TÉCNICO LISBOA

Site-Dependent Vehicle Routing Problem with Hard Time Windows

Gui Xavier Pereira Carvalho

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. Susana Margarida da Silva Vieira
Prof. Joaquim Paul Laurens Viegas

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. Susana Margarida da Silva Vieira
Members of the Committee: Prof. Agostinho Cláudio da Rosa
Prof. Francisco João Duarte Cordeiro Correia dos Santos

November 2019

Dedicated to someone special

Acknowledgments

Antes de mais peço desculpa por estar a fazer os agradecimentos em português, mas acredito que é mais apropriado referir estes agradecimentos de uma maneira mais genuína. Gostaria de agradecer a todos aqueles que de alguma maneira me ajudaram a que esta tese fosse possível. Começando com os meus orientadores, Prof. Susana Vieira e Prof. Joaquim Viegas, que me ajudaram a estrutura ideias que mais tarde seriam bastante importantes para o desenvolvimento desta tese. Ao Prof. João Sousa que apesar de não ser meu orientador oficial teve um papel igualmente importante no desenvolvimento da tese. Gostaria de agradecer também a Worten e ao Pedro Costa que me permitiram fazer a tese com eles e me ensinaram o funcionamento pratico onde toda a tese se assenta. A minha família e amigos que apesar de não me terem dado um apoio tão direto no desenvolvimento da tese estiveram lá para todas as ocasiões essenciais. Para concluir gostava de mandar um grande agradecimento ao Miguel Martins que não tendo nenhuma obrigação perante esta tese perdeu inúmeras horas comigo seguindo de raiz toda a tese, discutindo comigo quase todo os pontos da mesma e revendo a tese na sua totalidade. Acredito que o Miguel tem um potencial imenso como orientador e como professor. Não tenho qualquer duvida que o seguimento do Miguel foi uma das ajudas mais importantes para o desenvolvimento desta tese.

Resumo

Esta tese resolve uma variante do problema clássico ”Vehicle Routing Problem (VRP)”, que surgiu de um problema real específico, ainda não resolvido, proposto pela empresa Worten. O problema foi formulado matematicamente e analisado para ser otimizado. O problema pode ser classificado como ”Site-Dependent Vehicle Routing Problem With Hard Time Windows (SDVRTHTW)”, e precisa de ser resolvido diariamente em duas ou três horas. Para resolver essa variante, foram propostos, testados e modificados dois algoritmos diferentes: a Pesquisa Local e um Algoritmo Genético Híbrido com a Pesquisa Local. Uma adaptação das técnicas de Clarke and Wright Heuristic foi usada para inicializar os algoritmos de pesquisa local e híbrido.

O objetivo é encontrar a melhor combinação de rotas que permita reduzir os custo, servindo todos os clientes da empresa sempre garantido que as restrições do problema real, como as restrições rodoviárias da UE, não sejam violadas.Os algoritmos foram testados e implementados em três semanas diferentes do ano em que as quantidades dos diferentes clientes da Worten são baseadas em previsões e onde algumas rotas são sugeridas e analisadas pela empresa trasportadora sendo possível fazer uma comparação entre as rotas feitas pela empresa e as rotas feitas pelos algoritmos propostos.

Ambos os algoritmos apresentam melhores resultados do que o conjunto de rotas proposto pela empresa, o que sugere que o uso de algoritmos de planejamento de rotas para o problema da Worten diminui substancialmente os custos de entrega sem violar os constragimentos.

Palavras-chave: Problema De Roteamento De Veículos, Local Dependente, Janela Temporal Fixa, Algoritmo Genetico, Pesquisa Local

Abstract

This thesis solves a variant of the classic Vehicle Routing Problem (VRP), a variant which emerged from a specific and not yet solved real-world problem proposed by the company Worten. This problem has been analyzed and formulated mathematically so that it can be optimized. The problem can be called Site-Dependent Vehicle Routing With Hard Time Windows (SDVRPHTW), and needs to be solved in two or three hours every day. In order to solve this variant two different algorithms were proposed, tested and modified: a Local Search and a Hybrid Genetic algorithm with Local Search. An adaptation of the Clarke and Wright Heuristic was used to start both the Local Search and the Hybrid Algorithms.

The goal is to find the best combination of routes that allows to spend the least amount of money to supply all the customers of the company, whilst always guaranteeing that the restrictions of the real problem, such as the EU road restrictions, are not violated. The algorithms were tested and implemented in three different weeks of the year where the demands of the different customers of Worten are forecast and where some routes are suggested and analyzed by the shipping company in order to make a comparison between the routes made by the company and the ones given by the algorithms.

Both algorithms give better results than the set of routes proposed by the company. This suggests that the use of route planning algorithms for the Worten problem substantially decreases delivery costs without violating the constraints.

Keywords: Vehicles Routing Problem, Hard Time Window, Site Depended, Genetic Algorithm, Local Search

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

η Vehicle efficiency.

λ Longitude.

ϕ Latitude.

Roman symbols

a_i Pallets that were not supply to customer i.

B_i^k Exact time of service at each point i by vehicle k, 0 if vehicle k did not supply customer i.

C_k Quantity supply to customer i when j is supply after i by a vehicle k.

C_{pal} Cost of a vehicle k.

C_{stop} Cost of a vehicle making a stop in a customer.

d_i Demand of customer i.

D_i^k Binary variable, 1 if vehicle k can supply customer i, 0 otherwise.

E_i Latest time that a customer i can be supply.

L_i Latest time that a customer i can be supply.

O_k Cost of a vehicle k do an open route.

Pk Maximum number of pallets that may not be supply in one route.

Pt Maximum number of pallets that may not be supply in total.

Q_k Maximum capacity of vehicle k.

s_i Pallets that were supply to customer i.

T_f Fixed time that a vehicle takes when supplying a customer.

T_v Variable time that a vehicle takes when supplying one customer.

t_{ij}^k Time that a vehicle k takes to go from customer i to customer j.

X_{ij}^k	Binary variable, 1 if j is supply after i by a vehicle k, 0 otherwise.
Y_{ij}^k	Quantity supply to customer i when j is supply after i by a vehicle k.
R	Earth's radius.
d_{ij}	Distance to go from customer i to customer j.
l_i	Latest possible time that the costumer i can be served.
S_{ij}	Saving matrix of insert a costumer i on the route of the customer j.
f_i	Fitness of the chromosome i.
P_{sel_i}	Probability of a chromosome i being selected

Glossary

ACO	Ant Colony Optimization
B&B	Branch And Bound
B&C	Branch And Cut
CVRP	Capacitated Vehicle Routing Problem
DARP	Dial-A-Ride Routing Problem
EPT	Earliest Possible Time
FSM	Fleet Size And Mix
GA	Genetic Algorithm
HFVRP	Heterogeneous Fleet Vehicle Routing Problem
HGA	Hybrid Genetic Algorithm
HVRP	Heterogeneous Vehicle Routing Problem
ILP	Integer Linear Programming
LPT	Latest Possible Time
LP	Linear Programming
LS	Local Search
MDVRP	Multi-Depot Vehicle Routing Problem
NN	Nearest Neighbor
PVRP	Periodic Vehicle Routing Problem
SDVRPHTW	Site-Dependent Vehicle Routing Problem With Hard Time Windows
SDVRPTW	Split Delivery Vehicle Routing Problem With Time Windows
SDVRP	Site-Dependent Vehicle Routing Problem
SDVRP	Split Delivery Vehicle Routing Problem
SVRP	Stochastic Vehicle Routing Problem
TSP	Traveling Salesman Problem
TS	Tabu Search
VRPB	Vehicle Routing Problem With Backhauls
VRPPDTW	Vehicle Routing Problem With Pickup And Delivery With Time Windows

VRPPD	Vehicle Routing Problem With Pickup And Delivery
VRPSPD	Vehicle Routing Problem with Simultaneous Pickup And Delivery
VRPTW	Vehicle Routing Problem With Time Windows
VRP	Vehicle Routing Problem

Chapter 1

Introduction

The transportation of goods is one of the most important aspects in the industrial sector, also known as vehicle routing problem (VRP). The transportation sector normally has great competitiveness which leads to a constant need of optimization to ensure competitive advantage [2].

Transport costs are of great importance in logistics costs so if transport costs go down, so do logistics costs, for that reason the VRP has assumed a major importance in operational research.

Most of the VRP applied to realistic problems, also called rich VRP [2], are highly complex and need to optimize more than one variable while fulfilling a high number of particular restrictions, which makes the problems highly non feasible. Using a manual solution is not advised if competitive solutions are desired, to obtain competitive results it is necessary to make use of computational power applied to the specific problem.

1.1 Objectives and Contributions

Worten is a company of household appliances that needs to supply its different sales posts daily. The customers that need to be supplied are not only the stores of Worten but also other stores from the parent group, Sonae. The sales post are divided in 7 different areas all over Portugal, having a higher concentration in the Lisbon area. To supply the different customers the fleet of a shipping company is subcontracted. The fleet of the shipping company it's composed by different vehicles, but only the bigger ones are going to be used, these have a maximum capacity of 33 pallets.

Worten needs to increase the efficiency of the subcontracted vehicles in order to decrease the transportation costs. For this, all the different parts of the delivery process were studied and two algorithms were design having into account all the different restrictions. The algorithms were programmed in phython from scratch since the restrictions are specific to this issue.

The main contribution were the designed of two algorithm that are able to improve the routes made and we believe that those algorithms are also able to give competitive results in different vehicle routing problems variations, mostly the Hybrid algorithm due to this one be a very flexible algorithm, so we believe that with the right parameterization and formulation the Hybrid algorithm is able to solve most of the VRP's variations.

1.2 Thesis Outline

In this chapter a brief explanation of the necessity and importance of this thesis was made, also a summary of each chapter is presented in this section, as a way to sum up the work done.

At chapter (2) an introduction to the VRP is made followed by an explanation of the different VRP variations. After diverse ways to solve the different VRP are introduced, namely exact methods, heuristics and metaheuristics.

At chapter (3) the different characteristics of the problem are introduced, or in other words, how do the different parts of Worten transportation system works. After a comparison with the theoretical models introduced in chapter (2) a mathematical formulation of the problem is introduced. Finally, a small example to provide some sensibility of the model is stated.

In chapter (4) a first proposed algorithm is introduced and after, two theoretical ways to solve the problem stated in chapter (3) are described and explained, a Local Search and a Genetic Algorithm. In the end of chapter (4) a final proposed algorithm is introduced where a Hybrid Genetic Algorithm is used.

