



Matheus Oliveira Meirim

**Vehicle Routing Problem with Occasional
Drivers for E-Commerce Last-Mile Delivery: A
Metaheuristic Approach**

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-graduação em Engenharia de Produção of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção.

Advisor: Prof. Rafael Martinelli Pinto

Rio de Janeiro
August 2023



Matheus Oliveira Meirim

**Vehicle Routing Problem with Occasional
Drivers for E-Commerce Last-Mile Delivery: A
Metaheuristic Approach**

Dissertation presented to the Programa de Pós-graduação em Engenharia de Produção da PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia de Produção. Approved by the Examination Committee:

Prof. Rafael Martinelli Pinto

Advisor

Departamento de Engenharia Industrial – PUC-Rio

Prof. Claudio Contardo Vera

Concordia University

Prof. Olivier Gallay

HEC Lausanne

Rio de Janeiro, August 17th, 2023

All rights reserved.

Matheus Oliveira Meirim

Has degree in Production Engineering from Pontifical Catholic University of Rio de Janeiro (PUC-Rio, 2021). Currently pursuing a Master's degree in Production Engineering at PUC-Rio, focusing on Operations Research with an emphasis on Algorithms and Optimization. Currently working as a Software Developer at Americanas S.A..

Bibliographic data

Meirim, Matheus Oliveira

Vehicle Routing Problem with Occasional Drivers for E-Commerce Last-Mile Delivery: A Metaheuristic Approach / Matheus Oliveira Meirim; advisor: Rafael Martinelli Pinto. – 2023.

55 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, 2023.

Inclui bibliografia

1. Engenharia Industrial – Teses. 2. Roteamento de veículos com motoristas ocasionais. 3. Iterated Local Search. 4. Metaheurísticas. 5. Entregas de e-commerce. 6. Last-mile. I. Pinto, Rafael Martinelli. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Industrial. III. Título.

CDD: 658.5

To my parents, girlfriend, family, and
friends for their support and encouragement.

Acknowledgments

I would like to thank my supervisor, Professor Rafael Martinelli for providing guidance and feedback throughout this project.

I am profoundly thankful to God for granting me the strength, wisdom, and perseverance to complete this master's dissertation. Thanks because You have showed the way, illuminating the path toward knowledge and understanding.

Thanks also to my fiancé Sara for all of your help, encouragement, understanding, and love over the previous two years. Your presence and encouragement have made every challenge easier to overcome. Thank you for being an integral part of my life.

I would like to express my deepest gratitude to my parents and my family for their unwavering support and unwavering belief in me throughout this journey. Your encouragement and presence in my life have provided a sense of belonging and security that is invaluable.

I extend my heartfelt appreciation to my dear friends from the church community for their prayers, encouragement, and uplifting spirit that sustained me during this challenging academic endeavor.

To the professors who served on the examining committee, I extend my sincere gratitude for their valuable time, insightful feedback, and contributions to the evaluation of this dissertation.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Meirim, Matheus Oliveira; Pinto, Rafael Martinelli (Advisor). **Vehicle Routing Problem with Occasional Drivers for E-Commerce Last-Mile Delivery: A Metaheuristic Approach**. Rio de Janeiro, 2023. 55p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

In recent years, e-commerce has become widespread in society, and the logistics of product delivery is a crucial pillar for this market to maintain a high level of service and remain advantageous for consumers choosing to make purchases online. The present work aims to study the problem of last-mile vehicle routing for e-commerce deliveries and apply an Iterated Local Search (ILS) metaheuristic to optimize the routing of parcels in a Brazilian e-commerce company. With the objective of finding routes with the lowest cost for the deliveries, this study proposes an extension to the Vehicle Routing Problem with Occasional Drivers (VRPOD), considering a heterogeneous fleet and occasional drivers handling multiple deliveries. For the methodology application, data provided by an e-commerce company are used, and they are properly anonymized to prevent the identification of the company and its clients, respecting ethical principles. A total of 121 instances are used, ranging from the smallest with one vertex to the largest with 344. The results of the proposed model are presented in two scenarios: firstly, considering routing without the use of occasional drivers, and secondly, considering the availability of occasional drivers for some routes. Both scenarios are compared with the routes generated by the current router used in the company, and preliminary results indicate that without the use of occasional drivers, the proposed ILS obtains better solutions in 53.72% of the instances, and when occasional drivers are incorporated into the route, improvements occur in 76.03% of the instances. The utilization of occasional drivers also provides a 10.30% reduction in the average routing cost.

Keywords

Vehicle Routing with Occasional Drivers; Iterated Local Search; Metaheuristics; E-commerce delivery; Last-mile.

Resumo

Meirim, Matheus Oliveira; Pinto, Rafael Martinelli. **Problema de Roteamento de Veículos com Motoristas Ocasioneis para Entregas de Last-Mile: Uma Abordagem Meta-heurística**. Rio de Janeiro, 2023. 55p. Dissertação de Mestrado – Departamento de Engenharia Industrial, Pontifícia Universidade Católica do Rio de Janeiro.

Nos últimos anos o comércio eletrônico tem se difundido na sociedade e a logística de entrega dos produtos é um dos pilares para que este mercado mantenha o nível de serviço alto e continue sendo vantajoso para o consumidor decidir por realizar a compra pela internet. O presente trabalho se destina a estudar sobre o problema de roteamento de veículos de entrega *last-mile* para e-commerce e aplicar a metaheurística *Iterated Local Search (ILS)* visando otimizar o roteamento do trecho *last-mile* de encomendas realizadas em uma empresa de comércio eletrônico brasileira. Com o objetivo de encontrar rotas de menor custo para as entregas a serem realizadas, este trabalho propõe uma extensão para o *Vehicle Routing Problem With Occasional Drivers (VRPOD)*, considerando frota heterogênea e motoristas ocasionais realizando o transporte de mais de uma entrega. Para a aplicação do método foram utilizados dados fornecidos por uma empresa de e-commerce que foram devidamente anonimizados de forma a não ser possível identificar a empresa e nem os clientes, respeitando os princípios éticos. Foram utilizadas 121 instâncias, sendo a menor com um vértice e a maior com 344. Os resultados do modelo proposto são apresentados em dois cenários, primeiramente considerando que o roteamento é realizado sem a utilização de motoristas ocasionais. O segundo cenário considera a disponibilização de motoristas ocasionais para serem utilizados em algumas rotas. Ambos os cenários foram comparados com as rotas geradas pelo roteador existente hoje na companhia e os resultados preliminares indicam que o sem a utilização de motoristas ocasionais o ILS proposto obtém melhores soluções em 53.72% das instâncias e quando os motoristas ocasionais são incorporados a rota ocorre melhoria em 76.03% das instâncias utilizadas. A utilização de motoristas ocasionais também proporciona uma redução de 10.30% no custo médio de roteamento.

Palavras-chave

Roteamento de veículos com motoristas ocasionais; Iterated Local Search; Metaheurísticas; Entregas de e-commerce; Last-mile.

Table of contents

1	Introduction	13
1.1	Research Problem and Motivation	14
1.2	Objectives	15
1.3	Thesis Contributions and Relevance	15
1.4	Thesis Research Methods	16
2	Theoretical framework	18
2.1	The Vehicle Routing Problem With Occasional Drivers	18
2.2	Literature Review	19
2.3	Solution Methods	21
2.4	Mathematical Formulation	23
3	Iterated Local Search for the Vehicle Routing Problem With Occasional Drivers	26
3.1	Solution Representation	26
3.2	Constructive Algorithm	27
3.3	Local Search	31
3.4	Perturbation	34
3.5	Acceptance Criterion	35
4	Numerical Experiments	37
4.1	Case study	37
4.2	Data Acquiring and treatment	38
4.3	Experiments	39
5	Conclusion	43
6	Bibliography	45
A	Complete Results	48

List of figures

Figure 3.1	Solution Representation	26
Figure 3.2	Example Route	31
Figure 3.3	A set of three subfigures: (a) insert move; (b) remove move; (c) shift move.	31
	(a) insert	31
	(b) remove	31
	(c) shift	31
Figure 3.4	A set of two subfigures: (a) swap intra-route move; (b) swap inter-route move.	32
	(a) swap intra-route	32
	(b) swap inter-route	32
Figure 3.5	A set of two subfigures: (a) relocate intra-route move; (b) relocate inter-route move.	32
	(a) relocate intra-route	32
	(b) relocate inter-route	32
Figure 4.1	Company supply-chain Adapted from (ZHOU et al., 2018)	38
Figure 4.2	Average number of customers attended and costs comparison between ILS and company Router	40
Figure 4.3	Computational time in seconds by the number of vertices in the instance	41
Figure 4.4	Average number of customers attended and costs comparison between ILS with occasional drivers, ILS without occasional drivers and company Router	42

List of tables

Table A.1	Iterated Local Search with NearestNeighbor Constructive	48
Table A.2	Iterated Local Search with NearestNeighbor Constructive (Continuation)	49
Table A.3	Iterated Local Search with NearestCheaperNeighbor Constructive	50
Table A.4	Iterated Local Search with NearestCheaperNeighbor Constructive (Continuation)	51
Table A.5	Iterated Local Search with Occasional Drivers prioritization	52
Table A.6	Iterated Local Search with Occasional Drivers prioritization (Continuation)	53
Table A.7	Iterated Local Search with Occasional Drivers not prioritized	54
Table A.8	Iterated Local Search with Occasional Drivers not prioritized (Continuation)	55

List of Abbreviations

API – *Application programming interface*

GRASP – *Greedy randomized adaptive search procedure*

ILS – *Iterated Local Searches*

OD – *Occasional driver*

SA — *Simulated Annealing*

VND — *Variable Neighborhood Descent*

VNS – *Variable neighborhood search*

VRP — *Vehicle Routing Problem*

VRPOD – *Vehicle Routing Problem with Occasional Drivers*

VRPODTWmd – *Vehicle Routing Problem with Occasional Drivers and Time Windows*

VRPPC – *Vehicle Routing Problem with a Private Fleet and Common Carrier*

TS – *Tabu Search*

*Come unto me, all you that labor and are
heavy laden, and I will give you rest.*

Jesus Christ, Mt 11, 28.

1

Introduction

The increasing technological development that has occurred in recent decades has brought about changes in the consumption habits of the global population, and one of these changes is the rise of e-commerce. According to a report by Nielsen (2022), the revenue of Brazilian e-commerce grew by 27% in 2021 compared to the previous year, and the number of orders increased by 16.9% in the same period, compared to 2020 (PAGAR.ME, 2022). Among the benefits provided by e-commerce, Colla and Lapoule (2012) highlight that online shopping is popular among consumers seeking to reduce the time spent in the purchasing process, among other conveniences such as the ability to shop anytime and without physical effort.

The delivery time and the freight cost are factors that directly impact the consumer's perception of whether buying through digital means is a good deal. Therefore, companies today seek to balance low costs and delivery speed. The report prepared by Nielsen (2022) indicates that in 2021, there was a 10% increase in the number of orders with free shipping compared to 2020, accounting for 47% of all e-commerce orders that year. With this established scenario, logistics has become a key differentiator for e-commerce companies, requiring the provision of high-quality delivery services that can ensure timely deliveries with minimal costs (ÜLKÜ; BOOKBINDER, 2012).

In this context, an increasing number of companies are aiming to establish warehouses closer to end consumers to enable deliveries within urban centers, intensifying discussions on optimizing last-mile deliveries in recent years (BOYSEN; FEDTKE; SCHWERDFEGGER, 2021). In order to optimize deliveries in this last stretch, companies and logistics providers employ routing algorithms that generate routes of minimum cost while respecting the constraints imposed by each scenario. In the literature, this problem is known as the Vehicle Routing Problem (VRP), and its objective is to meet the demands of geographically dispersed consumers from one or more depots using a fleet of vehicles with varying capacities and respecting the imposed constraints (LAPORTE, 1992).

Over the years, as consumer needs and consequently the constraints imposed on logistics providers have changed, new variations of the Vehicle Routing Problem (VRP) have been developed. The Vehicle Routing Problem with Occasional Drivers (VRPOD), proposed by Archetti, Savelsbergh and Speranza (2016), is one such variant that considers the availability of the

company's vehicle fleet and also the availability of drivers who can be called upon to make deliveries using their own vehicles. The use of occasional drivers for deliveries proves to be an interesting option for urban centers, as, in many cases, larger vehicles are underutilized, spending most of their time parked or are not allowed in certain urban centers during specific time windows during the day.

1.1 Research Problem and Motivation

In a market where customers are reluctant to spend high amounts on shipping and value receiving orders in short time frames, the absence of optimized routes results in a loss of money in a context where profit margins are narrow. Consequently, the adoption of algorithms capable of performing vehicle routing while minimizing delivery costs becomes necessary.

In the context of e-commerce, one widely adopted alternative by companies is to divide their logistics chain into multiple axes, allowing for the existence of intermediate warehouses close to major urban centers, capable of efficiently handling this last portion of the delivery. This delivery segment closer to the end consumer is known as the last-mile, and the routing of deliveries in the last mile has been the subject of study for many e-commerce companies and logistics providers seeking to reduce expenses.

The last-mile delivery in large urban centers presents various challenges, necessitating the enhancement of efficiency in this service. In this context, Bachofner et al. (2022) identify several points of concern, such as increased traffic congestion and urban pollution due to an aging vehicle fleet that may not be adapted to the complexities of urban environments.

The e-commerce company that will be the subject of study in this work operates throughout the Brazilian territory, with distribution centers nationwide, warehouses close to some urban centers with higher population densities, and the use of occasional drivers, in addition to a fleet of vehicles for deliveries from these warehouses to customers' homes or collection points for customer pickup.

Therefore, this work proposes an extension of the Vehicle Routing Problem with Occasional Drivers (VRPOD) to perform routing of last-mile deliveries, considering that occasional drivers are capable of transporting more than one package. The study further proposes a metaheuristic aiming to obtain the solution with the lowest cost.

1.2 Objectives

The main objective of this study is to develop a metaheuristic for vehicle routing in the last-mile segment of a Brazilian e-commerce company, aiming to maximize the number of customer served and minimize the total cost of order deliveries while respecting the constraints imposed by the company to ensure no reduction in the level of service provided to the customer.

