	423.07	0.33
53	35	45.14	3600	45.36	298.42	0.48
54	35	41.57	3600	41.84	452.29	0.65
55	35	37.92	68.5	37.94	335.3	0.05
56	35	44.49	3600	44.53	338.52	0.08
57	35	40.83	3600	41.28	475.85	1.09
58	35	41.22	127.9	41.22	435.33	0.00
59	35	43.43	245.1	43.65	381.44	0.51
60	35	42.27	443.3	42.63	298.23	0.84
61	40	46.35	3600	46.42	577.57	0.15
62	40	48.35	550.3	48.49	606.81	0.29
63	40	44.48	3600	44.78	472.4	0.68
64	40	43.75	3169.7	43.91	512.37	0.37
65	40	43.46	3600	43.45	620.59	-0.02
66	40	44.68	3600	44.82	442.94	0.32
67	40	46.96	3600	47.57	550.44	1.28
68	40	45.02	3600	45.12	322.4	0.22
69	40	43.2	3600	43.26	459.4	0.14
70	40	43	3600	43.47	502.79	1.09
71	45	51.78	3600	51.34	786.74	-0.85
72	45	52.13	3600	51.93	678.14	-0.39
73	45	41.7	3600	42.09	835.15	0.92
74	45	47.84	3600	47.95	837.96	0.23
75	45	49.86	3600	49.82	533.35	-0.07
76	45	52.09	3600	50.25	1811.73	-3.65
77	45	51.18	3600	51.53	454.25	0.67
78	45	53.95	3600	53.66	613.01	-0.54
79	45	48.21	3600	48.32	735.09	0.23
80	45	50.57	3600	50.4	613.71	-0.33
81	50	58.85	3600	58.31	613.63	-0.93
82	50	53.2	3600	53.02	1020.14	-0.35
83	50	60.67	3600	58.86	1149.4	-3.07
84	50	56.38	3600	54.3	979.57	-3.83
85	50	56.07	3600	55.22	1190.06	-1.53
86	50	54.76	3600	53.37	856.5	-2.60
87	50	54.14	3600	53.88	784.61	-0.48
88	50	56.91	3600	56.44	1022.95	-0.84
89	50	61.51	3600	60.53	914.08	-1.61
90	50	59.55	3600	59.17	781.14	-0.64

Source: Authors.

Table A.2 – HA detailed results for literature instances by Dalmeijer and Spliet (2018)