In chapter (5) the baseline solutions made by the company are introduced, followed by a parameterization and modification of the algorithms introduced in chapter (4) and finally a comparison is made between the results obtain using the algorithms and the results of the baseline solution.

Finally in chapter (6) a little summary of the main conclusions achieved in this master thesis is made, followed by the proposed future work.

Chapter 2

Vehicle Routing Problem (VRP)

2.1 Introduction to the VRP

The Vehicle Routing Problem (VRP) was first introduced in the literature by Dantzig and Ramser in 1959 [3]. This type of problem consists in delivering a product from a depot center, to different customers in different spacial places, subject to side constraints(time window, capacity, etc.). Some of the different variations of the classic VRP can be see in the figure bellow:

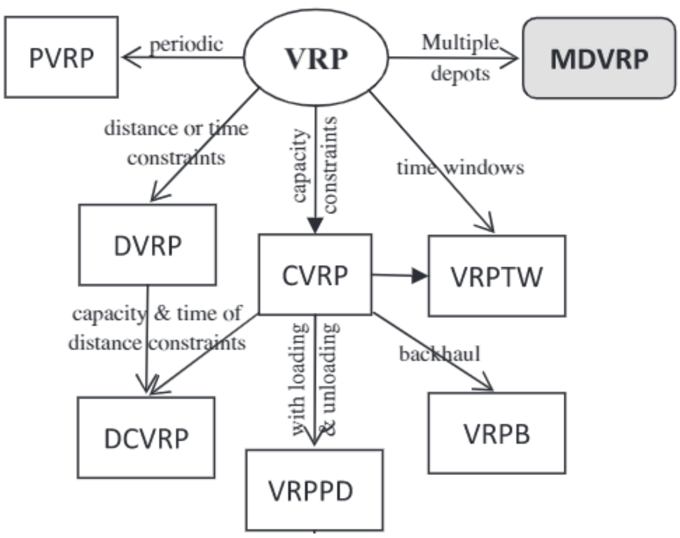


Figure 2.1: Different types VRP from Montoya-Torres et al.[4]

In the classic VRP each customer has a deterministic and known demand and the objective is, with a homogeneous fleet of vehicles, to minimize the total distance traveled.

This type of problem is one of the most studied problems in the field of operations research, and since the VRP is an NP-hard problem , Lenstra and Kan [5], exact algorithms are only efficient for small size problems. For problems of a larger scale, normally in practical applications, heuristics and metaheuristics are more suitable approaches.

Formulation: There are different places where the formulation can be found in the literature, the following formulation is from Cordeau et al. [6].

The Symmetric VRP is defined by a Graph $G = (V, E)$, where the $V = \{0, ..., n\}$ is a vertex set, each vertex except 0 ($i \in V \backslash \{0\}$) represents a customer and 0 corresponds to the depot. Each edge ($e \in E = \{(i,j) : i,j \in V, i < j\}$) is associated with a travel cost C_{ij}. A fixed flee of *m* identical vehicles with a capacity *Q* is available at the depot. The objective is to minimize the total travel cost with the constrains that: (1) each customer can be visited exactly once, (2) each route begins and ends in the depot, (3) the total demand of the route does not exceed the capacity *Q* and (4) the length of a route is smaller that the preset limit, L.

Some of the most common and relevant variation of the classic VRP will be introduced and explained in the rest of this chapter. At the end of the chapter some methods to solve the VRP and its variants will be introduced.

2.2 Different types of vehicle routing problems

2.2.1 Capacitated Vehicle Routing Problem (CVRP)

The Capacitated Vehicle Routing Problem (CVRP) is one of the most simple variations of the VRP. The CVRP incorporates an additional constraint where, every vehicle has an uniform capacity and must be homogeneous. All vehicles must start and end at the depot and each customer cannot be visited more than once.

The literature on this topic is very extensive and a comprehensive formulation of the problem can be find in Toth and Vigo [7], where the goal is to minimize the total distance traveled. Both exact methods such as Branch and Bound algorithms [7, 8] and metaheuristics methods, such as Genetic [9] and Tabu Search [10], were used to solve this variation of the VRP.

2.2.2 Multi-Depot Vehicle Routing Problem (MDVRP)

The Multi-Depot Vehicle Routing Problem (MDVRP) is also an extension of the VRP where multi depots are allowed instead of only one. The MDVRP is a very important problem due its similarity with many real world scenarios. The MDVRP was first introduced in 1972 by Wren and Holliday [11] and it was solved using different methods such as Genetic Algorithm (GA) (Liu et al. [12]) and Tabu Search (TS) [13, 14]. In Montoya-Torres et al. [4] a complete state of the art for the MDVRP is presented, using heuristics, metaheuristics and exact models.

The mathematical formulation of the MDVRP can be found in Montoya-Torres et al. [4] and the extension constraint to the VRP are: (1) there is more than 1 depot node; (2) the number, location and the number of vehicles in each depot are predetermined; (3) the vehicles have to return to the same depot after doing the route assigned; Generally, the objective is to minimize the total delivery distance.

In Cordeau et al. [14] can be seen that the MDVRP can also be formulated as a special case of the Periodic Vehicle Routing Problem (PVRP).

2.2.3 Periodic Vehicle Routing Problem (PVRP)

The Periodic Vehicle Routing Problem (PVRP) is once more a generalization of the VRP where an horizon of *t* days is planed and, each customer has a predefined set of allowable combinations of visit days (Cordeau et al. [14]), instead of a single day (example: $Combinations = \{\{1,3\},\{2,4\}\}$). In each day: (1) all vehicles begin and end their day at the depot. (2) The demand of each customer is deterministic and know. (3) The number of vehicles is know and have a limited capacity. The main objective is to minimize the total traveling cost.

The PVRP was first introduced in Beltrami and Bodin [15] in 1974. Only in 1979 the problem is formally defined by Russell and Igo [16], and the mathematical formulation appear 5 years later by Christofides and Beasley [17]. In the begging, the PVRP was more focused on garbage collection from different centers.

The early paper publications used heuristics to solve the problem [15–17]. Those heuristics sometimes assign customers to days before routing them, and sometimes create routes and then attempt to assign these routes to days [18]. In the paper ”Forty years of periodic vehicle routing” of Campbell and Wilson [18] are enunciated a set of papers that solve the PVRP using heuristic (some of them were previously enunciate) and a set of papers that solve the PVRP using metaheuristics, such as Tabu Search (Cordeau et al. [14]).

2.2.4 Split Delivery Vehicle Routing Problem (SDVRP)

There is a variant of the VRP where each customer is allowed to be visited more than once, this variant is called Split Delivery Vehicle Routing Problem (SDVRP). In the SDVRP the demand of a customer can be greater than the capacity of the vehicles, the demand of all the customers visit by a vehicle must be smaller that its maximum capacity and each vehicle must start and end in the same depot. The main objective is to minimize the total distance travelled. The SDVRP is necessary when the maximum capacity of the vehicle is small and when a customer has a bigger demand that the maximum capacity of the vehicle [19].

The SDVRP was proposed by Dror and Trudeau [20] in 1989 and a Local Search was used to find some results and showed that savings can be generated by allowing split deliveries. A mathematical formulation was presented by Archetti and Speranza [19] and in the same paper is shown that split delivery can save up to 50% of the cost. The SDVRP can have multiple variants that are enunciated in ”Vehicle routing problems with split deliveries” by Archetti and Speranza [21], that are basically the combinations of the SDVRP with other VRP’s variations, for example SDVRP with Time Windows (SDVRPTW).

In the literature the SDVRP is solved in different ways, Archetti et al. [22] uses a metaheuristic (in this paper is only called heuristic) and in Archetti et al. [23] Branch-and-cut algorithms are used to find the exact solution, those are only able to solve very small instances. In Archetti and Speranza [21] are enunciated the different ways of solving the SDVRP in the literature, using both heuristics and exact models.

2.2.5 Stochastic Vehicle Routing Problem (SVRP)

The Stochastic Vehicle Routing Problem (SVRP) is a VRP with the difference that some of the variables are not know, this is, some variables are random. The most common SVRP is stochastic demand, where the demand of the customer is characterized by a probability and this demand is only know when the vehicle arrives at that customer. There are also two big variations, the stochastic customer where each vertex (v_i) is presented with a probability p_i, and the stochastic travel time where each arc (v_{ij}) is also characterized with a probability. The main objective of the SVRP is to minimize the expected value of the cost [24, 25].

The SVRP was first introduced in 1969 in "The multiple terminal delivery problem with probabilistic demands" by Tillman [26], normally the SVRP is divided in 2 phases. In a first phase, an "a priori" solution is determined. In the second phase a corrective action is then applied to the first phase solution if necessary. Normally the corrective action is coming back to the depot to load/unload when the capacity of the vehicle is not respected [25]. This happens because due to randomness, the planned route may not be feasible. Note that large routes will tend to more frequent route failures but less distance traveled, so a compromise is necessary. Two variants of the stochastic programming are used, the Chance-constrained programming that involves the replacing of a deterministic constraint by a set of change-constraints (used when the violation of the capacity is not well defined and, can be modify to be solve by deterministic programming) and stochastic programming with recourse where the cost of the violated constraint is considered in formulating the problem. These formulation can be seen in Stewart Jr and Golden [24].

The stochastic demand's is the most studied of all SVRP, and it was solved in the begging by Tillman [26] using an adaptation of the Clarke and Wright [1] heuristic algorithm. In Reimann [27] an adaptation of the Ant Colony Optimization (ACO) is used to solve the SVRP and in Stewart Jr and Golden [24] the authors show the formulation and the heuristics used to solve the stochastic demand's. In Laporte et al. [28] the stochastic travel times are defined and a Branch and Cut (B&C) algorithm is used to solve the different formulations proposed.

2.2.6 Vehicle Routing Problem with Backhauls (VRPB)

The Vehicle Routing Problem with Backhauls (VRPB) is an extension of the VRP that involves both deliveries (linehauls, $L = \{v_1, ..., v_l\}$) and pickups (backhauls, $B = \{v_{l+1}, ..., v_n\}$) nodes. All the linehauls must be served before the backhauls by a fleet of homogeneous vehicles "This is caused by the fact that the vehicles are rear-loaded and rearrangement of the loads on the trucks at the delivery points is not deemed feasible." (Goetschalckx and Jacobs-Blecha [29]), also the lineshauls normally have a higher service. The demand both for the linehauls and the backhauls is known and, for each route the total load associated must not exceed, separately, the capacity of the vehicle. The main objective is to minimize the total distance traveled by the fleet (Toth and Vigo [30]).