To achieve the main objective, this study has the following secondary objectives:

- To comprehend the innovations being implemented in vehicle routing in the context of last-mile deliveries.
- To understand the peculiarities of freight transportation in urban areas in Brazil.
- To evaluate the benefits of using occasional drivers for last-mile deliveries in Brazil.
- To compare the routes proposed by the metaheuristic with the routes generated by the existing algorithm in the company, allowing for the assessment of the improvements achieved.

1.3 Thesis Contributions and Relevance

Considering the logistical challenges generated by the increasing adoption of online shopping, it is necessary to understand what has been done in other countries regarding deliveries that occur within urban centers, in the closest kilometers to the end consumer. Such understanding can have a positive impact on e-commerce companies and logistics providers, as it may lead to cost reduction and shorter delivery times, which are important decision factors for consumers. Additionally, it can indirectly benefit the environment by potentially reducing the vehicle fleet and improving the quality of life for urban residents, as the increase in delivery volume contributes to congestion, and optimizations in routes and vehicles can mitigate this problem.

Various variations of the Vehicle Routing Problem (VRP) have been proposed to optimize last-mile deliveries, and the Vehicle Routing Problem with Occasional Drivers (VRPOD) is one of them. Due to the novelty of the concepts of casual drivers and crowd-shipping for last-mile deliveries, there is a limitation in the quantity of studies on this subject. The number of publications becomes even more limited when seeking those that use real-world instances to

validate the developed solution methodologies. Therefore, this work stands out due to the utilization of instances with real data from a Brazilian e-commerce company and the comparison of results between the routes executed and the routes generated by the proposed solution method.

1.4

Thesis Research Methods

1.4.1

Thesis Research Phases

To achieve the previously stated objectives, this study is divided into five stages. The first stage is dedicated to problem characterization and data collection, encompassing the specifics and constraints of the problem, verifying data availability, and identifying data attributes. To accomplish this, contact was made with the e-commerce company, presenting the proposed research work, and seeking information on their current delivery planning methods, including whether the company currently performs vehicle routing for deliveries and how it is conducted. This was done to gain a deeper understanding of the order delivery logistics chain and the business constraints imposed, enabling the development of a solution method that aligns better with the company's reality.

The second stage of this study involves conducting a literature review, examining how academia has addressed the vehicle routing problem for last-mile deliveries. From this review, approaches that are most relevant to the context of the analyzed company were identified, and the chosen solution method for addressing the routing problem in this study was determined.

The third stage is dedicated to data collection and preprocessing. During this stage, the company's professionals assisted in extracting the required data, which was then treated to select only the necessary information for this study and structured to optimize the computational operations to be executed by the optimization algorithm.

In the fourth stage, the proposed metaheuristic was implemented, and computational experiments were conducted using different scenarios based on real data to validate the proposed algorithm in comparison with the model currently used by the company.

The fifth stage involves the analysis of the results obtained from the computational experiments and the discussion of these results, considering each of the scenarios proposed in the previous stage.

1.4.2 Thesis Structure

The following sections of this document are organized into 5 chapters, according to the structure described below. Chapter 2 will provide a literature review to present relevant works on vehicle routing problems, specifically focusing on the vehicle routing problem with occasional drivers. Additionally, the selected method for solving this problem will be introduced.

Chapter 3 addresses a more detailed description of the problem, presenting the constraints and characteristics of the problem to be addressed in this dissertation. It includes the proposal of the selected metaheuristic, the assumptions made, and the solution representation.

Chapter 4 presents the conducted computational experiments, outlining the structure of the instances used and the motivations behind each experiment. In addition, this chapter shows the analysis of the results obtained from the computational experiments, presenting the distinctive features of the proposed model and verifying its efficiency.

Finally, the last chapter presents the final considerations and suggestions for future research.

2

Theoretical framework

In this section a the attributes and a definition with a mathematical model of the problem will be presented. Previous literature is reviewed and the selected solution method is described.

2.1

The Vehicle Routing Problem With Occasional Drivers

Braekers, Ramaekers and Nieuwenhuyse (2016) in their literature review, identify the work developed by Dantzig and Ramser (1959) as the first to address the vehicle routing problem, and since then, other authors have dedicated themselves to studying, developing new models, and identifying variations for this problem. The objective of the vehicle routing problem is to meet the demands of geographically dispersed consumers from one or more depots using a fleet of vehicles with varying capacities (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016). This problem can be applied in various contexts, such as the delivery of medical supplies or military materials, but this dissertation will focus on the context of last-mile deliveries. Each of these contexts presents unique characteristics and constraints, leading to variations in the models and heuristics used to solve them.

A recent variation of the VRP, proposed by Archetti, Savelsbergh and Speranza (2016), is the Vehicle Routing Problem with Occasional Drivers (VRPOD). This variant was created to represent the dynamics of crowd-shipping, which has been used in urban centers to make deliveries of goods. The VRPOD considers that the company has its own fleet of vehicles with a maximum capacity but can also utilize occasional drivers to make the necessary deliveries. In this scenario, it is worth noting that in the VRPOD, the occasional drivers are ordinary citizens who make deliveries with their personal vehicles, which distinguishes VRPOD from the Vehicle Routing Problem with a private fleet and common carrier (VRPPC), in which drivers from other carriers are hired to make deliveries that the company cannot fulfill with its own fleet.

The VRPOD model proposed by Archetti, Savelsbergh and Speranza (2016) considers that the occasional driver can only make one delivery, accepting deviations from their route in exchange for financial compensation. Advancements in this variant were proposed by Macrina et al. (2017), considering the possibility of the occasional driver making multiple deliveries, limited

to the capacity of their vehicle, and that the deliveries have time windows for occurrence, introducing the Vehicle Routing Problem with Occasional Drivers and Time Windows (VRPODTWmd).

Due to the computational complexity of the VRPOD, recent literature has opted for the use of non-exact algorithms to solve it, especially for large instances. Macrina et al. (2017) use an exact algorithm to find optimal solutions for VRPODTWmd instances with up to 15 customers and 3 occasional drivers, but to solve the same problem for larger instances, with up to 100 customers and 30 occasional drivers, Macrina, Pugliese and Guerriero (2020) chose to use a heuristic.

2.2

Literature Review

Archetti, Guerriero and Macrina (2021) list challenges encountered in vehicle routing for last-mile delivery of e-commerce orders, namely: real-time order reception; short delivery deadlines; real-time arrival of products at last-mile distribution centers; concentration of order reception availability at the same time, when considering time windows. One category of vehicle routing problems for last-mile deliveries involves considering the use of vehicles that are not owned by the company. The adoption of collaborative strategies, such as using public transportation vehicles, crowd-shipping, and occasional drivers, has become an option to minimize costs, optimize delivery times, and reduce environmental impact. Among the points of attention listed by Bachofner et al. (2022) are the increase in congestion and urban pollution, due to the aging fleet of vehicles in some cases and their lack of adaptation to the complexity of urban environments. Archetti, Guerriero and Macrina (2021) also consider the use of crowd-shipping as one of the challenges to be addressed for last-mile vehicle routing models.

The Vehicle Routing Problem with Occasional Drivers (VRPOD) is recent, with its first study published by Archetti, Savelsbergh and Speranza (2016). This variant of the traditional vehicle routing problem was developed in the wake of the growth of e-commerce. Thus, VRPOD considers that online sales deliveries can be carried out by the company's regular driver fleet or by occasional drivers who make a detour in their route to make the delivery in exchange for financial compensation.

After Archetti, Savelsbergh and Speranza (2016), other publications contributed to the advancement of VRPOD by making extensions and proposing new solution methods. Archetti, Guerriero and Macrina (2021) proposed to classify VRPODs according to three characteristics: Information about cus-

customer orders, information about the availability of occasional drivers, and the number of deliveries that an occasional driver can perform. This work will present existing studies in the literature in light of these aspects.

Firstly, Archetti, Savelsbergh and Speranza (2016) developed VRPOD considering that customer orders and the availability of occasional drivers were deterministic and that only one delivery could be made by each occasional driver. As a solution method, the authors devised a mathematical model and a heuristic, combining variable neighborhood search (VNS) and tabu search (TS). Throughout the article, they studied different models of financial compensation for occasional drivers to understand the impact of this compensation on the routing result.

Macrina et al. (2017) proposed an extension to VRPOD, also considering that customer demand and the availability of occasional drivers were deterministic, and that occasional drivers could make multiple deliveries. They also considered that both occasional driver and company's own fleet deliveries had time windows, and this variant became known as VRPODTWmd. The authors studied the impact of using split deliveries on routing costs, and their results showed a positive impact. As a solution method, they adopted a mathematical model, capable of solving small instances to optimality. Later, Macrina and Guerriero (2018) developed a VNS heuristic, subsequently improved by Macrina, Pugliese and Guerriero (2020), making it possible to solve larger instances.

Pugliese et al. (2022) proposed new solution algorithms for the Vehicle Routing Problem with Occasional Drivers, Time Windows, and multiple deliveries (VRPODTWmd). They developed three versions of the GRASP metaheuristic, combining it with variable neighborhood search (VNS) and variable neighborhood descend (VND), and conducted experiments comparing the new solution methods with the algorithms proposed by Macrina et al. (2017) and Macrina, Pugliese and Guerriero (2020), as well as comparing them with the commercial solver CPLEX for small instances. The computational results showed that the heuristics are more efficient in terms of time than CPLEX, with a gap of only 3% from optimality. Moreover, as the size of the instances increases, the GRASP metaheuristics become more efficient and effective, achieving good quality results with acceptable execution times.

Torres, Gendreau and Rei (2022b) evaluated several uncertainties associated with the use of occasional drivers, due to their heterogeneous fleet and vehicles not being originally designed for carrying cargo. They proposed a model that considers the use of occasional drivers and uncertainty in the presence of the customer to receive the delivery, and they suggested other un-

certainties that could be considered in future works. Subsequently, in another study, Torres, Gendreau and Rei (2022a) further explored uncertainty in the routing decision made by the occasional driver. Other approaches for the use of crowdshipping in e-commerce last-mile delivery can be found in Mohamed and Ndiaye (2018); Huang and Ardiansyah (2019); Azcuy, Agatz and Giesen (2021); Martín-Santamaría et al. (2021); Wang, Xu and Qin (2023).

2.3

Solution Methods

Considering the solution methods, the literature contains both exact mathematical models and heuristic algorithms. Exact mathematical models are found in various articles; however, they are limited to solving small instances, as for instances closer in size to real-world problems, finding a solution in a reasonable amount of time is not feasible. Therefore, heuristic algorithms, despite not guaranteeing optimality for all cases, are more commonly used due to their ability to generate high-quality solutions in a short computational time. According to (BRAEKERS; RAMAEKERS; NIEUWENHUYSE, 2016), in their review of vehicle routing problems, 71.25% of the studies use metaheuristics as a solution method.

Considering the recent literature for the VRPOD Archetti, Savelsbergh and Speranza (2016) proposed a mathematical model and an heuristic with variable neighborhood search (VNS) and tabu search (TS) combined. Macrina et al. (2017) proposed a mathematical model solving small instances and in their work extensions developed a VNS heuristic ((MACRINA; GUERRIERO, 2018); (MACRINA; PUGLIESE; GUERRIERO, 2020)). Pugliese et al. (2022) developed three versions of the GRASP metaheuristic, combining it with Variable Neighbourhood Search (VNS) and Variable Neighbourhood Descent (VND) and solved small instances with the commercial solver CPLEX to compare the metaheuristics solution with the optimal one.

For this work solving the instances with a mathematical model was not considered due to the size of the instances and the need of an algorithm that provides the solution in low execution times. The Iterated Local Search metaheuristic was adopted due to its simplicity, easy implementation and understanding of the algorithm by the company which is the object of the study.

2.3.1

Iterated Local Search

The ILS metaheuristic was firstly proposed by Lourenço, Martin and Stützle (2003) and its essence is generate an initial solution (S_0) and iteratively execute movements to improve this solution, updating the best solution (S^*) if the acceptance criterion is satisfied. In each iteration, a perturbation movement is made in order to better explore the solution space. The calibration of the perturbation is important to avoid heavy changes on the solution (S^*) because this will lead to randomic exploration of the solution space. The Algorithm 1 presents the ILS structure and its four components: Initial solution, Local Search, Perturbation and Acceptance Criterion.

Algorithm 1: ILS

```

1  $S_0 = \text{generateInitialSolution}()$ 
2  $s^* = \text{localSearch}(S_0)$ 
3 while Termination Condition is not reached do
4    $s' = \text{perturb}(s^*)$ 
5    $s^{*' } = \text{localSearch}(s^{*' })$ 
6    $s^* = \text{acceptanceCriterion}(s^*, s^{*' })$ 

```

The ILS starts with an initial solution generation, after that a local search procedure is applied in line 2, improving the initial solution S_0 and finding a local optimum solution s^* . Between lines 3 and 6 the iterative part of the ILS occurs, first executing a perturbation in line 4 that removes the solution from the local minimum and generates the solution s' . Other local search move is applied in line 5 in order to find other optimal solution $s^{*'}$, if this solution meets the acceptance criterion then the current solution s^* is updated to $s^{*'}$. This loop occurs until the defined termination condition is not reached.

The **Initial Solution** could be generated randomly or by an greedy constructive, both options has its benefits but in general, the greedy constructive is a better choice because the quality of the initial solution affects quality of the final solution and simplify the movements made in the local-search step (LOURENÇO; MARTIN; STÜTZLE, 2003).

The **Local Search** objective is to improve the previous solution by exploring the neighborhood, this can be done using a single neighborhood structure or changing the local search algorithm systematically. The Variable Neighborhood Descent (VND) proposed by Mladenović and Hansen (1997) is an option when applying distinct neighborhood structures.

The **Perturbation** is used to force the diversification of the solution, preventing the solution to be stucked in a local minimum and increasing

the chances of improvements. The number of components modified by the perturbation defines its strength, if the perturbation made is weak the next local-search will fall back to the previous solution and the solution space will not be explored sufficiently. On the other hand, if the perturbation modifies many elements this can lead to a random exploring, which is not good too.