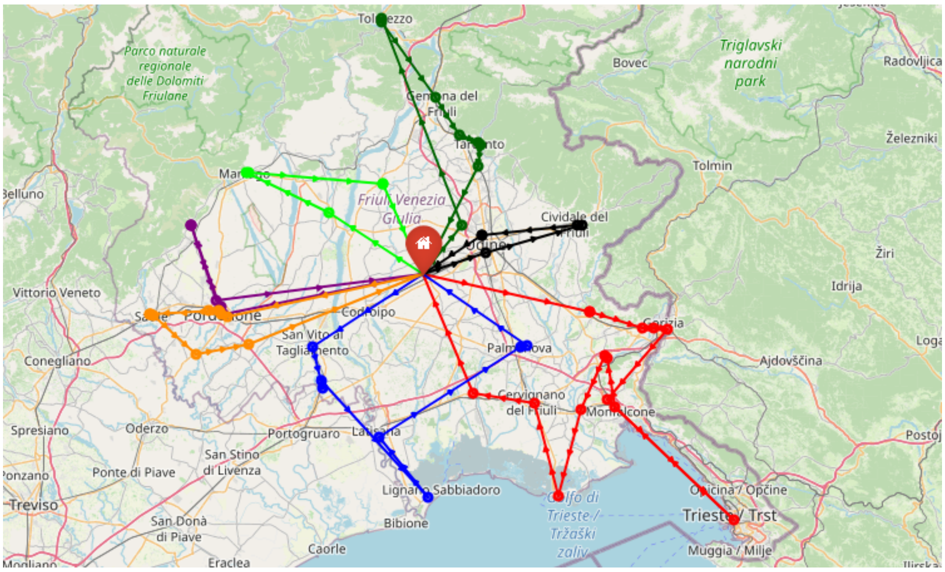
Instance		B & C		HA		
#	N. customers	Opt. Value	Opt. Time	Value	Total time	Gap (%)
1	10	17.65	0	17.65	21.96	0.00
2	10	15.56	0.1	15.89	21.71	2.10
3	10	17.42	0	17.42	23.88	0.00
4	10	18.51	0.1	18.51	22.59	0.00
5	10	16.07	0.3	16.41	25.06	2.09
6	10	18	0	18.00	21.03	0.00
7	10	17.02	0	17.02	19.65	0.00
8	10	23.89	0.1	24.23	23.00	1.39
9	10	20.31	0	20.57	20.48	1.28
10	10	16.31	0	16.31	15.76	0.00
11	15	17.78	0	17.78	49.54	0.00
12	15	27.1	39.1	27.34	67.13	0.87
13	15	29.37	2.6	29.55	64.20	0.61
14	15	23.18	0.2	23.19	70.11	0.03
15	15	24.15	0.6	24.26	57.72	0.47
16	15	21.03	0.3	21.05	58.57	0.09
17	15	22.04	0.1	22.04	60.17	0.00
18	15	22.3	0.8	22.42	71.40	0.55
19	15	26.52	1.3	26.80	52.81	1.04
20	15	22.11	0.4	22.16	68.03	0.23
21	20	28.08	1.2	28.09	146.04	0.05
22	20	29.8	9	29.80	134.63	-0.01
23	20	30.3	0.4	30.33	118.48	0.11
24	20	24.16	1.7	24.32	151.06	0.67
25	20	29.84	6.9	29.86	143.56	0.08
26	20	29.72	0.2	29.72	120.21	0.01
27	20	26.48	0.3	26.62	124.99	0.51
28	20	26.14	1.1	26.51	133.38	1.38
29	20	26.61	0.5	26.65	118.51	0.15
30	20	26.36	0.3	26.36	109.32	0.00
31	25	31.43	2.3	31.56	258.28	0.41
32	25	30.71	1.3	30.99	237.53	0.90
33	25	33.71	9.4	33.78	206.75	0.22
34	25	33.34	11.1	33.34	291.36	0.01
35	25	29.05	6.1	29.10	257.01	0.18
36	25	30.5	39.2	30.57	253.30	0.23
37	25	28.68	22.4	28.88	256.51	0.69
38	25	35.69	9.7	36.10	312.48	1.14
39	25	32.55	7.2	32.75	275.68	0.60
40	25	32.14	15.2	32.17	271.82	0.08
41	30	36.38	137	36.39	430.60	0.04
42	30	34.74	3600	34.82	555.09	0.24
43	30	35.48	187.6	35.74	506.96	0.72
44	30	35.88	60.6	35.99	474.37	0.31
45	30	35.55	110.8	35.65	368.57	0.27
46	30	37.47	7.7	37.78	499.15	0.83
47	30	32.54	17.8	32.89	416.04	1.06
48	30	36.32	357.9	36.45	575.65	0.37
49	30	35.3	930.3	35.40	580.78	0.28
50	30	40.27	30.7	40.48	455.08	0.53
51	35	43.46	18.2	43.64	690.27	0.42
52	35	41.84	14	42.02	713.93	0.44
53	35	45.14	3600	45.35	1008.33	0.47
54	35	41.57	3600	41.75	743.78	0.42
55	35	37.92	68.5	37.95	761.49	0.08
56	35	44.49	3600	44.53	905.96	0.08
57	35	40.83	3600	41.28	722.29	1.09
58	35	41.22	127.9	41.23	847.74	0.03
59	35	43.43	245.1	43.59	925.83	0.37
60	35	42.27	443.3	42.51	865.45	0.56
61	40	46.35	3600	46.39	1216.42	0.09
62	40	48.35	550.3	48.47	1114.95	0.24
63	40	44.48	3600	44.65	1149.48	0.39
64	40	43.75	3169.7	43.90	1147.18	0.33
65	40	43.46	3600	43.47	1190.26	0.03
66	40	44.68	3600	44.82	1133.07	0.31
67	40	46.96	3600	47.80	1278.87	1.76
68	40	45.02	3600	45.16	1154.60	0.31
69	40	43.2	3600	43.27	1164.02	0.16
70	40	43	3600	43.52	1138.33	1.19
71	45	51.78	3600	51.43	1161.07	-0.67
72	45	52.13	3600	52.04	1145.32	-0.18
73	45	41.7	3600	42.17	1119.53	1.11
74	45	47.84	3600	48.22	1254.34	0.78
75	45	49.86	3600	49.81	1119.96	-0.10
76	45	52.09	3600	50.35	1138.68	-3.45
77	45	51.18	3600	51.63	1247.06	0.87
78	45	53.95	3600	53.72	1112.59	-0.42
79	45	48.21	3600	48.40	1185.60	0.39
80	45	50.57	3600	50.69	1215.56	0.23
81	50	58.85	3600	58.50	1267.12	-0.60
82	50	53.2	3600	53.20	1385.75	0.00
83	50	60.67	3600	58.97	1245.45	-2.88
84	50	56.38	3600	54.33	1279.79	-3.77
85	50	56.07	3600	55.51	1262.58	-1.00
86	50	54.76	3600	53.48	1290.54	-2.39
87	50	54.14	3600	53.98	1358.15	-0.30
88	50	56.91	3600	56.63	1277.59	-0.49
89	50	61.51	3600	61.22	1431.15	-0.47
90	50	59.55	3600	59.29	1125.81	-0.44

Source: Authors.