The VRPB has a great practical and theoretical importance due to its frequency in real life situations [30, 31]. The VRPB allows significant cost savings due to the utilization of the capacity of the vehicle after all the delivers are made, something that is not done in the VRP. The mathematical formulation and the relaxations of the formulation can be seen in Toth and Vigo [30].

It is easy to find in the literature the VRPB solved in many different ways. Deif and Bodin [32] proposed an extension of the Clarke and Wright [1] algorithm. Goetschalckx and Jacobs-Blecha [29] used a space-filling curve algorithm to solve the VRPB. An exact method, using Branch and Bound (B&B) was proposed by Toth and Vigo [33] and in Zachariadis and Kiranoudis [31] and Gajpal and Abad [34] some metaheuristics can be found to solve the VRPB.

2.2.7 Vehicle Routing Problem with Pickup and Delivery (VRPPD)

The Vehicle Routing Problem with Pickup and Delivery (VRPPD) can be seen as a generalization of the VRPB (Desaulniers et al. [35]) and subsequently a generalization of the VRP. The VRPPD consists in picking up goods from a certain location, and drop those goods at their destination. Therefore it is necessary that the pick-up and the drop-off are made by the same vehicle and in the same route (coupling), and that pick-up is done before the drop-off (precedence). There are also depot constrains to assure that the vehicles return to the appropriate depot (Desaulniers et al. [35]).

The VRPPD was first introduced by Min [36] in 1989 and is a very broad and general problem. The VRPPD can be formulated as a graph, this formulation can be seen in Berbeglia et al. [37]. All the variations of the VRPD can be defined by 3 fields (Structure—Visits—Vehicles), the definitions of the fields and the variations are explain in Berbeglia et al. [37], and in Parragh et al. [38] are stated the different know variations of the VRPD. The mathematical formulations of two special and very study cases, VRPPD with Time Windows (VRPPDTW) and VRP with Simultaneous Pickup and Delivery (VRPSPD), can be found in Desaulniers et al. [35] and in Dethloff [39] respectively. It is also important to stress that the VRPPD is normally associated with the delivery of goods, when it comes to delivering people it is called Dial-A-Ride Problem (DARP), this is again a problem that has some importance in the literature due to its practical applications (Berbeglia et al. [37]).

2.2.8 Vehicle Routing Problem with Time Windows (VRPTW)

A popular extension of VRP, the Vehicle Routing Problem with Time Windows (VRPTW), consists in serving a set of customers with an homogeneous fleet where all the nodes (customers) have a specific time window ($[a_i, b_i]$). If a vehicle arrives before the Earliest Possible Time (EPT),a_i, it must wait in the customer location, in other hand a vehicle cannot arrive after the Latest Possible Time (LPT),b_i, for the solution to be feasible. All customers are assigned to only one vehicle and the vehicles cannot exceed their maximum capacity. The main objective is normally to first minimize the number of vehicles, followed by minimizing the total traveling time [40–42].

Solomon [43] was the first to introduced the VRPTW in the literature in 1987, and used a variation of the Clarke and Wright [1] to solve it. In the VRPTW, the time window constraints can be divided in soft constraints, where a vehicle is able to arrive after the LPT but a penalization is added to the objective function (Taillard et al. [44]), and in hard constraints here the vehicle must arrive before the LPT for the solution to be feasible (Cordeau et al. [45]). The mathematical formulation for the VRPTW can be found in Cordeau et al. [45].

A vast literature can be found about the VRPTW due to being ”an important problem occurring in many distribution systems” [46]. Bräysy [47] also used an heuristic to solve the VRPTW that gave good results according to Bräysy and Gendreau [46]. Homberger and Gehring [42] used an hybrid algorithm to solve the VRP, and in Taillard et al. [44] the VTPW with soft constrains is solved using a Tabu Search algorithm.

2.2.9 Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

Another variation of the VRP, is the Heterogeneous Fleet Vehicle Routing Problem (HFVRP). The HFVRP has the difference that the fleet is heterogeneous, this is, the fleet is composed by vehicles that are allowed to have different costs and capacities (Baldacci et al. [48]). In the HFVRP the customers can only be served by one vehicle and the total demand of the customers visited by a vehicle must not exceed the vehicle capacity. Customers can also have restriction on the types of vehicle that are allowed to visit, this is called Site-Dependent VRP (SDVRP) (Baldacci et al. [48]). The main objective of the HVRP is the minimization of the total routing cost (Golden et al. [49]).

The HFVRP is considered a ”rich” VRP, more similar to real-life problem (Baldacci et al. [48]), HFVRP increase flexibility in distribution planning (Penna et al. [50]) and fleets in the industrial sector are rarely homogeneous (Penna et al. [50]). There are different variations for the Heterogeneous problem, those are enunciated in Baldacci et al. [48]. The most important Heterogeneous problems are the HFVRP with unlimited fleet, also know as Fleet Size and Mix (FSM) first introduced in Golden et al. [49] in 1984, and the HFVRP with limited fleet, also know as Heterogeneous Vehicle Routing Problem (HVRP) introduced by Taillard [51] in 1999. The complete mathematical formulation can found in Baldacci et al. [48].

In the literature both Golden et al. [49] and Taillard [51] used heuristics to solve the HFVRP. Semet and Taillard [52] used a Tabu Search method to solve the HFVRP with unlimited fleet, and in Baldacci et al. [48] can be seen how different authors using both heuristics and metaheuristics solve the FSM and the HFVRP.

2.3 Different ways to solve the vehicle routing problems

2.3.1 Exact Methods

Exact algorithms are the ones that search all the research space until the optimal solution is found. These algorithms often require large computational time, which generally makes it impossible to solve large-scale problems. Normally VRP's are modeled and solved using Integer Linear Programming (ILP) [53]. Some of the most common exact algorithms to solve the VRP are going to be explain next.

Branch and bound (B&B)

The Branch and bound (B&B) is an algorithm that was first introduced in 1960 by Land and Doig [54]. This algorithm has the objective of solving discrete and combinatorial optimization problems and find the optimal solution [55]. The B&B consists in dividing the initial problem in sub-problems and solving each using some relaxations such as Assignment Problem, Shortest Spanning Tree and more recently Lagragian Relaxations (Toth and Vigo [7]). The B&B is divided in 3 parts the Branching, select the sub-problem that was creating more recently; the Bounding, for each sub-problem find the solution applying the relaxation; and Fathoming, compare the value obtain with the bounds and discards the ones that do not fit it. When all the sub-problems are solved the lower bound is the optimal solution (Hillier and Lieberman [56]).

The B&B is solved more efficiently the better the lower bound is calculated because less sub-problems fit the Fathoming condition. Some of the relaxations and lower bounds can be seen in Toth and Vigo [7], where the B&B was used extensively to solve the CVRP. In Laporte and Nobert [57] a VRP is solved using B&B and in Toth and Vigo [33] an B&B algorithm is used again to solve the VRPB.

Branch and Cut (B&C)

The Branch and Cut (B&C) is an extension of the classic B&B, that is normally used when the number of constraints is big enough that the relaxation of the problem can not be solved using a simple Linear Programming (LP) algorithm and a Cutting Plane technique has to be used to solve the LP (Naddef and Rinaldi [58]). The Cutting Plane Technique consists in using a subset of the initial problem and solving the LP relaxation. If the LP is not feasible in the original problem, a new constraint is created, the size of the subset is increased and a new sequence is made until the LP relaxation is feasible in the original problem and the optimal is found (Naddef and Rinaldi [58]). The main objective of the Cutting Plane Technique is to add new constraints near the optimal feasible solution which makes the problem easier to solve. If an optimally solution is not found the problem is divided in two, for example doing an upper and lower bound to a variable, and the (B&B) algorithm is applied.

In Laporte et al. [28] it is possible to see the SVRP being solved using the B&C algorithm, reaching a optimal solution in different scenarios with a maximum of 20 cities. In Archetti et al. [23] a B&C algorithm is also used to solve the SDVRP and the optimal solution can be found for a problem with 100 cities. New upper bounds were found in different proposed problems.

2.3.2 Heuristics

Heuristics are algorithms that only exploit a subset of all the possible solutions and because of that near-optimal solutions are not always found. For the reason previously explained, heuristics algorithms find good solutions in reduced computational time, so large problems can be solved in reasonable time.

The heuristics for the Traveling Salesman Problem (TSP) can be divided in 3 types, and this division can also be apply to the VRP: construction procedures, improvement procedures and composite procedures [46, 59]. Some of the most common algorithms for the VRP will be explained next.

Saving heuristic of Clarke and Wright

The Saving heuristic was first introduced by Clarke and Wright [1] in 1964. This heuristic was originally developed for the VRP and is one of the best route construction algorithms in the literature [46]. The Saving heuristic begins with all the customers being served individually by a dedicated vehicle. After is calculated what is the saving of combining two routes i and j, $S_{ij} = d_{i0} - d_{0j} - d_{ij}$, and are connected the two customers that have the bigger S_{ij}, always ensuring the feasibility of the solution. This process is done iteratively until there are no more positive savings. The savings can only be applied to customers that are connected to the depot, not being possible to use nodes in the middle of the route. The algorithm can be used to improve the total distance traveled, by forming partial routes, or to minimize the number of vehicles used, in this case the algorithm is used until the vehicle is fully loaded.

Even though the Clarke and Wright method was developed for the VRP, many authors extended the algorithm to be used in different problems. Solomon [43] improved the algorithm to be used in the VRPTW by adding a parameter of the time window of the customers in the saving cost, S_{ij}, taking into account that two costumers can be close in space but distant in time. Tillman [26] and Deif and Bodin [32] also adapted the algorithm to solve the SVRP and the VRPB respectively. In 2007 Liu et al. [12] used the Saving heuristic to do the initialization of a Hybrid Genetic Algorithm to solve the MDVRP.

Nearest Neighbor (NN)

The Nearest Neighbor (NN) is an algorithm that starts every new route by finding the unsigned customer that is closer to the depot, in each iteration the NN finds the closer unsigned customer that can form a feasible solution from the last signed customer and adds that node to the end of the route. When there are no more feasible solutions a new route is started in the depot until all the costumers are already signed to a route. A variation of the NN, Time-Oriented Nearest-Neighbor Heuristic, is formulated for the VRPTW where both the distance and the temporal closeness are considered, two costumers can be close in space but distant in time (Bräysy and Gendreau [46], Solomon [43]). In Liu et al. [12] the NN algorithm is used to initialize the first generations of a Genetic Algorithm.