The **Acceptance Criterion** is used to define when a the best solution will be updated with the intermediate solutions generated by the local-search procedure. In order to improve the ILS diversification this criteria should accept not only better solutions, but some worse solutions could be accepted, when only improvement solutions are accepted the acceptance criterion favors the intensification of the solution. An intermediate criteria to balance intensification and diversification like the Simulated Annealing (SA) could be used.

2.4

Mathematical Formulation

The VRPOD proposed in this work considers that the occasional driver can make multiple deliveries, similar to Macrina et al. (2017), but there is no time window for the deliveries. Additionally, the extension proposed in this work imposes a maximum financial value limit that can be transported by both the own fleet and the occasional drivers on each trip. This is because, in the real scenario, companies have insurance for the transported goods, so that in case of any problem during transportation, the company reduces its financial losses. As a result, the total value of the cargo cannot exceed the limit imposed by the insurer. Besides the maximum monetary value, the capacity of the vehicle is measured in terms of weight and volume.

Another difference that the problem defined in this work has is that occasional drivers do not have a final destination; they are simply seeking to make deliveries in order to receive their financial compensation. The vehicles from the private fleet has the possibility to not return to the depot too. Another assumption regarding occasional drivers is that their initial location is considered to be the depot from which they will pick up the product. This assumption is valid since, in companies using this occasional driver system, the financial compensation does not consider the route that the driver travels to the product pickup location, and the time factor is not considered in this work. Therefore, the time spent to reach the depot also does not impact the model.

The compensation paid for the occasional drivers differs from the previous studies proposed. This model considers that both the occasional drivers and

the private fleet costs are the sum of a fixed cost paid for use the vehicle and a variable cost based on the distance traveled. This modification in the compensation schema occurs to made the model more suitable to the real-world scenario described by the company. A consequence of this adjustment is the model simplification, eliminating the need of working with occasional drivers and private fleet as distinct vehicle sets, because the unique difference between the fleets in the previous proposed formulations was the cost evaluation.

Based on these extensions defined for this work and the formulation proposed by Pugliese et al. (2022), we can define the problem considering only the objective of minimize the total routing cost as a directed graph $G = (N, A)$, where N represents the nodes and A represents the arcs. The set of nodes N is composed of the set C , which contains the locations of customers who wish to receive the products, and the depot, denoted as s . The fleet (company's own drivers plus occasional drivers) is heterogeneous, so each vehicle has a weight and volume capacities and a value limit for transportation, denoted as W_v, S_v and $P_v, v \in V$, respectively, where V represents the set of vehicles in the fleet.

Besides that, each vehicle has the possibility to return to the depot or not, this is denoted by the binary parameter r^v with a value of 1 if the vehicle must return to the depot. Each arc $a \in A$ has a cost c_a^v and a duration z_a^v for each vehicle $v \in V$, and each customer $i \in C$ has a demand with weight, volume and monetary value denoted by dw_i, ds_i, dp_i , respectively.

Let x_{ij}^v be the binary variable with a value of 1 if vehicle v traverses the arc (i, j) and 0 otherwise. The variables $w_i^v, s_i^v, p_i^v, t_i^v$ indicate, for each node $i \in C$, the weight, the volume, the value loaded on the vehicle v and t_i^v the total time traveled by the vehicle after visiting node i . Therefore, we can define the proposed VRPOD as follows:

$$\min \sum_{v \in V} \sum_{i \in C \cup \{s\}} \sum_{j \in C \cup \{s\}} c_{ij}^v x_{ij}^v + f^v x_{sj}^v \quad (2-1)$$

$$\text{s.t.} \sum_{v \in V} \sum_{i \in C \cup \{s\}} x_{ij}^v = 1 \quad \forall j \in C \quad (2-2)$$

$$\sum_{v \in V} \sum_{j \in C \cup \{s\}} x_{ij}^v = 1 \quad \forall i \in C \quad (2-3)$$

$$\sum_{j \in C} x_{sj}^v - \sum_{j \in C} x_{js}^v = 0 \quad \forall v \in V \quad (2-4)$$

$$\sum_{v \in V} \sum_{j \in C} x_{sj}^v \leq |V| \quad (2-5)$$

$$\sum_{j \in C} x_{sj}^v \leq 1 \quad \forall v \in V \quad (2-6)$$

$$\sum_{j \in C} x_{js}^v \leq r^v \quad \forall v \in V \quad (2-7)$$

$$w_j^v \geq w_i^v + dw_j x_{ij}^v - Q_v(1 - x_{ij}^v) \quad \forall v \in V, i, j \in C \cup \{s\} \quad (2-8)$$

$$s_j^v \geq s_i^v + ds_j x_{ij}^v - S_v(1 - x_{ij}^v) \quad \forall v \in V, i, j \in C \cup \{s\} \quad (2-9)$$

$$p_j^v \geq p_i^v + dp_j x_{ij}^v - P_v(1 - x_{ij}^v) \quad \forall v \in V, i, j \in C \cup \{s\} \quad (2-10)$$

$$t_j^v \geq t_i^v + z_{ij} x_{ij}^v - T_v(1 - x_{ij}^v) \quad \forall v \in V, i, j \in C \cup \{s\} \quad (2-11)$$

$$x_{ij}^v \in \{0, 1\} \quad \forall (i, j) \in A, v \in V \quad (2-12)$$

$$0 \leq w_j^v \leq Q_v \quad \forall i \in C \cup \{s\}, v \in V \quad (2-13)$$

$$0 \leq p_i^v \leq P_v \quad \forall i \in C \cup \{s\}, v \in V \quad (2-14)$$

$$0 \leq s_i^v \leq S_v \quad \forall i \in C \cup \{s\}, v \in V \quad (2-15)$$

$$0 \leq t_i^v \leq T_v \quad \forall i \in C \cup \{s\}, v \in V \quad (2-16)$$

The Objective function (2-1) consists of two terms. The first term computes the transportation variable costs i.e., the cost for each arc the traversed in the graph. The second term relates to the fixed cost paid for using the vehicle.

Constraints (2-2) to (2-4) are flow conservation constraints, ensuring that each vehicle that visits a node also leaves it. Constraints (2-5) and (2-6) state that the number of vehicles used will not exceed a certain limit, and each vehicle departs from the depot only once. Constraint (2-7) is used to fix in zero the value of the arcs from all customer to the depot for all vehicles that must not return to the depot.

Finally, Constraints (2-8) to (2-11) ensure that the customers demand is met, avoid cycles, and ensure that the vehicle's load capacities (weight, volume and monetary value) is not exceeded nor the maximum route duration. Constraints (2-12) to (2-16) define the domain of the problem's variables.

3

Iterated Local Search for the Vehicle Routing Problem With Occasional Drivers

In this chapter the proposed ILS algorithm will be detailed, describing the procedures and the problem-specific solution strategies that was used.

3.1

Solution Representation

The problem solution is represented by a set of itinerary structures, each one representing a vehicle. Each itinerary structure contains a vector representing the route defined for that vehicle, the total cost, its fixed and variable costs components as well as the weight, volume and monetary value loaded. The last attributes of the itinerary structure are the distance traveled, the duration of the route, the number of vertices visited and the vehicle identification. The solution still have its total, fixed and variable costs, and a vector with the no-serviced customers.

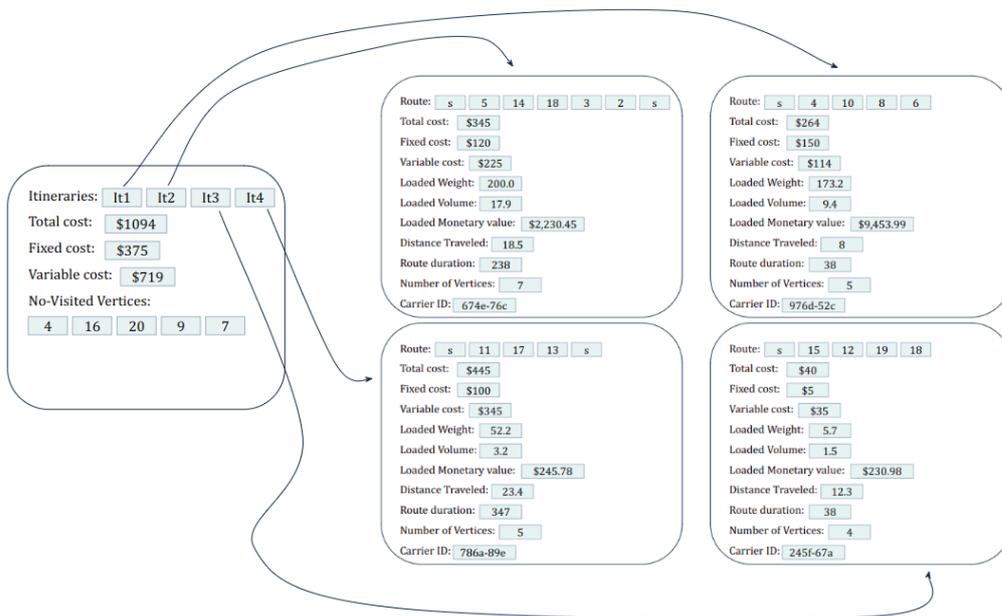


Figure 3.1: Solution Representation

Figure 3.1 shows a solution with four itinerary objects i.e., four distinct vehicles executing its route. All routes starts on the depot s but, some not return. The itinerary It1 has a route that starts and finish on the depot and visits five customers (5, 14, 18, 3, 2), the total cost of this itinerary (\$345) is the sum of the fixed cost (\$120) and the variable cost (\$225). All packages loaded

sum 200.0 Kg, 17.9 M³ and its monetary value is \$2,230.45. This itinerary route travels 18.5 Km and takes 238 minutes to finish. Finally, the solution total, fixed and variable costs are \$1094, \$375 and \$719 respectively, and five customers are not visited.

3.2 Constructive Algorithm

In order to generate the initial solution, two constructive algorithms are generated. The first one uses the nearest neighbor strategy, adding first the depot and after that selecting the closest vertex to the last element visited in route.

The second constructive algorithm selects the vertices i.e., customers to be added to the route considering the distance and the monetary value of the package to be delivered. The first vertex added for each route is the depot, after that a list of vertices is generated using the nearest neighbor strategy, adding to the list the two closest vertices from the last vertex in the route. From the closest vertices list the cheaper element is selected and added to the route.

This procedure is repeated for both algorithms until one of these constraints is reached: (i) maximum duration of route, (ii) maximum weight capacity, (iii) maximum volume capacity, (iv) maximum monetary value of cargo capacity. When no more elements can be added to the route the depot is added to the end if the vehicle must return, or in case of the return is not needed the route is ended in the last customer attended. This procedure is repeated until exists vehicles disposable.

The set of vertices i.e, customers to be served is represented by $V = \{s, 2, 3, \dots, n\}$, being $v = s$ the depot. Each vertex has its own demand attributes (weight, volume and monetary value), the location and the stop time needed to deliver the package there. The Vehicles set is represented by $C = \{1, 2, 3, \dots, k\}$. Each vehicle has its own capacities attributes (weight, volume and monetary value), costs attributes (variable cost per kilometer and fixed cost of using the vehicle), the average speed and the information if returning to the depot is required. Besides these data, the other inputs for the algorithms are the distance matrix and the routes duration limit.

The first constructive is presented in the Algorithm 2. In the lines 1-8, the parameters are loaded from the inputs and the variables for the capacities loaded in the routes are created. The while loop (lines 9-29) iterates over the vehicles, assigning vertices to each one until the capacity constraints are met. The for loop starting in line 11 iterates over the vertices, defining the initial

values for the best vertex index and its cost in lines 13. Lines 15-22, validates if the vertex is already routed and whether adding the vertex in the route will violate the capacity constraints or the maximum duration of the route. When none of these conditions is met the cost to deliver in the vertex is calculated considering the distance from the last element in route and the variable cost of the vehicle and if the calculated cost is lower than the lowest calculated cost the best vertex and best cost values are updated.

In the lines 23 to 26 if no vertex is selected the route is closed, adding the depot in the end if it is needed and resetting the variables for the capacities loaded in the routes to zero. In other cases i.e, a vertex is selected, the vertex is assigned to the route and the variables of the capacities loaded in the route is updated, this occurs between lines 27 and 29. The last step of the algorithm is fulfill the unattended customers list with the vertices which were not assigned to any vehicle route.

The constructive heuristic presented in Algorithm 3 is an enhancement of the first one, selecting the cheapest delivery among the nearest neighbors. The parameters loaded from the input and the variables are the same (l. 1-8) and the iteration over the vehicles occurs in the same way of the Algorithm 2, but the vertex selection criteria changes. A list of candidates vertices is created in line 11 is fulfilled in the for loop (l. 12-24). The validations (l. 14-17) that occurs are the same of the previous algorithm, after that the cost of adding this element in route is calculated, if the number of elements in the candidate list is lower than its maximum size the element is added. If the maximum size of the list is already reached a new element is only added in place of the maximum cost element, if the cost of visiting the candidate vertex is lower than the greatest cost in list (l. 19-23). If exists an element in candidates list after iterating over all vertices the element with minimum cost in the list is assigned to the route and the variables of the capacities loaded in the route is updated, this occurs between lines 25 and 27. In other case i.e, candidates list is empty, the route is closed, adding the depot in the end if it is needed and resetting the variables for the capacities loaded in the routes to zero. After iterating over all vertices and vehicles the unattended customers list is fulfilled with the vertices which were not assigned to any vehicle route.

The output of booth constructive algorithms are a list of routes, where each route has its own costs, variable, fixed and total, the values for weight, volume and monetary value carried in the vehicle, the total distance and duration of the routed and the number of vertices visited in the route. Besides that the solution object has the costs (variable, fixed and total) aggregating all routes and the list of unattended customers.