B SCENARIOS ILLUSTRATION FOR THE BEST SOLUTION

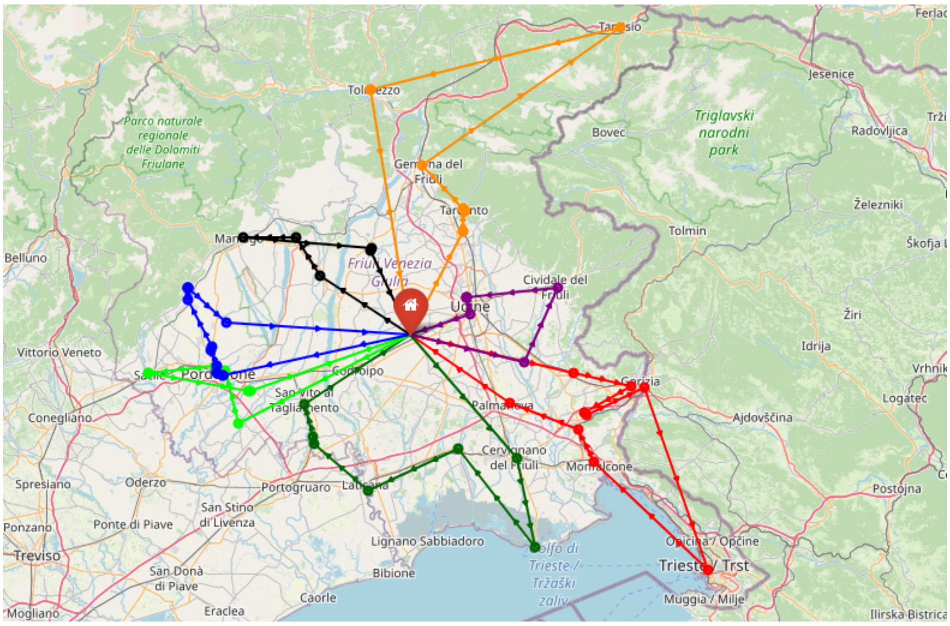
In this appendix we make available the illustrations of the routes generated for each period of the best solution discussed on Section 5.

Figure B.1 – Routes for period 1.



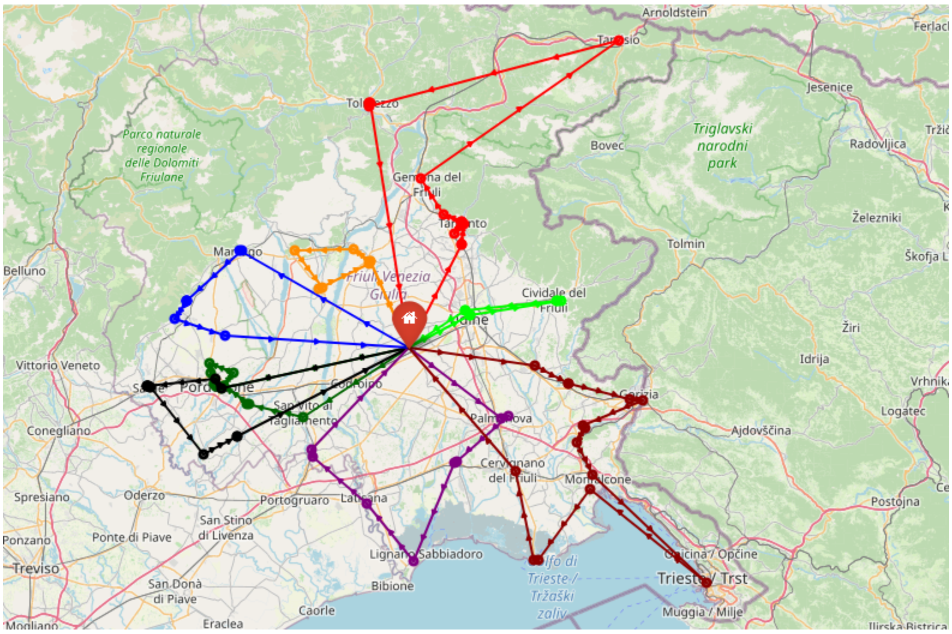
Source: Authors with the library *Folium*.