Local Search (LS)

The Local Search (LS) is an heuristic based on improving the current solution iteratively by exploring the neighboring space. To design the algorithms is necessary to first generate an initial solutions, after is necessary to know how the exploring will be done and finally what is the stopping criteria [46]. Normally the exploring is done by applying k-exchange, replacing k edges by another k edges. In Shaw [60] a Local Search in applied to the classic VRP and good results are achieved, in Potvin and Rousseau [61] different Local Search algorithms have been tested for the VRPTW also achieving good results. In

Bräysy and Gendreau [46] an extensive research on the Local Search, mostly applied to the VRPTW, can be found.

2.3.3 Metaheuristics

The metaheuristics are algorithms that try to explore local improvements by exploring the current best solutions and at the same time try to escape from local optima by looking for new solutions in the search space. The metaheuristics as a concept were first introduced in 1986 in Glover [62]. Due to their nature these algorithms are able to provide near-optimal solutions to large problems in reasonable computing time (Gendreau et al. [63]).

Genetic Algorithm (GA)

The Genetic Algorithm (GA) was developed in the 60′s by Holland et al. [64] and is a stochastic optimization technique. The GA is based on natural evolution following Darwin's theory (Nazif and Lee [9]), where a population of individuals are maintained and a reproductive process occurs where the individuals with the best fitness are more likely to survive and reproduce (Baker and Ayechew [65]). First of all to start the GA is necessary to represent each individual by a string (chromosome), which is one of the critical issues (Anbuudayasankar et al. [66]). The chromosomes with the best fitness are more likely to be chosen to generate new solutions (offspring), this process is called selection. To generate those solution normally two operations are used, crossover and mutation. The crossover attempts to combine the genetic information of two parents, so the offspring have information from both parents and that combination can lead to solutions with better fitness. The crossover can be compared to a Local Search (LS) (Ho et al. [67]) and sometimes the crossover is replaced by a LS this hybridization is called Memetic (Prins [68]). The mutation operation is the one that normally avoids the convergence to a local minimum by maintaining the diversity (Holland et al. [64]). This operation normally selects one chromosome and changes part of its original state, one classic mutation is to change a bit to its inverse. In every iteration exists a population of individuals that are used to reproduce, and the individuals with the best fitness will tend to be in the next generation, with this the mean fitness of the population will tend to improve in every iteration.

The Genetic Algorithms applied to the VRP's are not one of the most used methods in the literature. In Baker and Ayechew [65] a GA is used to solve the classic VRP achieving results almost as good as using the Tabu Search (TS). In Chand et al. [69] a Genetic Algorithm is used to solve the VRP with multi objective achieving high quality solutions. In Ho et al. [67] an hybrid GA was used to solve the MDVRP and concluded that initializing the population using a constructive method allows to achieve better results. Tasan and Gen [70] also used the GA to solve the PDVRP performing well and efficiently.

The GA shows to be a competitive method to solve large combinatorial problems in terms of time and solution quality (Baker and Ayechew [65]) and can be a simple and effective method (Prins [68]). The GA is also an algorithm that that is well suited for multi objective optimizations problems.

Tabu Search (TS)

The Tabu search (TS) was first introduced in 1989 in Glover [71] and is a local search methaheuristic designed to solve combinatorial problems. Tabu Search is based on looking for the best solution in a subset of its neighborhood, $N(s)$ of the currently solution S. The solution may be worse from one iteration to the other, this happens because there is a list of temporary forbidden moves to avoid cycling, *tabu list* (Cordeau et al. [14]). Some moves are possible even if they are in the tabu list if they reach some aspiration level, normally the aspiration level is reached when the current move is better than the best solution found yet (Cordeau et al. [14]). The moves are normally divided in two categories, *intensification* where the current solution is improved and *diversification* where a new neighborhood is explored, for example by penalizing the most frequent moves. Finally an initialization is normally used to begin the algorithm, a generalized insertion procedure, GENI heuristic, is some times used for this purpose (Archetti et al. [22]).

The TS is one of the most used algorithms to solve the VRP variations, since the VRP can be formulated in a way that the use of the TS is almost straightforward. The Tabu Search is also very used on the VRP variations due to be one of the algorithms that gives better results. In Cordeau and Laporte [72] the TS is used to solve the PVRP and the SDVRP, in Cordeau et al. [14] a PVRP and a MDVRP are also solved. In Archetti et al. [22] a Tabu Search is used to solve the SVRP. All the TS literature previously listed achieves good results.

The TS is a complex and time spending algorithm even if is one of the most used methods to solve the VPR's, so a difficult implementation with various set up parameters, like the way to use the taboo list, is necessary to achieve good results for the solution not to get stuck in a local optimal.

Ant Colony Optimization (ACO)

The Ant Colony Optimization (ACO) is a population based approach that was first introduced by Dorigo et al. [73] in 1996 and is based in real ant behavior in the search for food (Reimann [27]). The ACO is based on having a population of artificial agents that leaves pheromones when they move, the more the ants follow a trail the more attractive it becomes, more pheromones will be left in that trail (Dorigo et al. [73]). The probability of choosing a solution is normally based on the pheromones and on some heuristic information, gradually the weight of pheromones gets stronger and all the ants tends to to converge to the same solution (Balseiro et al. [74]).

The ACO was already been used is different variants of the VRP showing successful results (Reimann [27]). In Tang et al. [75] a variant of the Ant Colony is used to solve the SPVRP, in Reimann [27] the algorithm is used to solve the SVRP and in Balseiro et al. [74] and Kalayci and Kaya [76] the VRPB and the VRPPD are solved respectively.

The ACO is not advised to be used for the VRP when the number of vehicles are fixed, for that two types of ants must be used, one for the number of vehicles and another for the route total length tour, also called cluster first and route second (Gajpal and Abad [34]). An hybridization with Local Search is also some times used to improve the results [34, 74, 76].

Chapter 3

Problem formulation

As it was explained before the main objective of this thesis is to solve a real routing problem proposed by the company Worten in about 2/3 hours. This problem is specific and not yet solved, for that reason is necessary to know in detail how does the different parts of the logistics works to be possible to define and model the problem, and with that solve it.

In the rest of the chapter is going to be introduced the different characteristics of the optimization problem, the logistics, the customers, the pallets on the depot, the reverse logistics and the feasibility of the solution. After a comparison with the theoretical models introduced in chapter (2) is made, and a mathematical formulation of the problem is introduced. Finally, in the end of the chapter, a small example to gain some sensibility of the model is done.

3.1 Problem Description

3.1.1 Logistics

The first thing that is important to do, before starting the construction of the algorithm, is to define the problem. For that the transportation logistics is going to be analyzed.

The main objective of Worten is to minimize the total transportation costs. Minimizing this cost is equivalent to minimizing the fee paid to the company that is subcontracted to make the deliveries. To be able to fulfill this objective it is necessary to know how the shipping company is subcontracted. The fleet of the shipping company is composed by different vehicles, those vehicles have a maximum capacity of 33 pallets and can also have or not a support platform. This platform is mandatory if the customer does not have one of his own, this is, if a customer does not have this platform the vehicle that visits this customer must have this component.

Three subcontracting models were studied:

- The company only pays for the pallets that are shipped. In this case all the risk of bad route planning is on the shipping company because the inefficiency is not being payed. This results in a larger tariff for the pallets shipped by thw shipping company;

- The company pays for the pallets shipped with the difference that a dynamic adjustment is made according to the inefficiency. In this case the risk of bad route planning is both Worten and the shipping, because the pallets are paid by the unit, but if the vehicles are almost empty an extra price must be included. This make the tariff smaller than in the first case;

- The company pays the full truck (Full Truck Load). In this case all the risk of bad route planning is in Worten because the inefficiency of the vehicle have a direct impact on the price of each pallet, because the price of the vehicle is the same if the vehicle is fully loaded or not.

The type of subcontracting that is going to be implemented to build the routes is the third one. The Full Truck Load is going to be used because the main objective is to find a way to optimize the price paid to the shipping company. If the first hypothesis was chosen there would be no need to optimize the inefficiency of the vehicles. If the second hypothesis was chosen there would be only need to optimize the average monthly inefficiency. Choosing the third hypotheses it is guaranteed that the price per pallet is going to be minimized regardless of the type of tariff chosen.

The vehicle prices depend on different characteristics, the zone that the truck goes (it is assumed, that if he visit more than one zone, the price is of the more expensive one), the number of stops made and if it is an open or a close route, if the vehicle returns to the depot or not, a more complete explanation is made in the subsection (3.1.5).

3.1.2 Customers

The customers that need to be supplied by the company are the different stores of Worten, and not only, that are divided all over the country. The demand of the customers is the products that are order by each store to the depot. The fleet that is subcontracted must supply a total of at most 713 customers. Each customer is located in a different place in Portugal, that can be obtained by its Geo-reference (latitude, longitude), and they are divided in 7 zones, as said before the price of the vehicle depends on the zone that each trucks goes, the different zones can be seen in figure (3.1):

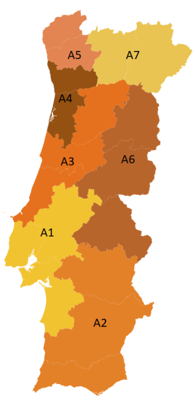


Figure 3.1: Division of zones in Portugal regarding vehicle prices

The most expensive zone is the zone A7 that contains Bragança and Vila Real, followed by the zone A5 that contains Braga and Viana do Castelo. The cheapest zone, is the zone A1 which contains Lisboa and Santarem. The other zones all have the same prices with a small variance, and the value is around the middle of the prices of the cheapest and more expensive zones.

The retail business is very sensitive to seasonality. According to Worten data the demand of the customers follow a curve that can be divided in 4 big periods. In the beginning of the year the demand of each store is lower that the mean of the year and the curve decreases gradually until the summer. In the beginning of the summer an exponential growth appears, after the summer the curve decrease s(not as much as in the beginning of the year) until November, where Black Friday appears and a rise can be seen again. In the end of the year the curve begins to decrease once more. The curve can be seen in figure (3.2).