Algorithm 2: Nearest Neighbor constructive

Input: Vehicles, capacities (weight, volume, monetary value), vertices, distance matrix, route maximum duration

Output: Routes, costs (total, variable, fixed), unattended customers

```

1 routeDurationLimit ← route maximum duration
2 maxCandidatesListSize ← candidates list maximum size
3 maxKg ← total weight carried in the route
4 maxM3 ← total volume carried in the route
5 maxTicket ← total monetary value carried in the route
6 maxDuration ← total duration of route
7 routes ← list of routes
8 routedVertices ← list of vertices assigned to a route
9 while  $k$  in  $C$  do
10   add depot to route
11   for  $v$  in  $V$  do
12     dist = calculate distance from last element in route to  $v$ 
13     bestVertice = -1
14     bestCost = bigM
15     if  $v \notin$  routedVertices &
16        $maxKg + v.Kg \leq k.maxKg$  &
17        $maxM3 + v.M3 \leq k.maxM3$  &
18        $maxDuration + \frac{dist}{k.averageSpeed} + v.stopDuration \leq$ 
       routeDurationLimit then
19       newCost = dist × k.costPerKm
20       if newCost ≤ minimumCost then
21         bestVertice =  $v$ 
22       bestCost = newCost
23   if bestVertice == -1 then
24     if  $k$  must return to depot then
25       add depot in the end of the route
26     resets maxKg, maxM3, maxTicket and maxDuration to Zero
27   else
28     assigns bestVertice to the route and to the routedVertices list
29     increases maxKg, maxM3, maxTicket and maxDuration
30 unattendedCustomers ← all customers  $\notin$  routedVertices

```

Algorithm 3: Nearest Cheaper Neighbor constructive

Input: Vehicles, capacities (weight, volume, monetary value), vertices, distance matrix, route maximum duration

Output: Routes, costs (total, variable, fixed), unattended customers

```

1 routeDurationLimit ← route maximum duration
2 maxCandidatesListSize ← candidates list maximum size
3 maxKg ← total weight carried
4 maxM3 ← total volume carried
5 maxTicket ← total monetary value carried
6 maxDuration ← total duration of route
7 routes ← list of routes
8 routedVertices ← list of vertices assigned to a route
9 while  $k$  in  $C$  do
10   add depot to route
11   candidatesList ← list of nearest vertices from the last element in
   route
12   for  $v$  in  $V$  do
13     dist = calculate distance from last element in route to  $v$ 
14     if  $v \notin$  routedVertices &
15        $maxKg + v.Kg \leq k.maxKg$  &
16        $maxM3 + v.M3 \leq k.maxM3$  &
17        $maxDuration + \frac{dist}{k.averageSpeed} + v.stopDuration \leq$ 
   routeDurationLimit then
18       newCost = dist × k.costPerKm
19       if candidatesList ≤ maxCandidatesListSize then
20         | add  $v$  to candidatesList
21       else
22         if newCost ≤ maximumCost(candidatesList) then
23           | add  $v$  to candidatesList in place of the maximum
           | cost element
24     |
25   if candidatesList size ≥ 1 then
26     | assigns the cheapest  $v$  to the route and to the routedVertices
     | list
27     | increases maxKg, maxM3, maxTicket and maxDuration
28   else
29     if  $k$  must return to depot then
30       | add depot in the end of the route
31     | resets maxKg, maxM3, maxTicket and maxDuration to Zero
32   unattendedCustomers ← all customers  $\notin$  routedVertices

```

3.3 Local Search

The following step of the ILS is the local search. In this phase the initial solution is improved by the exploration of the neighborhood. In this work we considered the insert, remove, shift, swap and relocate moves, exploring the neighborhood inter-routes and intra-route. In Figure 3.2 an example solution with three routes leaving from and returning to the depot, represented by the triangle, and with four non visited vertices is presented and the local search moves will be explained based in this solution.

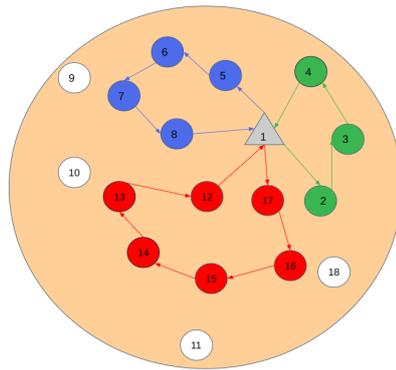


Figure 3.2: Example Route

Considering that for the studied problem it is possible not visiting all vertices, the insert move intends to add one of the unattended vertices to a route, without violating the capacity and duration constraints. Figure 3.3a shows the node 17 being added to the green route between nodes 2 and 3. The remove move presented in Figure 3.3b select one vertex that is assigned to a route and removes it from there. The shift move changes the position of two consecutive elements of a route, being considered an intra-route move. The Figure 3.3c shows the nodes 7 and 8 being permuted in the blue route.

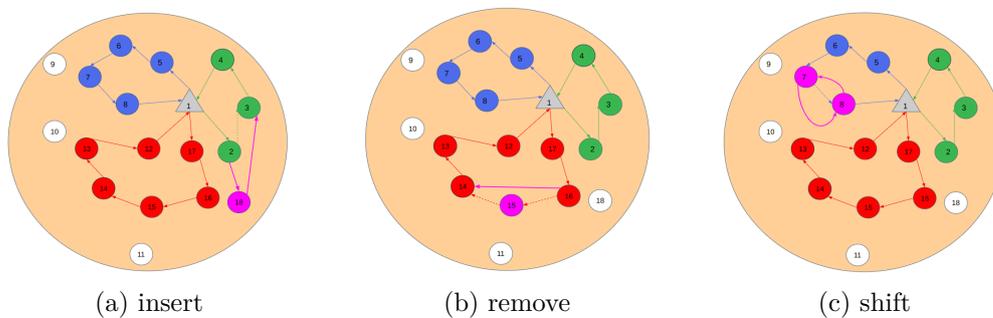


Figure 3.3: A set of three subfigures: (a) insert move; (b) remove move; (c) shift move.

The swap move changes the position of two non consecutive vertices, which could be in the same route or not. The Figure 3.4a shows an example when both vertices are in the same route and the Figure 3.4b presents a move with the vertices in distinct routes. Another move that can be intra-route or inter-route is the relocate. This move removes an element from it's original position in route and adds it in another position, in the same route (Figure 3.5a) or in another one (Figure 3.5b).

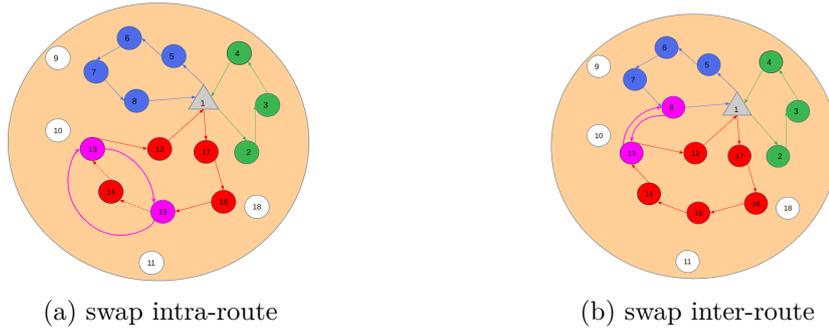


Figure 3.4: A set of two subfigures: (a) swap intra-route move; (b) swap inter-route move.

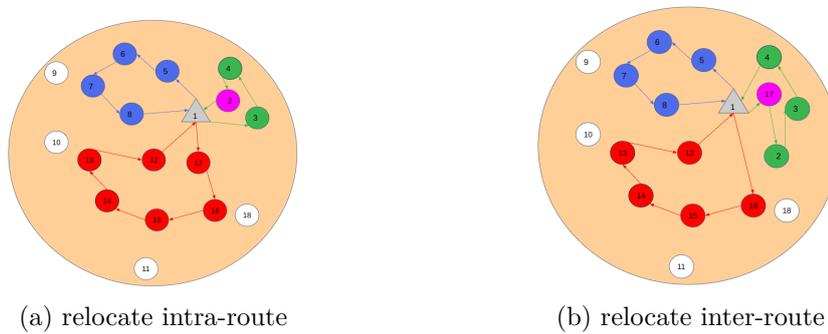


Figure 3.5: A set of two subfigures: (a) relocate intra-route move; (b) relocate inter-route move.

All moves made in the local search phase must respect the capacity constraints for weight, volume and monetary value, and the route maximum duration constraint. Using the VND, the neighborhoods are randomized and using the first improvement strategy, the first successful move of each iteration was accepted. The Algorithms 4, 5 and 6 presents the evaluation made in order to accept the enhancement made by the local search. This evaluation occurs for all the neighborhoods and guarantees that only moves that improve the quality of the solution, respecting the capacity and route duration constraints are executed. For this study an improvement occurs when the number of routed

vertices increase or when this number of vertices do not decrease but a solution with lower total cost is found.

The procedure for insert moves described in Algorithm 4 starts in line 1 with a loop for all routes in solution. In lines 2-9 another loop occurs for the selected route, iterating over all indices in the route, and in line 3 a loop is made for all unattended vertices. Then, the insert move is made in line 4 and in lines 5-9 the move is evaluated. Two evaluations are made, if the number of vertices in the solution is improved (l. 5), the changes made by the move is executed. If the changes in the visited vertices number is equal to zero but a cost reduction (l. 8) is made the move is executed. If none of this evaluation criteria are met the move is not executed. This iteration occurs for all index in each route trying to add all unattended vertices, one by time, in the solution.

Algorithm 4: Local Search procedure for insert

```

1 for route1 in routes do
2   for index in route1 do
3     for element in unattended vertices do
4       costImprovement, numVerticesImprovement  $\leftarrow$  insert
5         element in index in route1
6       if numVerticesImprovement  $\geq 0$  then
7         | execute move
8       else
9         | if numVerticesImprovement = 0 & costImprovement  $\leq$ 
          | 0 then
          | | execute move

```

The procedure for remove moves described in Algorithm 5 is similar to the insert. In line 1 a loop for all routes in solution is made. Between lines 2 to 9 another loop occurs for the selected route, iterating over all indices in the route. Then, the remove move is made in line 3 and in lines 4-8 the move is evaluated. Two evaluations are made, if the number of vertices in the solution is improved (l. 4), the changes made by the move is executed. If the changes in the visited vertices number is equal to zero but a cost reduction (l. 7) is made the move is executed. If none of this evaluation criteria are met the move is not executed.

This iteration occurs for all index in each route trying to remove vertices from the solution, one by time. Considering the evaluation criteria, this move is more useful in the perturbation phase, because this two criteria does not have to be met.

Algorithm 5: Local search procedure for remove

```

1 for route1 in routes do
2   for index in route1 do
3     costImprovement, numVerticesImprovement ←remove element
       in index in route1
4     if numVerticesImprovement ≥ 0 then
5       | execute move
6     else
7       | if numVerticesImprovement = 0 & costImprovement ≤ 0
8         | then
           | execute move

```

The procedure for shift, swap and relocate moves is the same and is described in Algorithm 6. For this local search moves the procedure is different the the previous two described because two routes are selected. The procedure starts in with four nested loops for all routes in solution, and for the selected routes, iterating over all elements in the routes (l. 1-4). Then, the move is made in line 5 and in lines 6-10 the move is evaluated. The same evaluations from the other two local search procedures are made.

Algorithm 6: Local search procedure for shift, swap and relocate

```

1 for route1 in routes do
2   for route2 in routes do
3     for element1 in route1 do
4       for element2 in route 2 do
5         costImprovement, numVerticesImprovement ←evaluate
           move for the pair element1 and element2
6         if numVerticesImprovement ≥ 0 then
7           | execute move
8         else
9           | if numVerticesImprovement = 0 &
10            | costImprovement ≤ 0 then
              | execute move

```

3.4 Perturbation

After the local search, the third step of the ILS is the perturbation. This phase is enables the solution to explore better the solution space. For this study the perturbation step executes a number of moves to modify the solution until

it reaches a number of successful moves or a limit of total iterations. The algorithm 7 shows the moves made. Two termination conditions are evaluated, line 1 starts a while loop until the limit of iterations is met and in line 2 occurs an evaluation if the number of successful moves are met, until it does not occurs new iterations are made. Then, in line 3 a perturbation move is selected, the possible moves are the same of the described in previous section. In line 6 its evaluated if the capacity and route duration constraints are respected, when positive the move is made. For each iteration the number of total moves is updated and when the move respects the constraints and the perturbation is executed the number of successful move is update too.

Algorithm 7: Perturbation

```

1 while totalMoves ≤ limitTotalIterations do
2   if numberOfSuccessMoves < limitSuccessMoves then
3     move ← select a random neighborhood and evaluate a move
4     if move respects the capacity and route duration constraints
5       then
6         execute the move and update the numberOfSuccessMoves
           count
7     update the limitTotalIterations count

```

3.5

Acceptance Criterion

In order to evaluate the new solution generated in each iteration of the ILS and decide if the current solution will be updated the acceptance criterion chosen for this study is the Simulated Annealing. In this criteria the solution is updated given the probability $e^{-(f(s_c^*)-f(s^*))/T}$ where s_c^* is the candidate solution computed in the iteration, s^* is the current solution and $T > 0$ is the temperature. For this study we worked with a multi objective problem, the first one is maximize the number of vertices, i.e customers, visited and the second is to reduce the total cost of visiting these customers, because of that we must consider this distinct objectives in the Simulated Annealing.

Considering this two objectives we are going to have an initial temperature T_i for each one of them. This temperature decreases each iteration being multiplied by a factor r ($0 < r < 1$), according to the equation $T = Tr$. The initial and final temperatures are defined for each objective following the criteria defined by Pisinger and Ropke (2007), $T_i = (perc_{init} \cdot C) / -\log(perc_{init})$ and $T_f = (perc_{final} \cdot C) / -\log(perc_{final})$, being C the number of no-serviced vertices for the first objective and the total cost for the second one. The $perc_{init}$ is

the percentage of solutions that will be accepted with $perc_{init}$ of deterioration, the same is valid for $perc_{final}$. The factor is defined according to the number of total iterations n , $r = (T_f/T_i)^{(1/n)}$.

Taking into account that the objective of maximizing the number of customer visited is more important than the cost minimization this is the first condition to be evaluated. If the new solution is accepted considering the temperatures of this objective the other one is not evaluated and the current solution is updated, in other case the second objective is evaluated.

4 Numerical Experiments

This chapter presents the experiments realized using the proposed meta-heuristic to route last-mile deliveries from a real-world set of instances of a huge e-commerce company from Brazil. The results of the metaheuristic will be compared with the routes generated by the company using the number of customers serviced and the total cost of the route, being the first one more important, given that the company goal is to maximize the number of deliveries, in order to satisfy the majority of the customers, and then minimize the operational costs. All data provided was properly anonymized to prevent the identification of the company and its customers, respecting ethical principles.