Figure B.2 – Routes for period 2.



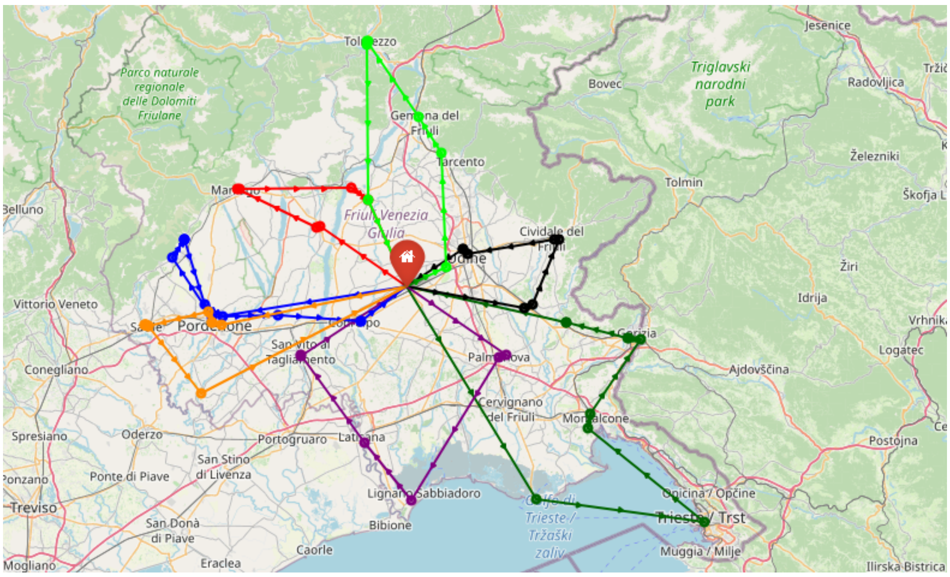
Source: Authors with the library *Folium*.

Figure B.3 – Routes for period 3.



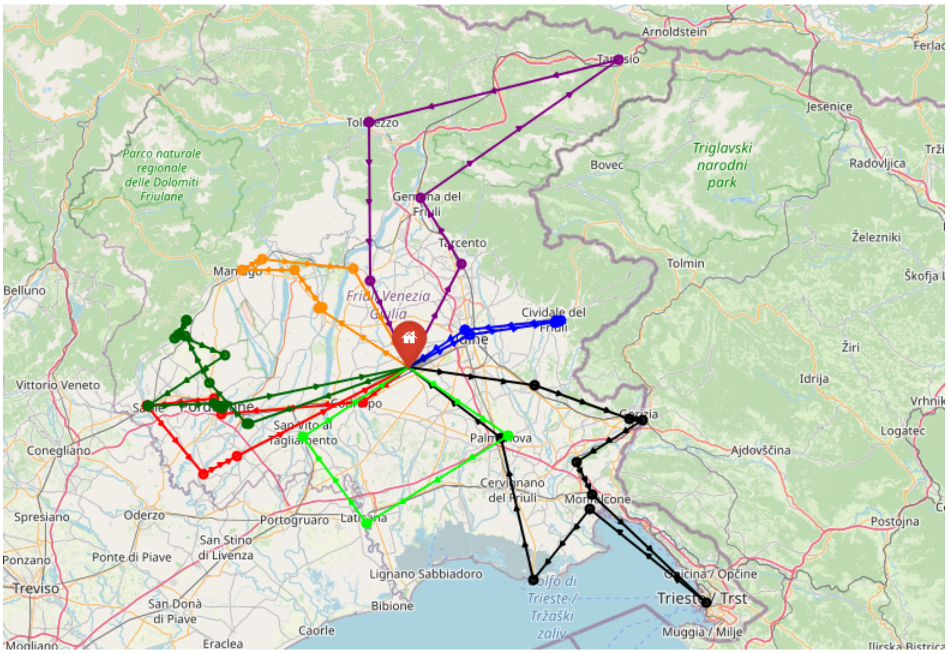
Source: Authors with the library *Folium*.

Figure B.4 – Routes for period 4.



Source: Authors with the library *Folium*.

Figure B.5 – Routes for period 5.



Source: Authors with the library *Folium*.