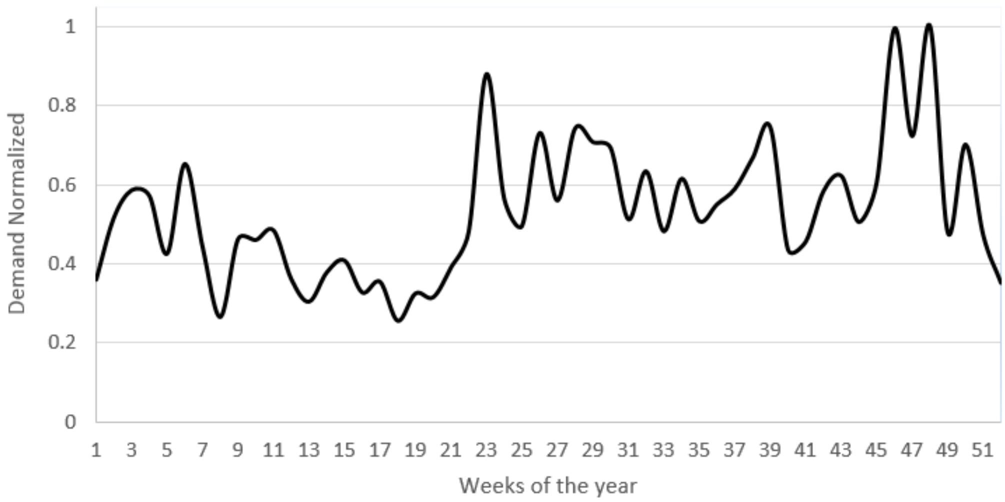


Figure 3.2: Demand variation of a typical store during the year

In those 4 periods the costumers are normally served in specific days of the week and only on those days, so a periodic supply could be applied. Since the main objective is to minimize the costs of the vehicles hired daily, and since the service days change with the period of the year, an optimization based on planning the routes daily is going to be applied.

Other two things that are important to know about the customers are that when a vehicle arrive to a customer, the customer must be ready for the delivery because some workers must be allocated to pickup the products. Due to that a time window is defined for each costumer and the vehicles can only arrive during that time window. The time window is defined by an Early Possible Time (EPT), if a vehicle arrives before that time a waiting time is allowed, counting this time as working time and not break time, and a Latest Possible Time (LPT), in this case if a vehicle arrives after this time the solution is not feasible. Unlike the service days, the time window of a store is fixed during the year and due to that, the time window can be considered a property of the customer itself. A table type with the customer information can be seen in table (3.1).

The last thing that is important to mention is, in order to increase the mean occupation rate of the vehicles, the customers of Worten can be merged with the customers of 4 other companies of Sonae Group, Sportzone, Modalfa, Zippy and Continente. This merge is only possible because the products that are shipped for the different types of customer are identical and have the same shipping rules, therefore it is possible to consider that the different customers are all from the same company. This strategy allow to increase the mean occupation rate because a more flexible system is created. A system with 99 customers has more flexibility to be optimized that 3 others with 33 customers each. This merge has another positive point, due to the different types of customers that belong to the same parent group, Sonae, some of the customers are in the same place, that allows to reduce the number of customers from 713 to 491, if two or more customers are in the same place it can be considered that only one customer that a demand equal to the sum of all demands. This allows to decrease the fixed time that each vehicle spends on the customers by decreasing the number of stops made, and therefore the amount of paperwork necessary. This also allows to minimize the total distance traveled because if the algorithm don't put the customers of the same cluster on the same route the total distance traveled will increase fairly. In conclusion this method allows to increase the mean occupation rate and therefore the price paid to the shipping company decreases.

3.1.3 Pallets on the depot

Leaving some pallets on the depot is one strategy that occurs in many real situations on the industrial and logistic sector, which allows large savings. This strategy is already being applied in the design of the current routes so it is really important to apply it in the algorithms.

Pallets on depot is when some pallets of the demand are left in the depot to avoid to subcontract a new vehicle in order to bring those few remaining pallets to the customer. This strategy can only be used if the type of sector and type of industry allows it, in this case is possible to leave some pallets on the depot and later deliver them. This strategy allows large savings because if some pallets are left on the depot it is possible to avoid hiring extra vehicles. The pallets that are left on the depot are going to be stored again, and are going to be shipped in the next possible delivery, along with the demand of that delivery.

One example is, if a customer has delivery days at Mondays and Wednesdays, and in Monday there is a demand of 34 pallets. Since the maximum number of pallets in a vehicle is 33, it make no sense to subcontract a new vehicle in order to deliver only one pallet. The best solution is to leave in the depot one of the 34 pallets and add that pallet to the demand of the next delivery (Wednesday).

The way to model this strategy is to allow the vehicles to delivery a capacity greater than their maximum capacity (in this case is going to be considered an extra capacity of 2 pallets per vehicle), so if a vehicle have a maximum capacity of 33 pallets, when the design of the route is being made, a route that has a maximum demand of 35 pallets is feasible, where 33 pallets can be supplied and two must stay at the depot.

With this strategy 2 problems arise, the first being the maximum number of pallets that is wise to leave in the depot. This number depends directly on price of leaving the pallets on depot, and the way to calculate this price is not straightforward. This price depends on the price of storing and on the "price" of customers dissatisfaction. Using some common sense is possible to find a range for this price. Is not wise to have a route that has a demand less than 10 pallets, also it is not wise in leaving 20 pallets or more on the depot. A range that can be apply is $[PriceCar/20, PriceCar/10]$, if the vehicle with capacity of 33 pallets in the more and less expensive zone is used it is obtain $[8.75, 64]$. This price must also be added to the price that Worten is going to spend on the shipping. The price chosen to run the algorithm was 20€, due to be business requirements.

The other problem that comes up with this strategy is which are the pallets that are going to be left in the depot and from which customers, recall that a route can have more than 1 customer. To solve this problem Hamilton's Method can be used, that divide the number of pallets on depot by the different customers having into account the demand of each one, however Hamilton's Method only gives the number of pallets that each customer should leave in the depot and not which ones. This process must be done by the internal network of Worten that knows what are the palettes that need to be shipped and what are the ones that can wait until the next delivery. For that reason the only thing the algorithm must do is to find the number of pallets that must be left in the depot in each route, and later it is decided which ones.

3.1.4 Reverse logistics

The reverse logistics is other aspect that is necessary to implement when designing the model. The reverse logistics is the transportation of the products from the customer to the depot and not the other way around. This is some products need to be supplied to the customers, also called direct logistic, and some products need to be pickup from the customer, called reverse logistic.

The reverse logistics is a necessary aspect in this problem because due to the type of sector some pallets must come back to the depot. Those pallets can be of old material that was not sold and is outdated, products that had some defects and need to be sent to the factory, products that were supplied in the wrong address or simply devolutions.

The reverse logistics can be modeled in two different ways:

- Some vehicles are dedicated to do only the reverse logistics, to pickup pallets from the customer to the depot, "broom cars";
- When a vehicle arrives at a customer both the direct and the reverse logistic are made, this is, the same vehicle can do both the delivery of the demand and pickup the products from the customer to the depot.

In the first type of reverse logistic, "broom cars", it is considered that the direct and the reverse logistics are fully independent. This allows a substantial reduction of computational time because it is only necessary to solve an identical problem twice and that can be solved in parallel. Note that the dimensions of the reverse logistic are much smaller than the dimensions of the direct logistic, due to that the number of vehicles necessary is also much smaller.

In the second type of reverse logistic it cannot be considered the direct logistics independent from the reverse logistics. In this case the time to delivery the demand of a customer depend on the number of pallets of reverse logistic that are already in a vehicle.

Next is considered an example of a vehicle that needs to supply 3 different customers:

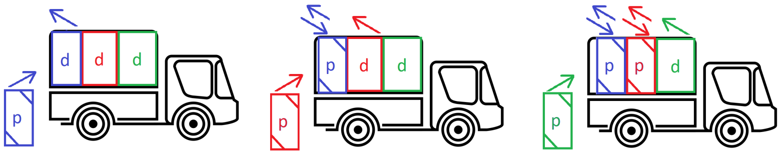


Figure 3.3: Example of simultaneous pickups and delivery

In the first customer the time spend on doing the delivery is only the time of unloading the demand and loading the pickup. In the second customer the time spend is equal to the time of unloading the pickup of customer 1 plus the time of unloading the demand of customer 2 plus the time of loading the pickup form customer 1 and 2. In the third customer the total time is equal to the time of unload the pickup from customer 1 and 2 and the demand of customer 3 plus the time of loading the pickups form 1, 2 and 3 into the vehicle again. Due to this dependence between delivery and collection the computation time necessary to calculate the solutions is going to increase substantial because the solution depends on both the pickup and the deliveries, and the number of possibilities increase fairly.

It is going to be implemented the "broom cars" in the algorithm because it is the way that it is implemented by the company in reality. Other thing is, since the demand of reverse logistic is much smaller than the demand of direct logistic, the saving in using the second method would not be relevant due to the large increase in computational time, since the final algorithm needs to be run under 2/3 hours.

3.1.5 Vehicles

The operation of the vehicles is also an important aspect to design a correct model. The vehicles that are used to supply the demands of the customers of Worten, are the ones that are subcontracted to a shipping company, for that reason is necessary to analyse the fleet of that company. The fleet of the shiping company is composed by vehicles with 33, 24, 16 and 12 of maximum capacity. Worten only subcontracts the vehicles that have a maximum capacity of 33 pallets for the big and normal flow of the demands of the customers. If a special delivery is necessary , another company is subcontracted to do it. The efficiency of each vehicle is calculated by dividing the number of pallets that a vehicle delivers by its maximum capacity, just like in equation (3.1).

$$\eta = \frac{Number\,pallets\,delivered}{Maximum\,capacity} \tag{3.1}$$

The vehicles can now be subdivided again in two, those who have a platform and those who do not have one. As it was said before this platform is only necessary if the customer that will be visited does not have one. Since most of the customers have the required platform the number of vehicles that need to have the platform will be much smaller than the other ones. Since the vehicles are going to be subcontracted it can be assumed that the capacity of the fleet is infinite.

It is assumed that vehicles travel at mean velocity of 80 km/h, they travel faster in the freeways but a slower within cities. To calculate the distance, and consequently the time between two customers, is used an approximation that assumes that the distance between two customer is straight. Later a correction factor is apply since vehicles need to travel on roads and the roads are not always straight. The straight distance is going to be calculated using the geographic coordinate of each customer, and with:

$$a = sin^2(\Delta\phi/2) + cos(\phi 1).cos(\phi 2).sin(\Delta\lambda/2) \tag{3.2}$$

$$c = 2.atan2(\sqrt{a}, \sqrt{1-a}) \tag{3.3}$$

$$d = R.c \tag{3.4}$$

Where ϕ is the latitude, λ is the longitude, R is earth's radius (mean radius = 6,371 km). Note that the angles need to be in radians to pass to trig functions [77]. With the straight distance calculated is applied a factor of 1.23, this factor was found comparing the straight distance with the real distance of the routes.