4.1 Case study

The Brazilian e-commerce has been growing in the last years, and the entrance of international companies such as Amazon and Mercado Livre made this market more competitive. One alternative to become more efficient and increase customers satisfaction is to delivery the products from depots closer to the urban centers, making faster deliveries. In order to increase efficiency, one alternative is to use vehicle routing softwares to generate more assertive routes, reducing the unnecessary costs.

The company that provided the data for this study is one of those who invested in a vehicle routing algorithm to generate the last-mile deliveries routes. Being one of the top 5 e-commerce companies in the year of 2022 in Brazil, when considering the market share (PICKERT, 2022), this company deliveries for all states and combines the online and offline operation in order to better serve the customers. The last-mile deliveries are made from depots closer to the urban centers and are served by distribution centers of the company, both echelons are routed separately.

The figure 4.1 shows an representation of the company supply-chain, the green arrows represents the first-tier routes and the red and blue arrows the last-mile delivery routes made by the company private fleet or by an occasional driver, respectively. There are several depots for last-mile delivery, but vehicle routing is done separately and the deliveries can be made to customers' homes, lockers or pick-up locations, the last two are points where the delivery is made ad the customer must walk to this places to withdraw their packages. Occasional drivers are capable of transporting more than one order as proposed

by Macrina et al. (2017) and its initial location is assumed to be the same of the depot where they are going to take the product and the final destination is not previously defined.

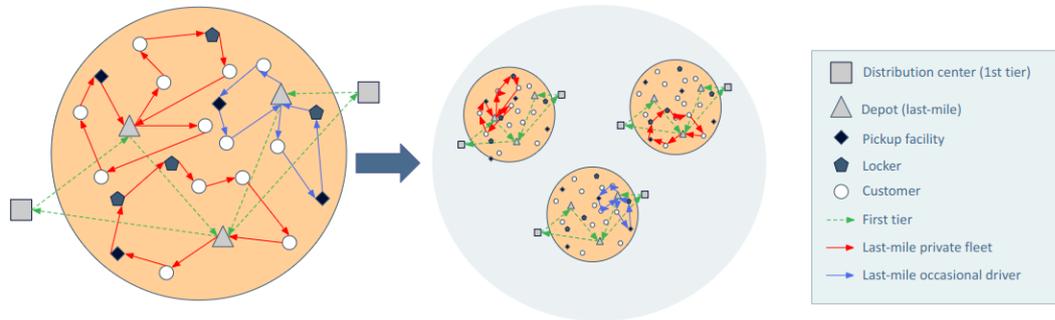


Figure 4.1: Company supply-chain
Adapted from (ZHOU et al., 2018)

Another important characteristic for the last-mile vehicle routing of this company is that besides the capacity constraints for volume and weight, the company imposes a constraint on the total monetary value of the packages loaded in each vehicle due to the need to respect the insurance company contractual limits and to reduce the losses if a problem occurs with the vehicle. Both fleets, occasional drivers and private fleet, are modeled as heterogeneous fleets.

4.2 Data Acquiring and treatment

The first step was to understand the company actual scenario and the data that was available. This was made in meetings with specialists who works with the last-mile vehicle routing generation in the company. After that, the next step was to define a time-window for collecting the data. Taking in consideration the months in which the number of sells is bigger the selected period was from 01/11/2022 to 09/11/2022. 1536 instances were collected and sorted by geolocation and date, which led to 13 states and 33 cities. A program was developed in Golang to access the data that was in a MongoDB database hosted by the company and the OpenStreetMap API was used to sort the instances by state and city.

For all the instances used in this study a distance matrix was generated using the OpenRouteService API in order to use real word distances, and the costs and duration of the routes generated by the company for this instances were updated taking in consideration this distance matrix generating a fair

comparison, by avoiding having different distance calculation methods between this study and the company.

4.3 Experiments

The experiments were conducted in a Intel® Core™ i7-10610U @ 1.80GHz with 16GB of RAM computer using Ubuntu 20.04.6 LTS as operational system and the ILS algorithm was coded and executed in Julia 1.8.3 version.

For the experiments a set of 121 instances from Rio de Janeiro state was selected. The smallest instance has 1 vertices and the biggest has 344, and all has it's own fleet of vehicles with it's capacity attributes and route maximum allowed duration. The experiments were conducted first using the constructive defined in algorithm 2 and after with the constructive that considers the cost of delivery and the monetary value of the customers demands (algorithm 3). Five different seeds were used to the random parts of the ILS, the the simulated Annealing number iterations was set to 50, the number of perturbations to 3 and the acceptance probabilities was set to 50% and 1% for the initial and final temperatures, respectively.

The last scenario considers the addition of occasional drivers to the list of available vehicles for the route. For each fleet was added at least one occasional driver and for each 30 customers a new occasional driver is added to the fleet. This experiment aims to understand the impact of using occasional drivers in the number of customer served and in the total cost. The occasional drivers costs are made up of a low fixed cost and a variable cost per kilometer driven, and each vehicle of this fleet has it's proper average speed, capacity limits for weight, volume and monetary value, and route maximum duration. This scenario was tested with both constructive algorithms.

4.3.1 Results

The proposed ILS was compared with the router algorithm developed by the company in terms of number of vertices visited and total cost of the route. As presented, maximize the number of customers attended is the first objective of the algorithm developed, then when comparing with the company router this was the first criteria and the cost minimization was considered as an secondary criteria to select the best solution when the number of vertices routed is equal for both, the ILS and the company routes.

The ILS with NearestNeighbor constructive (2) outperformed the existent company model in 53.72% of the instances and equaled the number of

visited customers with higher cost routes in 38.02% of the cases. In the cases where the ILS with NearestNeighbor constructive not performed better, the biggest difference in number of vertices visited occurred for instance **96399e59** when the proposed ILS attended 11 customers less than the actual router.

When adopting the constructive algorithm considering not only the distance between vertices but the financial value of the customer demand (3) the proposed ILS outperformed the existent company model in 50.41% of the instances and equaled the number of visited customers with higher cost routes in 45.45% of the cases. The biggest difference in the number of vertices routed is 7, and occurred for instance **9ecb96e7**.

Figure 4.2 presents a comparison between both ILS proposed and the company router. This figure shows that in average the ILS has similar performance when considering the number of vertices routed and both are better than the actual router used by the firm. In terms of average cost, the ILS with NearestNeighbor constructive (2) is the best between the algorithms proposed, but both has an average cost higher than the existent algorithm. An important remainder is that the number of vertices routed has direct interference in the cost and maximize the number of customers attended is the primary objective.

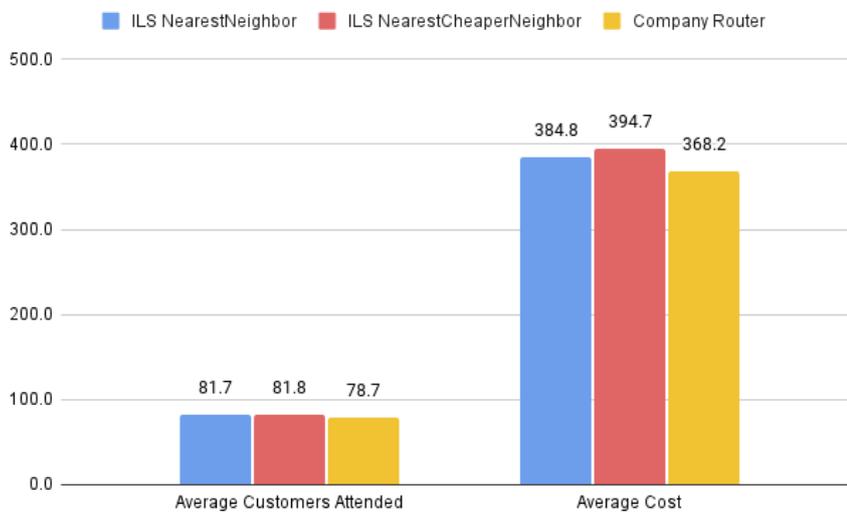


Figure 4.2: Average number of customers attended and costs comparison between ILS and company Router

The computational time needed to run both ILS is similar and increases as the number of vertices gets bigger. Figure 4.3 presents the computational time needed for both proposed algorithms by the number of vertices in the instance. The complete results of the ILS algorithms are presented in appendix ???. The number of vertices routed is highlighted in blue for the instances in which the ILS outperformed the existent routing algorithm.

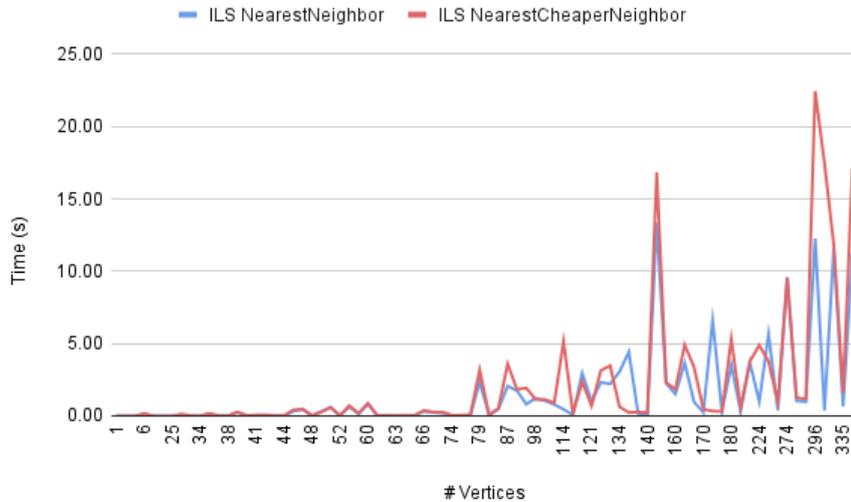


Figure 4.3: Computational time in seconds by the number of vertices in the instance

4.3.1.1 Occasional drivers scenario

The experiments with occasional drivers occurred in two manners. Firstly the occasional drivers had priority over the private fleet of the company, and a second experiment was conducted only using the crowdshipping alternative after loading the company fleet vehicles. This both experiments were realized in order to evaluate the efficiency of using occasional drivers and if the prioritization between occasional drivers and private fleet impacts in the result. The proposed ILS with NearestNeighbor constructive (2) was selected to this experiments since it had the best performance between the proposed algorithm.

At a first glance this difference on the fleet selection should not impact drastically on the final result, but how the occasional drivers fleet commonly are made of smaller vehicles with a lower capacity. The monetary value limit of the demand loaded on the vehicle suffers a downgrade too, since the drivers are considered more untrustworthy than the private fleet ones.

When the occasional drivers fleet has priority over the private fleet the proposed ILS outperformed the existent company model in 76.03% of the instances and equaled the number of visited customers with higher cost routes in 23.14% of the cases. Exists only one instance where the ILS with performed better, the difference in number of vertices visited occurred for instance **96399e59** when the proposed ILS attended 6 customers less than the actual router.

On the other hand, when the private fleet has priority over the occasional drivers fleet the proposed ILS outperformed the existent company model in

63.63% of the instances and equaled the number of visited customers with higher cost routes in 34.71% of the cases. In the cases where the ILS not performed better, the biggest difference in number of vertices visited occurred for instance **2993431a** when the proposed ILS attended 9 customers less than the actual router.

Figure 4.4 presents a comparison between both fleet prioritization strategies for the ILS proposed, the ILS without occasional drivers and the company router. This figure shows that in average both fleet prioritization strategies has similar performance when considering the number of vertices routed and both are better than the ILS without occasional drivers and the actual router used by the firm. In terms of average cost, the ILS with occasional drivers fleet being prioritized is the best between the algorithms proposed and the existent algorithm, even with the number of customers served upgrade. The complete results of the occasional drivers experiments are presented in appendix A

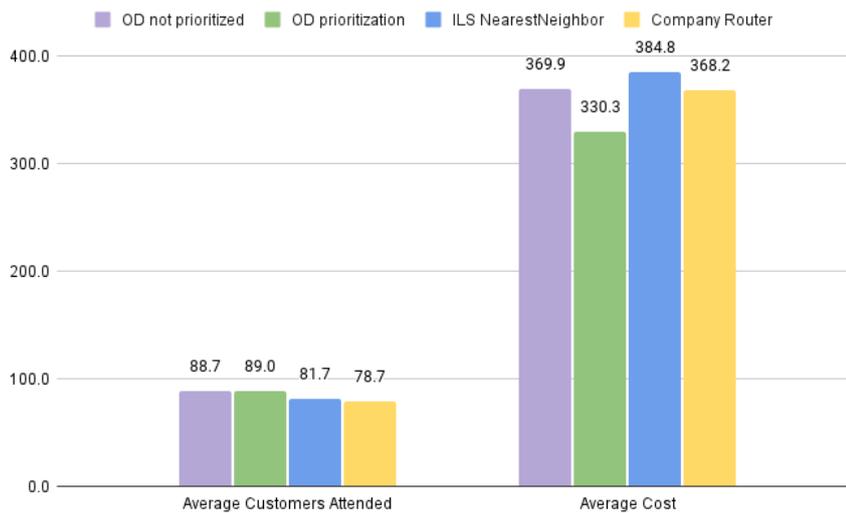


Figure 4.4: Average number of customers attended and costs comparison between ILS with occasional drivers, ILS without occasional drivers and company Router

After the experiments realized, the results shows that the proposed ILS outperforms the existent company router, respecting the real-world scenario restrictions and improving the average number of vertices routed. The experiments realized with the adoption of occasional drivers shows that the crowd-shipping helps on routing performance, leading to more customers served and cost reduction, in average.

5 Conclusion

This dissertation aims to solve the Vehicle Routing Problem with Occasional Drivers (VRPOD) for e-commerce last-mile delivery maximizing the number of customers served and minimizing the routing cost. In order to reach this objective an Iterated Local Search metaheuristic was proposed and data from a Brazilian e-commerce company was used to assess the quality of the developed algorithm and the impact of adopting the use of occasional drivers as an delivery strategy, complementing the company's private fleet.

The VRPOD is one of the VRP variants that considers the use of regular citizens to deliver packages using their own vehicles in addition to the company fleet, these citizens are called occasional drivers. For the proposed VRPOD, an heterogeneous fleet with maximum capacity is considered for the occasional drivers and the company fleet, limits on the maximum financial value loaded in each vehicle and on the rout duration is imposed.