It is also assumed that each vehicle spends a fixed time of 35min per customer, this time is spend on stopping and filling the paperwork, plus a variable time of 1mim/pallet. This is the time that it takes to unload the demand of each customer. The vehicles already have the demands of the customers by order of arrival when it leaves the depot, for that reason it is only necessary to take time to unload the respective demand of each customer. The process of loading pallets on the vehicles is done in the depot before the departure of the vehicle and arrival of the conductor, so not waiting times are considered here.

Finally a vehicle is able to do an open or a close route. This is, if a vehicle is able to return to the depot after all the deliverers are hand over without violating any of the restriction, a close route is made. In another way if at least one of the restrictions is violated due to the return of the vehicle to the depot, an open route must be made and an extra fee is paid, corresponding to 50% of the vehicle value. Note that all the restrictions are present in the next subsection (3.1.6).

3.1.6 Feasibility

In this subsection is going to be analysed when a route is feasible or not. This is a very important aspect because the algorithm implemented may have some randomness and may try to create routes that are not feasible, so a way to evaluate if the routes created are feasible is necessary.

Since the routes are going to be made in Portugal, the routes first of all need to fulfill the European legislation [78]:

- A daily driving period of no more than 9 hours;
- A total accumulated duty time of no more than 13 hours;
- A 45 minutes break after 4.5 hours of driving;
- Service times at customers are not considered as break time.

The daily driving period is the time spent in transporting the vehicle. This time does not have into account the waiting time. Since the maximum daily driving period cannot be bigger that 9h and some routes takes more than 4.5h to arrive to a customer, a close route (the first and last stop are the depot) is not always feasible, so an open route must be applied. This is, if the vehicle does not return to the deport after the last customer, an extra cost of 50% on the cost of the vehicle must be added to that route. An open route will always be applied when a close route is not possible.

The total accumulated duty time is the sum of the daily driving period plus the waiting time and the working times. The working time has a fixed time per stop of 35min, time necessary to complete the documentation, and a variable time of 1 min/pallet, depend of the number of pallets that are unload. If a customer have a demand of 10 pallets, the vehicle will be stop for 45min at that customer, 35min for the paperwork and 10min to unload the pallets.

For a route to be feasible is also necessary to comply with the restrictions of reality. The capacity delivery must not be bigger that the maximum capacity of the vehicle. A customer can only be visited by a vehicle that can fulfill the restriction of the customer, like it was referred in subsection (3.1.2). A customer can only be visited during the time window, if a vehicle arrives before the time window a waiting time at the customer is allowed, otherwise the route is considered not feasible. The maximum number of pallets that can be left in the depot by a vehicle is two.

In resume, a route can only be made if is feasible and for a route to be feasible all the restriction mention above must be fulfilled. If the algorithm arrives to a solution that does not fulfill at least one of the restriction above, it must be considered not feasible and be discarded.

3.2 Site-Dependent Vehicle Routing Problem with Hard Time Windows

In this section a comparison between the different types of the Vehicle Routing Problem (VRP) that were introduced in chapter (2) and the logistics that is made by Worten, that were introduce in the beginning of chapter (3) is going to be made.

In subsections (3.1.1) and (3.1.5) it was possible to see that each customer is only allowed to be supply by some types of vehicles this is, some customers must be supply by a specific type of vehicle. In subsection (2.2.9) it was explained that when the customers have restrictions on the type of vehicle that is allowed to visit the problem is called Site-Dependent VRP. A Periodic VRP (PVRP) (2.2.3) could be applied since the year can be divided in 4 periods like it was explained in (3.1.2), but since the demands of the customers are only obtained in the day before the delivery, an algorithm that solve the problem daily must be used. VRP with Backhauls (VRPB) (2.2.6) could also be applied once it exists both delivers and pickups but like it was explain in subsection (3.1.4) the deliveries and the pickups will be modelled independently. In subsection (3.1.2) was introduced that each customer has a fixed time window to be supplied so a VRP with Time Windows (VRPTW) (2.2.8) must be applied. Finally it was explain in subsections (3.1.1) and (3.1.3) that the vehicles have a maximum capacity, and the way to modelled the pallets on the depot is also going to be model as a maximum capacity, so an Capacitated VRP (CVRP) (2.2.1) must also be applied, where the pallets on the depot is a soft constraint.

This problem can be called Site-Dependent Vehicle Routing Problem with Hard Time Windows (SD-VRPHTW) and Independent Pickups. This is not a very studied problem with only 2 paper found in the literature [72, 79], and only one with hard time windows [72].

3.3 Mathematical Formulation

In this section will be presented a mathematical formulation of the model that was presented in the previous section.

The formulation is going to be adapted from Cordeau et al. [45] where a VRPTW is presented, from Baldacci et al. [48] where a HVRP is presented and from Toth and Vigo [7] where a classic CVRP formulation is presented.

To present the formulation a graph is used, $\mathcal{G}(\mathcal{N},\mathcal{A})$, where $\mathcal{A}$ is the arc set that is indicated by mean of the end points $i,j \in N$ and $\mathcal{N} = \{0...n\}$ is the vertex set where i=0 is the depot node and $i = \{1...n\}$ is the set of customers that must be served. To serve the customer a fleet that is composed by *m* different vehicle where, $k \in V = \{1...m\}$ will be used. All Decision variables and parameters that are used on the mathematical model are stated below.

- X_{ij}^k - Binary variable, 1 if *j* is supplied after *i* by a vehicle *k*, 0 otherwise;
- d_i - Demand of customer *i*;
- a_i - Pallets that were not supplied to customer *i*;

- s_i - Pallets that were supplied to customer *i*;
- Q_k - Maximum capacity of vehicle *k*;
- Pk - Maximum number of pallets that may not be supplied in one route;
- Pt - Maximum number of pallets that may not be supplied in total;
- Y_{ij}^k- Quantity supplied to customer *i* when *j* is supplied after *i* by a vehicle *k*;
- C_k - Cost of a vehicle *k*;
- C_{pal} - Cost of leaving one pallet in the depot;
- E_i - Earliest time that a customer *i* can be supplied;
- L_i - Latest time that a customer *i* can be supplied;
- B_i^k- Exact time of service at each point *i* by vehicle *k*, 0 if vehicle *k* did not supply customer *i*;
- t_{ij}^k - Time that a vehicle *k* takes to go from customer *i* to customer *j*;
- T_f - Fixed time that a vehicle takes when supplying a customer;
- T_v- Variable time that a vehicle takes when supplying one customer;
- D_i^k - Binary variable, 1 if vehicle *k* can supply customer *i*, 0 otherwise;
- O_k - Cost of a vehicle *k* do an open route;
- C_{stop} - Cost of a vehicle making a stop in a customer.

The first thing that is necessary to define is the objective function, present in (3.5) it has the objective of minimizing both the subcontracted cars, the number of pallets that are left in the depot, the number of open routes and finally the number of stops, all the parts are represented in euros. The smaller the combination of those factors, the smaller the objective function. A cost in euros is added for every type of car that is subcontracted, for every stop made, if a route is open instead of close and finally for every pallet that should be supplied and was left in the depot.

$$Min(\sum_{k=1}^{m}\sum_{j=1}^{n} X_{0j}^k C_k + \sum_{i=1}^{n} a_i C_{pal} + \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{m}(X_{0j}^k - X_{i0}^k)O_k + \sum_{j=1}^{n}\sum_{i=0}^{n}\sum_{k=1}^{m} X_{ij}^k C_{stop}) \tag{3.5}$$

Equation (3.6) guarantees that a costumer can be visited once and only once from a vehicle and that vehicle can only come from one other node.

$$\sum_{i=0}^{n}\sum_{k=1}^{m} X_{ij}^k = 1 \quad j = 1...n \tag{3.6}$$

In equation (3.7) it is guaranteed that the vehicles are able to do only one route per day.

$$\sum_{j=0}^{n} X_{0j}^k = 1 \quad k = ...m \tag{3.7}$$

In the expression (3.8) it is ensured that the demand of all customers visited by a vehicle *k* cannot be larger that the vehicle capacity.

$$\sum_{j=1}^{n}\sum_{i=0}^{n} s_j X_{ij}^{k} \leq Q_k \quad k = 1...m \tag{3.8}$$

In equation (3.9) is guaranteed that what is delivered to a customer must be equal to the demand of the customer *i* less the number of pallets of the customer *i* that are left in the depot. It can be noticed that from equation (3.6) that only one vehicle can supply a customer *i*, so the total quantity supplied is only supplied by one vehicle.

$$\sum_{k=1}^{m}\sum_{i=0}^{n} Y_{ij}^{k} = d_j - a_j = sj \quad j = 1...n \quad Y_{ij}^{k} \leq Q_k \tag{3.9}$$

In equation (3.10) it is imposed that the number of pallets that are left in the depot by a vehicle *k* must be smaller that a predefined constant (*Pk*).

$$\sum_{j=1}^{n}\sum_{i=0}^{n} a_j X_{ij}^{k} \leq Pk \quad k = 1...m \tag{3.10}$$

Equation (3.11) is very similar to the previously one with the difference that what must be smaller that a predefined constant is the sum of all palettes that are left in the depot.

$$\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=0}^{n} a_j X_{ij}^{k} \leq Pt \tag{3.11}$$

Equation (3.12) guarantees that a route is feasible only if the arrival time at costumer *j* is such that allows travel from *i* to *j*.

$$X_{ij}^{k}(B_i^k + t_{ij}^k - B_j^k) \leq 0 \tag{3.12}$$

In expression (3.13) it is assured that the arrival time at the customer *i* is such that it is not smaller that the earliest possible time nor bigger that the latest possible time.

$$E_i \leq \sum_{k=0}^{m} B_i^k \leq L_i \quad i = 0...n \tag{3.13}$$

Regarding the legislation it is necessary to guarantee that the total driving time is not bigger that 9h (540min), that restriction is made in equation (3.14).

$$\sum_{i=0}^{n}\sum_{j=0}^{n} X_{ij}^{k} t_{ij}^{k} \leq 540 \quad k = 1...m \tag{3.14}$$

Once more regards the legislation it is necessary to assure that the total working time (hours driving plus hours of unloading the demand of the customers) is smaller that 13h (780min), that is ensured in equation (3.15).

$$\sum_{i=0}^{n}\sum_{j=0}^{n} X_{ij}^{k} t_{ij}^{k} + \sum_{i=0}^{n}\sum_{j=1}^{n} X_{ij}^{k} Tf + \sum_{i=0}^{n}\sum_{j=1}^{n} X_{ij}^{k} Tvs_j \leq 780 \quad k = 1...m \tag{3.15}$$

In equation (3.16) is ensured that the vehicle k that is assigned to the customer *i* is one of the possible vehicles that can supply customer *i*. Once more is possible to notice that from equation (3.6) only one vehicle is allowed to visit the customer *i*, this means that X_{ij}^{k} is 1 only once.

$$\sum_{i=0}^{n} X_{ij}^{k} - \sum_{i=0}^{n} X_{ij}^{k} D_j^k = 0 \quad k = 1...m \quad j = 1...n \tag{3.16}$$

Finally X_{ij}^{k} must be integer, that is assured by equation (3.17).

$$X_{ij}^{k} \in \{0, 1\} \tag{3.17}$$

3.4 Problem and solution example

In this section is going to be used 5 different customers with distinct characteristics to gain some sensibility on the results given by the model. For that is going to be used three different test sets to see how the model reacts.