The tests conducted in 121 instances with number of customers between 1 and 344 shows that the proposed ILS improves the number of customers served in average from 78.7 to 81.7 when not including the occasional drivers fleet and to 89.0 when they are included on the available fleet and prioritized over the company's regular fleet. The average routing cost for the ILS without occasional drivers is greater than the company router cost from 368.2 to 384.8, this is an acceptable result considering that the cost is directly associated to the number of vertices routed and an increasing on the average number of customers served is perceived. When the occasional drivers are available to be used on the routing strategy even with the increment on the number of customers served the routing cost decreased to 330.3, this cost reduction is associated with the occasional drives variable compensation per kilometer traveled, that is lower then the regular fleet variable cost per kilometer in the tested scenario.

As a result of the exposed results, it's possible to conclude that this work objectives were achieved with the development of an ILS metaheuristic algorithm incorporating a fleet of occasional drivers and generating solutions with good quality in comparison to the existent company router. These results can be used by the company to improve the existent routing algorithm and to stimulate the adoption of occasional drivers to deliver last-mile e-commerce customers demand.

Future works should consider uncertainty on occasional drivers avail-

ability to deliver the assigned packages. Since they are not direct company employees they must reject the proposed route due to different situations, similar to what happens with Uber and other crowdsourcing companies, in which the driver could not accept the job offer in order to wait for a better one. Other future researches can be conducted considering restrictions for the occasional drivers fleet, for example defining a specific set of products that the crowdshippers are allowed to deliver, avoiding the most expensive products. Exploring different scenarios for occasional drivers compensation schema, understanding in which cases the use of these strategy becomes worse than not using the occasional drivers. Extensions can be realized on the solution methods, proposing an exact algorithm in order to test its performance in this real-world scenario and improving the proposed metaheuristic, adopting other local-search and perturbation strategies and testing other ILS parameters.

6

Bibliography

ARCHETTI, C.; GUERRIERO, F.; MACRINA, G. The online vehicle routing problem with occasional drivers. **Computers & Operations Research**, Elsevier, v. 127, p. 105144, 2021.

ARCHETTI, C.; SAVELSBERGH, M.; SPERANZA, M. G. The vehicle routing problem with occasional drivers. **European Journal of Operational Research**, Elsevier, v. 254, n. 2, p. 472–480, 2016.

AZCUY, I.; AGATZ, N.; GIESEN, R. Designing integrated urban delivery systems using public transport. **Transportation Research Part E: Logistics and Transportation Review**, Elsevier, v. 156, p. 102525, 2021.

BACHOFNER, M. et al. City logistics: Challenges and opportunities for technology providers. **Journal of Urban Mobility**, Elsevier, v. 2, p. 100020, 2022.

BOYSEN, N.; FEDTKE, S.; SCHWERDFEGER, S. Last-mile delivery concepts: a survey from an operational research perspective. **Or Spectrum**, Springer, v. 43, p. 1–58, 2021.

BRAEKERS, K.; RAMAEKERS, K.; NIEUWENHUYSE, I. V. The vehicle routing problem: State of the art classification and review. **Computers & industrial engineering**, Elsevier, v. 99, p. 300–313, 2016.

COLLA, E.; LAPOULE, P. E-commerce: exploring the critical success factors. **International Journal of Retail & Distribution Management**, Emerald Group Publishing Limited, v. 40, n. 11, p. 842–864, 2012.

DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. **Management science**, Informs, v. 6, n. 1, p. 80–91, 1959.

HUANG, K.; ARDIANSYAH, M. N. A decision model for last-mile delivery planning with crowdsourcing integration. **Computers & Industrial Engineering**, Elsevier, v. 135, p. 898–912, 2019.

LAPORTE, G. The vehicle routing problem: An overview of exact and approximate algorithms. **European journal of operational research**, Elsevier, v. 59, n. 3, p. 345–358, 1992.

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

MACRINA, G.; GUERRIERO, F. The green vehicle routing problem with occasional drivers. **New Trends in Emerging Complex Real Life Problems: ODS, Taormina, Italy, September 10–13, 2018**, Springer, p. 357–366, 2018.

MACRINA, G.; PUGLIESE, L. D. P.; GUERRIERO, F. A variable neighborhood search for the vehicle routing problem with occasional drivers and time windows. In: **ICORES**. [S.l.: s.n.], 2020. p. 270–277.

MACRINA, G. et al. The vehicle routing problem with occasional drivers and time windows. In: SPRINGER. **Optimization and Decision Science: Methodologies and Applications: ODS, Sorrento, Italy, September 4-7, 2017** 47. [S.l.], 2017. p. 577–587.

MARTÍN-SANTAMARÍA, R. et al. An efficient algorithm for crowd logistics optimization. **Mathematics**, MDPI, v. 9, n. 5, p. 509, 2021.

MLADENović, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997.

MOHAMED, E.; NDIAYE, M. Optimal routing and scheduling in e-commerce logistics using crowdsourcing strategies. In: IEEE. **2018 7th International Conference on Industrial Technology and Management (ICITM)**. [S.l.], 2018. p. 248–253.

NIELSEN, E. **Webshoppers Free: O relatório de maior credibilidade sobre o e-commerce brasileiro**. 45. ed. 2022. <<https://company.ebit.com.br/webshoppers/webshoppersfree>>. Accessed: 2023-02-01.

PAGAR.ME. **E-commerce no Brasil: dados sobre o mercado digital no país (2022)**. 2022. <<https://pagar.me/blog/ecommerce-no-brasil/#:~:text=Como%20j%C3%A1%20adiantamos%2C%20o%20e,em%20rela%C3%A7%C3%A3o%20ao%20ano%20anterior>>. Accessed: 2023-02-01.

PICKERT, L. **Quais são as redes de e-commerce do Brasil mais admiradas?** 2022. <<https://blog.aainovacao.com.br/ecommerce-brasil/>>. Accessed: 2023-07-31.

PISINGER, D.; ROPKE, S. A general heuristic for vehicle routing problems. **Computers & operations research**, Elsevier, v. 34, n. 8, p. 2403–2435, 2007.

PUGLIESE, L. D. P. et al. Solution approaches for the vehicle routing problem with occasional drivers and time windows. **Optimization Methods and Software**, Taylor & Francis, v. 37, n. 4, p. 1384–1414, 2022.

TORRES, F.; GENDREAU, M.; REI, W. Crowdshipping: An open vrp variant with stochastic destinations. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 140, p. 103677, 2022.

TORRES, F.; GENDREAU, M.; REI, W. Vehicle routing with stochastic supply of crowd vehicles and time windows. **Transportation Science**, INFORMS, v. 56, n. 3, p. 631–653, 2022.

ÜLKÜ, M. A.; BOOKBINDER, J. H. Optimal quoting of delivery time by a third party logistics provider: The impact of shipment consolidation and temporal pricing schemes. **European Journal of Operational Research**, Elsevier, v. 221, n. 1, p. 110–117, 2012.

WANG, L.; XU, M.; QIN, H. Joint optimization of parcel allocation and crowd routing for crowdsourced last-mile delivery. **Transportation Research Part B: Methodological**, Elsevier, v. 171, p. 111–135, 2023.

ZHOU, L. et al. A multi-depot two-echelon vehicle routing problem with delivery options arising in the last mile distribution. **European Journal of Operational Research**, Elsevier, v. 265, n. 2, p. 765–778, 2018.

A

Complete Results

Table A.1: Iterated Local Search with NearestNeighbor Constructive

Instance	# Vertices	ILS - NearestNeighbor			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
03e50b86	143	141	\$474.78	5.43	141	\$472.28
0401f205	42	36	\$152.67	0.04	38	\$141.16
0482d9bf	87	79	\$221.18	1.15	77	\$174.81
056b8837	6	5	\$226.35	0.01	5	\$229.44
07084e5c	61	61	\$255.96	0.02	61	\$253.29
075a08a4	32	32	\$239.94	0.00	32	\$236.71
08da04f0	134	133	\$473.65	3.05	134	\$468.41
0a3bca29	91	80	\$530.21	0.59	76	\$485.51
0c3bffaе	2	2	\$224.65	0.00	1	\$221.84
0f1929e3	273	273	\$1,333.35	0.40	269	\$1,458.70
0ff3e607	129	123	\$483.74	2.21	125	\$469.85
11217156	335	335	\$1,376.22	0.65	329	\$1,372.18
11ef48ef	103	98	\$699.68	0.79	94	\$518.80
135f75ef	51	50	\$240.54	0.60	49	\$237.71
159c10f9	35	35	\$241.31	0.01	35	\$240.12
16f36e73	45	44	\$225.46	0.32	44	\$221.90
1724e41b	151	79	\$302.46	1.09	74	\$329.36
1c79e41a	79	71	\$242.62	0.42	62	\$231.20
1ce32bbf	50	50	\$232.55	0.01	50	\$231.76
1d071dbb	98	95	\$243.85	1.18	86	\$223.09
1d3eb8ce	41	41	\$235.08	0.01	41	\$231.45
1d9e50cd	79	75	\$566.38	0.56	69	\$504.77
1ffe20da	18	18	\$232.26	0.00	18	\$232.26
21351861	32	32	\$238.25	0.00	32	\$238.06
22468a25	34	34	\$170.38	0.00	34	\$182.83
22834c44	1	1	\$221.62	0.00	1	\$221.62
28827e59	55	54	\$193.78	0.58	54	\$189.67
2993431a	94	85	\$252.83	0.81	89	\$236.64
2b2ed1fb	2	2	\$220.61	0.00	2	\$220.61
2b37d469	35	35	\$224.83	0.00	35	\$222.52
2f2dd38f	114	101	\$557.78	0.38	104	\$489.10
2f382f34	57	35	\$169.16	0.05	8	\$128.86
32a5faa3	80	80	\$168.17	0.05	80	\$167.91
3a7ead04	315	295	\$1,869.39	11.60	250	\$1,598.71
3cad2924	25	25	\$233.62	0.00	25	\$232.20
3d5a8b21	278	43	\$86.66	1.04	34	\$84.04
3e7231eb	35	35	\$244.64	0.01	35	\$241.46
403c2873	118	114	\$476.22	1.72	104	\$447.98
41905408	41	41	\$212.94	0.00	41	\$212.27
46cc7fa3	344	325	\$1,249.37	11.24	282	\$1,134.37
47847e28	137	137	\$479.21	0.16	137	\$468.55
489c72e8	31	29	\$218.74	0.08	27	\$216.19
4b7d7ccc	117	117	\$685.80	0.07	117	\$664.81
4ba05fe0	70	44	\$250.15	0.20	42	\$236.77
4d46e7e6	57	32	\$151.98	0.08	32	\$153.03
4d948891	78	78	\$184.10	0.04	78	\$179.34
4e683994	48	48	\$235.32	0.01	48	\$224.21
4e7765d7	43	43	\$235.88	0.01	43	\$231.57
52ced919	35	32	\$140.26	0.11	32	\$142.81
52e151a0	38	38	\$231.62	0.00	38	\$229.74
56541e5a	102	92	\$250.19	1.05	84	\$241.58
58f4f18f	164	154	\$512.44	3.66	152	\$472.06
5906f1af	55	55	\$250.54	0.01	55	\$245.01
5a7e84bf	38	38	\$231.95	0.00	38	\$231.84
5d928023	2	2	\$81.06	0.00	2	\$81.06
5e12a0ca	98	98	\$475.30	0.04	98	\$466.82
5e41ee44	224	103	\$123.34	0.99	87	\$90.43
5ef30dd7	160	160	\$642.01	0.17	160	\$621.03

Table A.2: Iterated Local Search with NearestNeighbor Constructive (Continuation)

Instance	# Vertices	ILS - NearestNeighbor			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
62208a1c	40	40	\$213.42	0.00	40	\$211.97
639b3b93	84	74	\$520.27	0.46	73	\$506.65
676b68cd	47	46	\$234.53	0.43	46	\$232.44
67e2e595	65	65	\$233.75	0.03	65	\$232.18
6cc939f8	67	50	\$204.98	0.24	44	\$202.87
6cd23867	2	2	\$80.12	0.00	2	\$80.12
6f6e9c65	170	170	\$690.40	0.24	166	\$684.72
7165c4cf	180	136	\$836.84	2.37	121	\$519.04
75560a05	202	202	\$811.56	0.32	190	\$793.84
75e83bf1	8	8	\$223.17	0.00	8	\$223.17
7682282d	143	114	\$765.58	1.19	108	\$508.22
7a38e29d	36	36	\$212.15	0.00	36	\$212.11
7cd081e3	282	282	\$914.53	0.97	282	\$1,131.27
7e239480	303	303	\$1,855.13	0.38	277	\$1,836.00
7e241d83	63	63	\$301.64	0.01	63	\$347.30
815ecc93	36	36	\$238.08	0.00	36	\$236.22
84bf4d36	76	76	\$177.37	0.03	76	\$175.01
8530155c	178	176	\$821.64	6.60	177	\$788.99
86bfe535	151	87	\$306.18	1.17	81	\$338.82
8720efbe	40	40	\$226.77	0.00	40	\$220.65
87cd3b86	3	3	\$84.98	0.00	3	\$84.98
88bf930	39	38	\$159.48	0.25	38	\$174.48
8b454b5b	50	49	\$155.00	0.28	49	\$161.32
8d83f97d	140	140	\$669.58	0.08	140	\$665.17
9236f1ca	1	1	\$222.21	0.00	1	\$222.21
94804764	2	2	\$223.73	0.00	2	\$223.73
9514e5b9	296	286	\$952.99	12.26	286	\$933.55
96399e59	169	139	\$488.18	0.97	150	\$452.13
97278703	245	196	\$1,178.68	5.65	180	\$1,033.82
97775f21	79	76	\$255.88	1.37	77	\$245.85
9c286137	87	79	\$250.40	0.91	83	\$238.30
9ecb96e7	41	30	\$226.77	0.06	28	\$225.06
9f354095	50	50	\$238.87	0.01	50	\$234.96
9fbf43b9	8	8	\$82.50	0.00	8	\$82.50
a054d949	206	133	\$488.16	3.67	110	\$459.72
a2b2e517	114	114	\$464.98	0.07	114	\$463.72
a49c4840	39	39	\$167.38	0.00	39	\$166.85
a52ed0a2	64	64	\$241.35	0.02	64	\$239.78
a7d6df77	44	44	\$167.13	0.01	42	\$153.51
a8b1ad39	160	115	\$730.06	1.34	109	\$505.90
a8f27b26	1	1	\$222.21	0.00	1	\$222.21
a8f3196b	52	52	\$232.04	0.02	52	\$230.59
a97aa0a3	62	62	\$244.65	0.02	62	\$241.57
aa86073c	121	95	\$248.61	1.01	91	\$233.31
b5dcba03	274	261	\$1,481.71	9.59	226	\$1,300.65
b7a8598e	3	3	\$136.32	0.00	3	\$136.32
bcd0cfe2	55	55	\$182.22	0.02	55	\$179.05
bcdc7968	180	168	\$561.74	1.15	163	\$605.14
befb47a1	74	74	\$176.06	0.02	74	\$171.63
befba3ff	91	89	\$234.14	1.15	87	\$223.28
bfd9983	66	52	\$208.56	0.29	49	\$205.56
c19bc08c	135	134	\$466.63	4.41	134	\$460.52
c3271f4e	55	55	\$179.72	0.01	55	\$174.62
c5bfa9aa	60	59	\$239.54	0.81	59	\$237.32
c852ff15	3	3	\$135.21	0.00	3	\$135.21
c8cf7356	43	43	\$244.52	0.00	43	\$236.91
ca4332b4	118	90	\$108.40	1.23	83	\$98.83
cd5af395	3	3	\$226.31	0.00	3	\$226.31
d03bdbfc	66	66	\$220.52	0.02	66	\$217.16
da9dc38a	179	179	\$964.44	0.30	179	\$1,033.81
e3d15fe5	128	122	\$469.37	2.32	126	\$456.03
e685f7dc	6	5	\$207.37	0.13	5	\$209.84
e98fd72f	8	8	\$232.81	0.00	8	\$232.81

Table A.3: Iterated Local Search with NearestCheaperNeighbor Constructive

Instance	# Vertices	ILS - NearestCheaperNeighbor			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
03e50b86	143	141	\$480.10	8.43	141	\$472.28
03e50b86	143	141	\$480.10	7.04	141	\$472.28
0401f205	42	39	\$155.04	0.08	38	\$141.16
0482d9bf	87	83	\$209.51	1.79	77	\$174.81
056b8837	6	5	\$224.56	0.14	5	\$229.44
07084e5c	61	61	\$266.09	0.04	61	\$253.29
075a08a4	32	32	\$245.58	0.01	32	\$236.71
08da04f0	134	134	\$481.68	0.63	134	\$468.41
0a3bca29	91	81	\$562.64	0.81	76	\$485.51
0c3bfae	2	2	\$224.65	0.00	1	\$221.84
0f1929e3	273	273	\$1,497.56	0.93	269	\$1,458.70
0ff3e607	129	125	\$479.72	3.47	125	\$469.85
11217156	335	335	\$1,396.95	1.57	329	\$1,372.18
11ef48ef	103	93	\$647.98	0.90	94	\$518.80
135f75ef	51	50	\$248.36	0.58	49	\$237.71
159c10f9	35	35	\$245.04	0.01	35	\$240.12
16f36e73	45	44	\$235.85	0.41	44	\$221.90
1724e41b	151	82	\$460.08	1.17	74	\$329.36
1c79e41a	79	73	\$244.71	0.61	62	\$231.20
1ce32bbf	50	50	\$236.27	0.02	50	\$231.76
1d071dbb	98	97	\$249.93	1.04	86	\$223.09
1d3eb8ce	41	41	\$236.88	0.01	41	\$231.45
1d9e50cd	79	74	\$738.46	0.49	69	\$504.77
1ffe20da	18	18	\$236.17	0.00	18	\$232.26
21351861	32	32	\$242.95	0.01	32	\$238.06
22468a25	34	34	\$174.16	0.01	34	\$182.83
22834c44	1	1	\$221.62	0.00	1	\$221.62
28827e59	55	54	\$197.70	0.64	54	\$189.67
2993431a	94	91	\$247.29	1.93	89	\$236.64
2b2ed1fb	2	2	\$220.61	0.00	2	\$220.61
2b37d469	35	35	\$234.30	0.02	35	\$222.52
2f2dd38f	114	105	\$506.78	0.83	104	\$489.10
2f382f34	57	36	\$160.94	0.13	8	\$128.86
32a5faa3	80	80	\$178.17	0.08	80	\$167.91
3a7ead04	315	281	\$1,773.36	11.83	250	\$1,598.71
3cad2924	25	25	\$238.96	0.00	25	\$232.20
3d5a8b21	278	45	\$87.25	1.27	34	\$84.04
3e7231eb	35	35	\$249.02	0.01	35	\$241.46
403c2873	118	115	\$486.36	1.14	104	\$447.98
41905408	41	41	\$216.51	0.01	41	\$212.27
46cc7fa3	344	320	\$1,313.95	17.12	282	\$1,134.37
47847e28	137	137	\$485.33	0.27	137	\$468.55
489c72e8	31	29	\$218.82	0.12	27	\$216.19
4b7d7ccc	117	117	\$689.00	0.17	117	\$664.81
4ba05fe0	70	43	\$243.52	0.27	42	\$236.77
4d46e7e6	57	33	\$153.46	0.06	32	\$153.03
4d948891	78	78	\$194.62	0.09	78	\$179.34
4e683994	48	48	\$235.88	0.04	48	\$224.21
4e7765d7	43	43	\$237.95	0.01	43	\$231.57
52ced919	35	33	\$165.67	0.13	32	\$142.81
52e151a0	38	38	\$237.31	0.01	38	\$229.74
56541e5a	102	89	\$257.51	1.13	84	\$241.58
58f4f18f	164	158	\$484.57	4.91	152	\$472.06
5906f1af	55	55	\$256.31	0.03	55	\$245.01
5a7e84bf	38	38	\$235.82	0.01	38	\$231.84
5d928023	2	2	\$81.06	0.00	2	\$81.06
5e12a0ca	98	98	\$477.37	0.09	98	\$466.82
5e41ee44	224	121	\$115.35	4.88	87	\$90.43
5ef30dd7	160	160	\$677.83	0.46	160	\$621.03

Table A.4: Iterated Local Search with NearestCheaperNeighbor Constructive (Continuation)

Instance	# Vertices	ILS - NearestCheaperNeighbor			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
62208a1c	40	40	\$216.41	0.02	40	\$211.97
639b3b93	84	74	\$521.14	0.51	73	\$506.65
676b68cd	47	46	\$234.63	0.48	46	\$232.44
67e2e595	65	65	\$238.33	0.04	65	\$232.18
6cc939f8	67	50	\$205.38	0.27	44	\$202.87
6cd23867	2	2	\$80.12	0.00	2	\$80.12
6f6e9c65	170	170	\$702.82	0.46	166	\$684.72
7165c4cf	180	131	\$798.78	2.85	121	\$519.04
75560a05	202	202	\$826.40	0.59	190	\$793.84
75e83bf1	8	8	\$223.17	0.00	8	\$223.17
7682282d	143	113	\$776.34	1.37	108	\$508.22
7a38e29d	36	36	\$213.78	0.01	36	\$212.11
7cd081e3	282	282	\$1,160.84	1.16	282	\$1,131.27
7e239480	303	301	\$1,867.60	17.48	277	\$1,836.00
7e241d83	63	63	\$313.70	0.02	63	\$347.30
815ecc93	36	36	\$240.09	0.01	36	\$236.22
84bf4d36	76	76	\$189.42	0.05	76	\$175.01
8530155c	178	178	\$802.55	0.33	177	\$788.99
86bfe535	151	84	\$360.60	1.14	81	\$338.82
8720efbe	40	40	\$229.56	0.02	40	\$220.65
87cd3b86	3	3	\$84.98	0.00	3	\$84.98
88bf930	39	38	\$162.36	0.25	38	\$174.48
8b454b5b	50	49	\$170.18	0.23	49	\$161.32
8d83f97d	140	140	\$680.17	0.22	140	\$665.17
9236f1ca	1	1	\$222.21	0.00	1	\$222.21
94804764	2	2	\$223.73	0.00	2	\$223.73
9514e5b9	296	291	\$969.59	22.45	286	\$933.55
96399e59	169	144	\$475.61	3.38	150	\$452.13
97278703	245	194	\$1,288.67	3.77	180	\$1,033.82
97775f21	79	77	\$258.11	2.07	77	\$245.85
9c286137	87	83	\$247.02	1.80	83	\$238.30
9ecb96e7	41	21	\$245.56	0.03	28	\$225.06
9f354095	50	50	\$241.65	0.02	50	\$234.96
9fbf43b9	8	8	\$82.50	0.00	8	\$82.50
a054d949	206	133	\$533.21	3.77	110	\$459.72
a2b2e517	114	113	\$470.09	4.32	114	\$463.72
a49c4840	39	39	\$174.12	0.01	39	\$166.85
a52ed0a2	64	64	\$247.38	0.04	64	\$239.78
a7d6df77	44	44	\$170.37	0.02	42	\$153.51
a8b1ad39	160	107	\$577.26	1.37	109	\$505.90
a8f27b26	1	1	\$222.21	0.00	1	\$222.21
a8f3196b	52	52	\$237.15	0.02	52	\$230.59
a97aa0a3	62	62	\$250.52	0.04	62	\$241.57
aa86073c	121	101	\$246.23	0.69	91	\$233.31
b5dcba03	274	258	\$1,426.67	9.53	226	\$1,300.65
b7a8598e	3	3	\$136.32	0.00	3	\$136.32
bcd0cfe2	55	55	\$192.47	0.02	55	\$179.05
bcdc7968	180	163	\$607.22	2.46	163	\$605.14
befb47a1	74	74	\$181.82	0.05	74	\$171.63
befba3ff	91	89	\$241.29	1.02	87	\$223.28
bfd9983	66	53	\$207.71	0.34	49	\$205.56
c19bc08c	135	135	\$490.89	0.24	134	\$460.52
c3271f4e	55	55	\$186.05	0.02	55	\$174.62
c5bfa9aa	60	59	\$242.15	0.87	59	\$237.32
c852ff15	3	3	\$135.21	0.00	3	\$135.21
c8cf7356	43	43	\$264.63	0.01	43	\$236.91
ca4332b4	118	83	\$119.17	1.22	83	\$98.83
cd5af395	3	3	\$226.31	0.00	3	\$226.31
d03bdbfc	66	66	\$230.15	0.03	66	\$217.16
da9dc38a	179	179	\$1,066.07	0.31	179	\$1,033.81
e3d15fe5	128	127	\$476.61	3.14	126	\$456.03
e685f7dc	6	5	\$210.26	0.01	5	\$209.84
e98fd72f	8	8	\$233.05	0.00	8	\$232.81

Table A.5: Iterated Local Search with Occasional Drivers prioritization

Instance	# Vertices	ILS - Occasional Drivers prioritization			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
03e50b86	143	142	\$465.18	4.86	141	\$472.28
0401f205	42	42	\$141.00	0	38	\$141.16
0482d9bf	87	87	\$191.71	0.04	77	\$174.81
056b8837	6	6	\$227.42	0	5	\$229.44
07084e5c	61	61	\$253.78	0.01	61	\$253.29
075a08a4	32	32	\$237.88	0	32	\$236.71
08da04f0	134	134	\$470.61	0.06	134	\$468.41
0a3bca29	91	86	\$539.55	0.8	76	\$485.51
0c3bffaе	2	2	\$224.65	0	1	\$221.84
0f1929e3	273	273	\$889.72	0.4	269	\$1,458.70
0ff3e607	129	129	\$465.54	3.26	125	\$469.85
11217156	335	335	\$926.35	0.5	329	\$1,372.18
11ef48ef	103	103	\$319.12	0.03	94	\$518.80
135f75ef	51	50	\$248.64	0.27	49	\$237.71
159c10f9	35	35	\$239.20	0	35	\$240.12
16f36e73	45	45	\$232.69	0.01	44	\$221.90
1724e41b	151	151	\$346.91	0.04	74	\$329.36
1c79e41a	79	76	\$246.54	0.65	62	\$231.20
1ce32bbf	50	50	\$231.11	0.01	50	\$231.76
1d071dbb	98	98	\$239.65	0.05	86	\$223.09
1d3eb8ce	41	41	\$233.62	0.01	41	\$231.45
1d9e50cd	79	77	\$520.62	0.61	69	\$504.77
1ffe20da	18	18	\$226.43	0	18	\$232.26
21351861	32	32	\$233.06	0	32	\$238.06
22468a25	34	34	\$145.04	0	34	\$182.83
22834c44	1	1	\$0.12	0	1	\$221.62
28827e59	55	54	\$168.09	0.68	54	\$189.67
2993431a	94	92	\$238.78	1.69	89	\$236.64
2b2ed1fb	2	2	\$0.05	0	2	\$220.61
2b37d469	35	35	\$223.48	0	35	\$222.52
2f2dd38f	114	106	\$472.55	0.84	104	\$489.10
2f382f34	57	55	\$134.07	0.23	8	\$128.86
32a5faa3	80	80	\$165.76	0.02	80	\$167.91
3a7ead04	315	315	\$1,627.67	0.57	250	\$1,598.71
3cad2924	25	25	\$231.57	0	25	\$232.20
3d5a8b21	278	144	\$100.11	3.17	34	\$84.04
3e7231eb	35	35	\$241.79	0	35	\$241.46
403c2873	118	117	\$455.65	2.67	104	\$447.98
41905408	41	41	\$213.25	0.01	41	\$212.27
46cc7fa3	344	344	\$1,160.23	0.3	282	\$1,134.37
47847e28	137	137	\$473.57	0.11	137	\$468.55
489c72e8	31	31	\$207.70	0.13	27	\$216.19
4b7d7ccc	117	117	\$444.14	0.11	117	\$664.81
4ba05fe0	70	65	\$246.24	0.4	42	\$236.77
4d46e7e6	57	44	\$166.71	0.09	32	\$153.03
4d948891	78	78	\$183.60	0.03	78	\$179.34
4e683994	48	48	\$234.54	0.01	48	\$224.21
4e7765d7	43	43	\$235.87	0.01	43	\$231.57
52ced919	35	34	\$138.09	0.1	32	\$142.81
52e151a0	38	38	\$230.71	0	38	\$229.74
56541e5a	102	102	\$261.68	0.03	84	\$241.58
58f4f18f	164	160	\$478.22	4.55	152	\$472.06
5906flaf	55	55	\$246.60	0.01	55	\$245.01
5a7e84bf	38	38	\$231.37	0	38	\$231.84
5d928023	2	2	\$0.10	0	2	\$81.06
5e12a0ca	98	98	\$470.49	0.04	98	\$466.82
5e41ee44	224	212	\$124.36	6.71	87	\$90.43
5ef30dd7	160	160	\$399.11	0.09	160	\$621.03