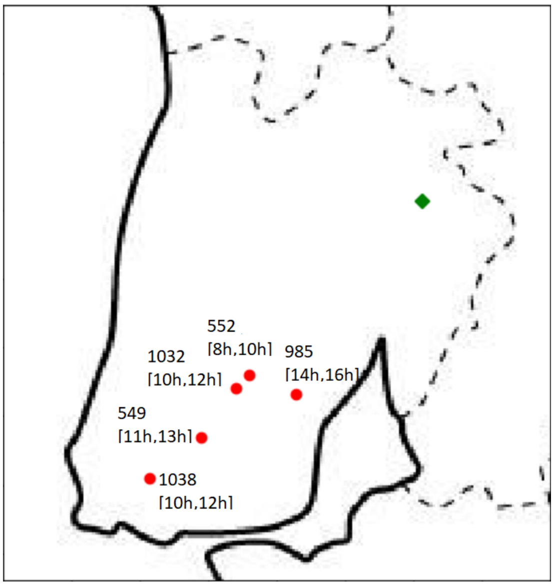


Figure 3.4: Place and time windows of customers used in problem example

In the first test set the demands will be such that the total demand is less than the maximum capacity of one vehicle, just like is possible to see in table (3.1):

Table 3.1: Characteristics of the customers in the first test set

Code Store	Name Store	Zone Code	Zone Name	Latitude	Longitude	Possible Types Car	EPT (Min)	LPT (Min)	Demand
985	Store 1	1	LISBOA	38.8107	-9.08964	[1,2]	840	960	10
552	Store 2	1	LISBOA	38.83517	-9.15585	[2]	480	600	2
1032	Store 3	1	LISBOA	38.81843	-9.17489	[1,2]	600	720	5
1038	Store 4	1	LISBOA	38.70625	-9.29854	[1,2]	600	720	7
549	Store 5	1	LISBOA	38.75776	-9.22455	[1,2]	660	780	12

In this first test set it is expected to have only one vehicle, once the sum of the demand is smaller that the maximum capacity of the vehicle (33 pallets). For the solution to be feasible the customer 552 must be supplied by a type 2 vehicle, so it is expected to have only one vehicle of the type 2 to supply all the customers. Since the algorithm also tries to minimize the path (minimum distance traveled) it is expected for the route to have no crossed lines.

Running the approach explain in chapter (4), with a number of generations big enough for the solution to converge, the final solution is:

$$[[1], [2, [552.0, 1032.0, 549.0, 1038.0, 985.0]]]$$

Figure 3.5: Chromosome solution of the first test set

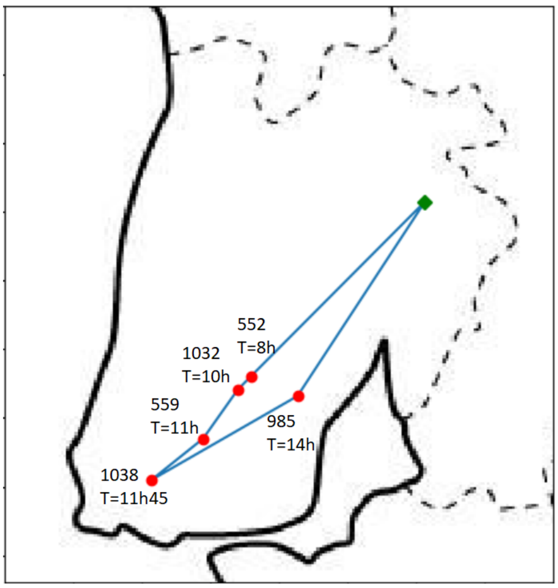


Figure 3.6: Route solution of the first test set

Just like it was expected, the chromosome of the final solution (3.5) is only one route, the representation of the chromosome (3.5) is latter explain in subsection (4.2.1), that is able to bring all the demands of the customers made by a type of car 2, with the distance travel minimized with no cross path. The vehicle begins to supply the customer 552 arriving at 8h. The vehicle departs for the depot at such a time that arrives at the first customer at the desired time, since the vehicle supply 2 units it spend 37 min there (32m+1*1m). Next the vehicle supplies the customer 1052, since it arrive early than the EPT it must wait in the customer until it can supply the customer. A similar thing happens to the 3º customer to be supplied, 549. The fourth customer can only be supplied at 11h45 so a waiting time is not necessary since the arrival is later than the EPT. Once more customer 985 is supplied in a similar way of the second customer. Finally the vehicle return to the depot.

For the second test set, the demand will be such that is necessary to use at least 2 vehicles to supply all the customers, with that we have:

Table 3.2: Characteristics of the customers in the second test set

Code Store	Name Store	Zone Code	Zone Name	Latitude	Longitude	Possible Type Car	EPT (Min)	LPT (Min)	Demand
985	Store 1	1	LISBOA	38.8107	-9.08964	[1,2]	840	960	10
552	Store 2	1	LISBOA	38.83517	-9.15585	[2]	480	600	10
1032	Store 3	1	LISBOA	38.81843	-9.17489	[1,2]	600	720	10
1038	Store 4	1	LISBOA	38.70625	-9.29854	[1,2]	600	720	15
549	Store 5	1	LISBOA	38.75776	-9.22455	[1,2]	660	780	12

If once more the algorithm explained in chapter (4) is used to solve the problem we get:

$$[[1, [549.0, 1038.0]], [2, [552.0, 1032.0, 985.0]]]$$

Figure 3.7: Chromosome solution of the second test set

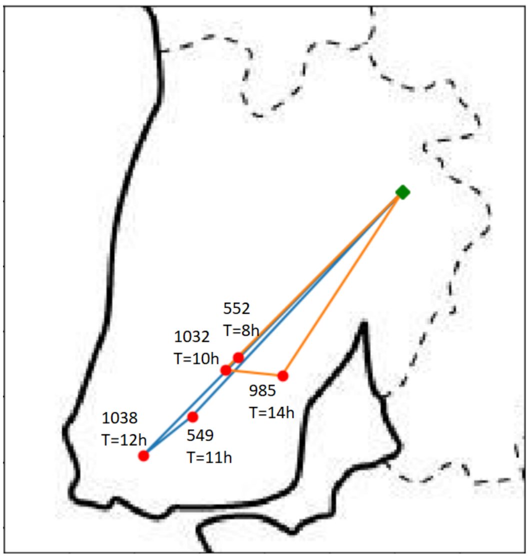


Figure 3.8: Route solution of the second test set

Once more if the chromosome of the final solution (3.7) is analysed, it is possible to notice that the best solution tends to have at lest two vehicles because the total demand of the customers is bigger that the maximum capacity of one vehicle (33 pallets), so the final solution must have at least two vehicles for the solution to be feasible.

In the third and last test set, table (3.3), it will be imposed a time windows to the customer such that the route will have to cross for the solution to be feasible with only one vehicle.

Table 3.3: Characteristics of the customers in the third test set

Code Store	Name Store	Zone Code	Zone Name	Latitude	Longitude	Possible Type Car	EPT (Min)	LPT (Min)	Demand
985	Store 1	1	LISBOA	38.8107	-9.08964	[1,2]	900	1020	1
552	Store 2	1	LISBOA	38.83517	-9.15585	[2]	540	660	1
1032	Store 3	1	LISBOA	38.81843	-9.17489	[1,2]	1020	1140	1
1038	Store 4	1	LISBOA	38.70625	-9.29854	[1,2]	720	840	1
549	Store 5	1	LISBOA	38.75776	-9.22455	[1,2]	420	540	1

Once more using the algorithm explained in chapter (4) to solve the problem, we get:

$$[[1], [2, [549.0, 552.0, 1038.0, 985.0, 1032.0]]]$$

Figure 3.9: Chromosome solution of the third test set

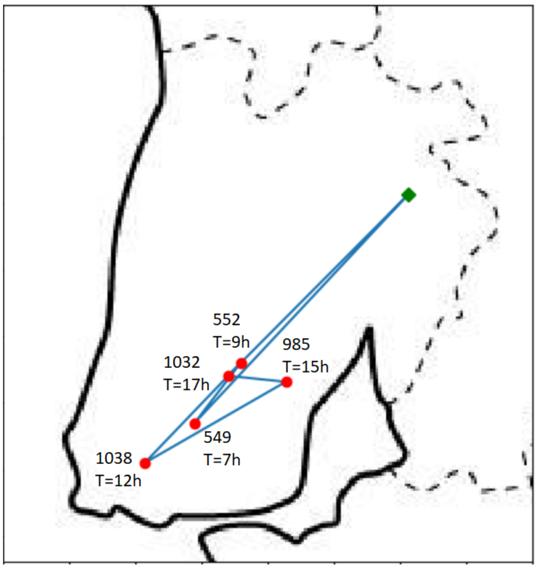


Figure 3.10: Route solution of the third test set

Analysing the solution of the last test set figure (3.10) it can be notice that the route does not follow the shortest distance so that one vehicle is used, because is cheaper to do a longer route than have to sub-contact another vehicle.

Chapter 4

Proposed Algorithms

As it was seen in the section (2.3), the literature shows different ways to solve different variants of the Vehicle Routing Problem (VRP). A Tabu Search (TS) could be used to solve the problem since is one of the methods that are most used and generaly gives better results. This is a powerful algorithm but requires substantial computing time (Baker and Ayechew [65]) and since a solution is desierd after 2/3 hours is not very convenient to use it, even if the implemented algorithm is just a prototype.