Table A.6: Iterated Local Search with Occasional Drivers prioritization (Continuation)

Instance	# Vertices	ILS - Occasional Drivers prioritization			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
62208a1c	40	40	\$212.12	0	40	\$211.97
639b3b93	84	81	\$582.70	0.54	73	\$506.65
676b68cd	47	46	\$234.54	0.28	46	\$232.44
67e2e595	65	65	\$233.17	0.01	65	\$232.18
6cc939f8	67	65	\$206.42	0.45	44	\$202.87
6cd23867	2	2	\$0.02	0	2	\$80.12
6f6e9c65	170	170	\$692.07	0.11	166	\$684.72
7165c4cf	180	179	\$476.92	5.17	121	\$519.04
75560a05	202	202	\$660.33	0.27	190	\$793.84
75e83bf1	8	8	\$222.19	0	8	\$223.17
7682282d	143	143	\$469.16	0.06	108	\$508.22
7a38e29d	36	36	\$212.72	0	36	\$212.11
7cd081c3	282	282	\$686.32	0.45	282	\$1,131.27
7e239480	303	303	\$1,623.08	0.5	277	\$1,836.00
7e241d83	63	63	\$160.44	0.01	63	\$347.30
815ecc93	36	36	\$236.92	0	36	\$236.22
84bf4d36	76	76	\$176.22	0.01	76	\$175.01
8530155c	178	178	\$654.48	0.16	177	\$788.99
86bfe535	151	151	\$342.86	0.05	81	\$338.82
8720efbe	40	40	\$219.66	0	40	\$220.65
87cd3b86	3	3	\$0.45	0	3	\$84.98
88bfb930	39	39	\$145.14	0	38	\$174.48
8b454b5b	50	50	\$144.46	0.01	49	\$161.32
8d83f97d	140	140	\$445.85	0.1	140	\$665.17
9236f1ca	1	1	\$0.17	0	1	\$222.21
94804764	2	2	\$223.73	0	2	\$223.73
9514e5b9	296	296	\$941.86	14.43	286	\$933.55
96399e59	169	144	\$446.33	2.24	150	\$452.13
97278703	245	239	\$1,019.63	7.74	180	\$1,033.82
97775f21	79	78	\$241.87	0.89	77	\$245.85
9c286137	87	84	\$247.16	0.96	83	\$238.30
9ecb96e7	41	33	\$274.64	0.08	28	\$225.06
9f354095	50	50	\$239.00	0.01	50	\$234.96
9fbf43b9	8	8	\$81.78	0	8	\$82.50
a054d949	206	200	\$499.05	6.2	110	\$459.72
a2b2e517	114	114	\$465.13	0.04	114	\$463.72
a49c4840	39	39	\$166.79	0	39	\$166.85
a52ed0a2	64	64	\$239.16	0.02	64	\$239.78
a7d6df77	44	44	\$164.03	0.01	42	\$153.51
a8b1ad39	160	160	\$493.53	0.05	109	\$505.90
a8f27b26	1	1	\$0.17	0	1	\$222.21
a8f3196b	52	52	\$230.97	0.01	52	\$230.59
a97aa0a3	62	62	\$241.94	0.02	62	\$241.57
aa86073c	121	114	\$235.29	3.39	91	\$233.31
b5dcba03	274	274	\$1,294.45	0.23	226	\$1,300.65
b7a8598e	3	3	\$0.42	0	3	\$136.32
bcd0cfe2	55	55	\$180.47	0.01	55	\$179.05
bcdc7968	180	180	\$492.80	0.12	163	\$605.14
befb47a1	74	74	\$169.85	0.02	74	\$171.63
befba3ff	91	90	\$226.17	1.48	87	\$223.28
bfd9983	66	57	\$209.67	0.2	49	\$205.56
c19bc08c	135	135	\$469.76	0.06	134	\$460.52
c3271f4e	55	55	\$175.77	0.01	55	\$174.62
c5bfa9aa	60	60	\$234.76	0.02	59	\$237.32
c852ff15	3	3	\$0.52	0	3	\$135.21
c8cf7356	43	43	\$242.70	0.01	43	\$236.91
ca4332b4	118	117	\$114.45	2.7	83	\$98.83
cd5af395	3	3	\$0.45	0	3	\$226.31
d03bdbfc	66	66	\$218.24	0.01	66	\$217.16
da9dc38a	179	179	\$736.85	0.14	179	\$1,033.81
e3d15fe5	128	128	\$457.86	3.68	126	\$456.03
e685f7dc	6	6	\$206.26	0.02	5	\$209.84
e98fd72f	8	8	\$226.82	0	8	\$232.81

Table A.7: Iterated Local Search with Occasional Drivers not prioritized

Instance	# Vertices	ILS - Occasional Drivers prioritization			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
03e50b86	143	142	\$465.18	4.86	141	\$472.28
03e50b86	143	142	\$470.73	3.93	141	\$472.28
0401f205	42	42	\$151.11	0	38	\$141.16
0482d9bf	87	86	\$165.77	1.33	77	\$174.81
056b8837	6	6	\$231.29	0	5	\$229.44
07084e5c	61	61	\$255.96	0.02	61	\$253.29
075a08a4	32	32	\$239.94	0	32	\$236.71
08da04f0	134	134	\$463.72	0.05	134	\$468.41
0a3bca29	91	84	\$515.28	0.89	76	\$485.51
0c3bfae	2	2	\$224.65	0	1	\$221.84
0f1929e3	273	273	\$1,333.35	0.43	269	\$1,458.70
0f3e607	129	126	\$465.00	2.33	125	\$469.85
11217156	335	335	\$1,376.22	0.7	329	\$1,372.18
11ef48ef	103	103	\$466.90	0.03	94	\$518.80
135f75ef	51	50	\$240.49	0.51	49	\$237.71
159c10f9	35	35	\$241.31	0	35	\$240.12
16f36e73	45	45	\$225.93	0.01	44	\$221.90
1724e41b	151	151	\$305.80	0.04	74	\$329.36
1c79e41a	79	78	\$247.13	0.32	62	\$231.20
1ce32bbf	50	50	\$232.55	0.01	50	\$231.76
1d071dbb	98	98	\$233.18	1.93	86	\$223.09
1d3eb8ce	41	41	\$235.08	0.01	41	\$231.45
1d9e50cd	79	78	\$518.99	0.6	69	\$504.77
1ffe20da	18	18	\$232.26	0	18	\$232.26
21351861	32	32	\$238.25	0	32	\$238.06
22468a25	34	34	\$170.38	0	34	\$182.83
22834c44	1	1	\$221.62	0	1	\$221.62
28827e59	55	54	\$194.49	0.53	54	\$189.67
2993431a	94	80	\$233.02	0.79	89	\$236.64
2b2ed1fb	2	2	\$220.61	0	2	\$220.61
2b37d469	35	35	\$224.83	0	35	\$222.52
2f2dd38f	114	112	\$430.59	1.06	104	\$489.10
2f382f34	57	44	\$199.87	0.1	8	\$128.86
32a5faa3	80	80	\$168.17	0.03	80	\$167.91
3a7ead04	315	315	\$1,626.95	15.65	250	\$1,598.71
3cad2924	25	25	\$233.62	0	25	\$232.20
3d5a8b21	278	143	\$108.88	4.07	34	\$84.04
3e7231eb	35	35	\$244.64	0	35	\$241.46
403c2873	118	116	\$468.44	2.16	104	\$447.98
41905408	41	41	\$212.94	0	41	\$212.27
46cc7fa3	344	343	\$1,148.87	22.94	282	\$1,134.37
47847e28	137	137	\$479.21	0.15	137	\$468.55
489c72e8	31	29	\$218.41	0.09	27	\$216.19
4b7d7ccc	117	117	\$685.80	0.07	117	\$664.81
4ba05fe0	70	63	\$251.14	0.33	42	\$236.77
4d46e7e6	57	42	\$193.01	0.1	32	\$153.03
4d948891	78	78	\$184.10	0.03	78	\$179.34
4e683994	48	48	\$235.32	0.01	48	\$224.21
4e7765d7	43	43	\$235.88	0.01	43	\$231.57
52ced919	35	34	\$133.93	0.1	32	\$142.81
52e151a0	38	38	\$231.62	0	38	\$229.74
56541e5a	102	102	\$251.12	0.05	84	\$241.58
58f4f18f	164	162	\$469.79	3.74	152	\$472.06
5906f1af	55	55	\$250.54	0.01	55	\$245.01
5a7e84bf	38	38	\$231.95	0	38	\$231.84
5d928023	2	2	\$81.06	0	2	\$81.06
5e12a0ca	98	98	\$475.30	0.04	98	\$466.82
5e41ee44	224	213	\$133.35	8.71	87	\$90.43
5ef30dd7	160	160	\$642.01	0.16	160	\$621.03

Table A.8: Iterated Local Search with Occasional Drivers not prioritized (Continuation)

Instance	# Vertices	ILS - Occasional Drivers prioritization			Company Router	
		#Routed Vertices	Total Cost	Time (Seconds)	#Routed Vertices	Total Cost
62208a1c	40	40	\$213.42	0	40	\$211.97
639b3b93	84	76	\$519.82	0.5	73	\$506.65
676b68cd	47	46	\$234.65	0.34	46	\$232.44
67e2e595	65	65	\$233.75	0.02	65	\$232.18
6cc939f8	67	64	\$207.34	0.39	44	\$202.87
6cd23867	2	2	\$80.12	0	2	\$80.12
6f6e9c65	170	170	\$690.40	0.22	166	\$684.72
7165c4cf	180	179	\$466.73	4.67	121	\$519.04
75560a05	202	202	\$811.56	0.32	190	\$793.84
75e83bf1	8	8	\$223.17	0	8	\$223.17
7682282d	143	143	\$467.82	0.06	108	\$508.22
7a38e29d	36	36	\$212.15	0	36	\$212.11
7cd081c3	282	282	\$914.53	1.05	282	\$1,131.27
7e239480	303	303	\$1,855.13	0.4	277	\$1,836.00
7e241d83	63	63	\$301.64	0.01	63	\$347.30
815ecc93	36	36	\$238.08	0	36	\$236.22
84bf4d36	76	76	\$177.37	0.03	76	\$175.01
8530155c	178	178	\$783.15	0.1	177	\$788.99
86bfe535	151	151	\$309.26	0.07	81	\$338.82
8720efbe	40	40	\$226.77	0	40	\$220.65
87cd3b86	3	3	\$84.98	0	3	\$84.98
88bfb930	39	39	\$159.73	0	38	\$174.48
8b454b5b	50	50	\$156.82	0.01	49	\$161.32
8d83f97d	140	140	\$669.58	0.07	140	\$665.17
9236f1ca	1	1	\$222.21	0	1	\$222.21
94804764	2	2	\$223.73	0	2	\$223.73
9514e5b9	296	293	\$927.83	19.38	286	\$933.55
96399e59	169	144	\$456.36	1.95	150	\$452.13
97278703	245	239	\$1,021.41	7.49	180	\$1,033.82
97775f21	79	78	\$256.11	1.12	77	\$245.85
9c286137	87	85	\$243.36	1	83	\$238.30
9ecb96e7	41	31	\$227.64	0.07	28	\$225.06
9f354095	50	50	\$238.87	0.01	50	\$234.96
9fbf43b9	8	8	\$82.50	0	8	\$82.50
a054d949	206	194	\$491.59	6.89	110	\$459.72
a2b2e517	114	114	\$464.98	0.06	114	\$463.72
a49c4840	39	39	\$167.38	0.01	39	\$166.85
a52ed0a2	64	64	\$241.35	0.02	64	\$239.78
a7d6df77	44	44	\$167.13	0.01	42	\$153.51
a8b1ad39	160	159	\$463.04	2.47	109	\$505.90
a8f27b26	1	1	\$222.21	0	1	\$222.21
a8f3196b	52	52	\$232.04	0.02	52	\$230.59
a97aa0a3	62	62	\$244.65	0.03	62	\$241.57
aa86073c	121	117	\$234.06	2.77	91	\$233.31
b5dcba03	274	274	\$1,297.10	0.44	226	\$1,300.65
b7a8598e	3	3	\$136.32	0	3	\$136.32
bcd0cfe2	55	55	\$182.22	0.01	55	\$179.05
bcde7968	180	179	\$510.11	4.14	163	\$605.14
befb47a1	74	74	\$176.06	0.02	74	\$171.63
befba3ff	91	90	\$227.47	1.53	87	\$223.28
bfd9983	66	65	\$210.18	0.42	49	\$205.56
c19bc08c	135	135	\$465.92	0.11	134	\$460.52
c3271f4e	55	55	\$179.72	0.02	55	\$174.62
c5bfa9aa	60	59	\$239.38	0.67	59	\$237.32
c852ff15	3	3	\$135.21	0	3	\$135.21
c8cf7356	43	43	\$244.52	0	43	\$236.91
ca4332b4	118	114	\$112.19	2.63	83	\$98.83
cd5af395	3	3	\$226.31	0	3	\$226.31
d03bdbfc	66	66	\$220.52	0.02	66	\$217.16
da9dc38a	179	179	\$964.44	0.29	179	\$1,033.81
e3d15fe5	128	128	\$458.64	0.09	126	\$456.03
e685f7dc	6	5	\$208.22	0.13	5	\$209.84
e98fd72f	8	8	\$232.81	0	8	\$232.81