To solve the problem a Genetic Algorithm (GA) was initially proposed. Using a modify Saving heuristic to initialize the algorithm and, a Local Search (LS) to improve the solution obtained. Note that this algorithm is changed later due to poor results, these results are observed in section (5.4) and the new proposed algorithm is explain later in section (4.4). In figure (4.1) is possible to see the flowchart of the first propose algorithm.

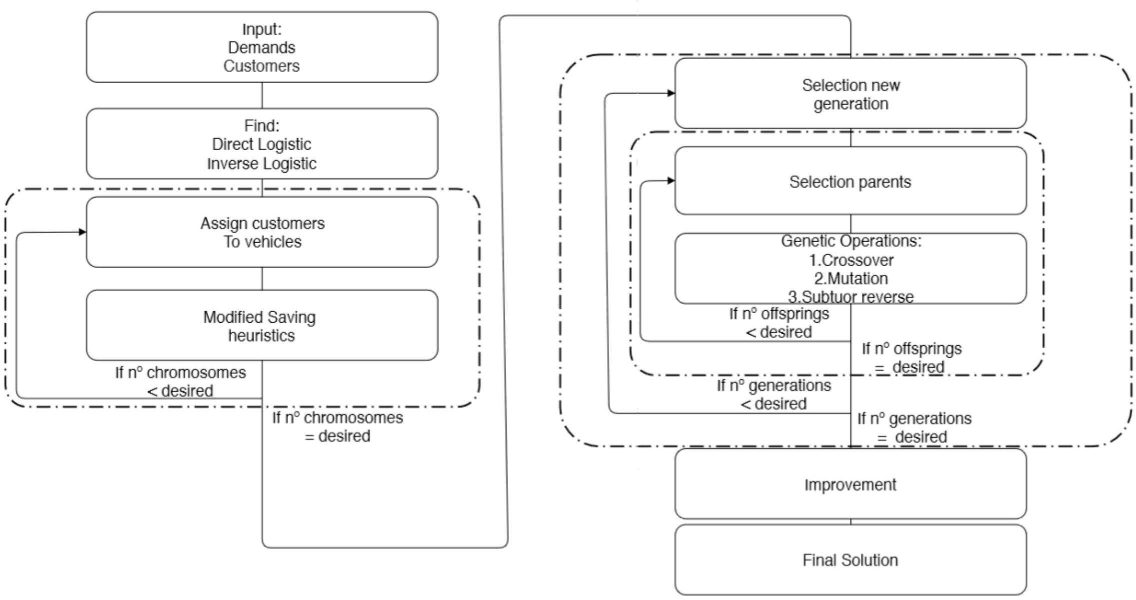


Figure 4.1: Flowchart of the initial proposed algorithm

The GA was used since it gives competitive results in terms of time and solution quality (Baker and

Ayechew [65]) in VRP's, and because it is a relative simple but effective method (Prins [68]). Once more is stressed that the final algorithm needs to be fast and give good results due to the computational time constrain. A negative point of the GA is that is necessary to find several parameters but after the calibration of the parameters the algorithm is able to provide very effective results (Prins [68]). Finally the GA is well suited for multiple objective optimization (Anbuudayasankar et al.[66]).

To do the initialization a variation of the (Clarke and Wright [1]) is going to be used. Ho et al. [67] shows that initializing the population using an heuristic leads to better results than initializing the solution randomly. Another advantage of this initialization is that since the solution is highly non feasible a random initialization would lead to non-evolution of the algorithm.

To improve the solution a Local Search (LS) is used, based on the crossover of Chand et al. [69]. This crossover requires long computation times therefore it cannot be always used in the GA.

In the rest of the Chapter (4) is going to be explain in detail the three different stages of the initial algorithm, starting with the initialization heuristic followed by the GA and ending with the improvement. Finally in the end of the chapter is going to be introduced some changes to the initial proposed algorithm.

4.1 Initialization

Since it is going to be used a Genetic Algorithm, is necessary to initialize the population. Normally the population is initialized randomly (Whitley [80]) but can also be initialized with a mixed population, where both random and structured individuals are part of the initial solution. In the VRP literature most cases use the second method (Baker and Ayechew [65]). In Ho et al. [67] a comparison is done between a random initialization and an initialization using a build heuristic and it's concluded that the second hypothesis achieves a better performance than the random initialization. It is also important to refer that when the population is initialized randomly, a repair procedure is normally necessary in order to obtain a feasible solution. An algorithm based on Clarke and Wright [1] was implemented since it is one of the best construction algorithms (Bräysy and Gendreau [46]). The pseudo algorithm can be seen below, algorithm (4.2), and an extensive pseudo algorithm can be seen in the annex (A.1).

Algorithm 1 Initial construct algorithm

INPUT: Customers with demands
OUTPUT: Routes

```
for each customer do
    Assign Customer to a type of car
end for

for each vehicle do
    Initialize customers routes
    Calculate Saving matrix
    while any value of Saving matrix >0 do
        Route ← Modify Saving Heuristics
        Update Saving matrix
    end while
end for
```

Figure 4.2: Initial construct algorithm

The algorithm starts by receiving the information of the customers that need to be served and their respective demands. With this information, each customer is assigned to one type of vehicle, this assignment is made by comparing the price of the vehicles that can supply the customer. The smaller the price the higher the probability of a customer be served by that type of vehicle. Note that this processes is not deterministic so each time the algorithm runs a different solution may appear.

After all the customers are assigned to one type of vehicle, an adaptation of the Clarke and Wright [1] is used to each type of car independently. First an initial solution is made, where all customers are served independently, one vehicle per customer, and all customers are considered "not assigned". After the "Saving" matrix of joining one customer to the route of a second customer is calculated. This "Saving" calculations differs on the original "Saving" used by Clarke and Wright [1], the original way can be seen in section (2.3.2). The "Saving" used in this algorithm is calculated taking into account not only the distance between the customers but also how far are the customer's time windows too. The further they are in space the less likely those customer are to be in the same route and the further they are in time less likely those customer be in the same route once more. Two customers can be very close in space but very far in time (Solomon [43]). To calculate the "Saving" is used:

$$S_{ij} = \frac{1}{C_1 + C_2}, \quad C_1 = d_{ij}, \quad C_2 = l_j - l_i \quad j = 1...n, \quad i = 1...n \tag{4.1}$$

Where d_{ij} is the distance to go from customer *i* to customer *j* and l_i is the latest possible time that the costumer *i* can be served. Note that the "Saving" (S_{ij}) of insert a customer *i* in the route of the customer *j* is different from the "Saving" of insert a customer *j* in the route of a customer *i*, also a bigger weighting is given to the customers that have the time window early in time to favor those connection first.

Next the pair of customers that have the biggest "Saving" are chosen and one of tree things can occur:

1. Both customers are "not assign", in this case customer *j* is going to be inserted on the route of customer *i*, customer *i* will not be able to be chosen again ($S_{ij} \rightarrow\ < 0, S_{ji} \rightarrow\ < 0, j = 1...n$), and both customers pass from "not assign" to "assign", if and only if the solution is feasible;

2. One of the customers is "assign" and the other is "not assign", in this case the "not assign" customer is inserted in the end of the route of the "assign", the "assign" customer will not be able to be chosen again, and the "not assign" pass to be an "assign" one, again if and only if the solution is feasible;

3. Both customers are "assign", in this case it is not possible to join the customers, so the customers can not be chosen simultaneously again.

This process is repeated until there are no more "Saving" $>$ 0. If the algorithm is run multiple times, different solutions will be generated. This allow us to initialize the Genetic Algorithm by choosing the best *N* solutions, where *N* is the size of the population used in the GA. This algorithm cloud be improved to achieve better solutions, but this is not only not necessary but also advised against. Having an initial solution too optimized could lead to less solution diversity and more difficulty of leaving the current solutions, this could lead to less flexibility and possible non-evolution of the GA.

4.2 Genetic Algorithm

The Genetic Algorithm (GA) is a stochastic optimization technique that is used to solve big combinatorial problems. The principle of the algorithm is explained in subsection (2.3.3). The GA was used due to getting fast and quality solutions and also due to being well suited for multiple objective optimization (Anbuudayasankar et al. [66]).

In the rest of this sections going to be explain the chromosome representation and the way to evolve the algorithm and the different operations used in this GA.

4.2.1 Gene representation

Since the GA will be used to solve the problem as mentioned earlier,first is necessary to find a way to represent all the relevant information in the chromosome, this representation is one of the critical issue when developing the GA (Anbuudayasankar et al. [66]). The chromosome representation was based to the chromosome of Chand et al. [69], the generic way to represent the chromosome is show bellow (4.3):

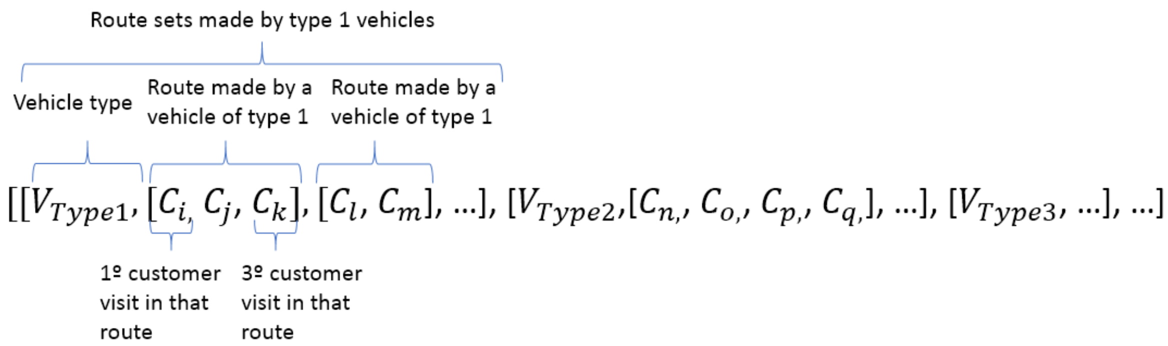


Figure 4.3: Generic representation of the chromosome

Were V_{type1} is the type 1 vehicle and V_{type2} is the type 2 vehicle, and $C_{i..q}$ are the different customer that are supplied. So in this case the customers $C_{i..m}$ are the customers that are supplied by vehicles of the type 1 were $C_{i..k}$ are supplied by one vehicle and $C_{l..m}$ are supplied by another vehicle of type 1. One simple example of a chromosome is represented below:

$$[[1, [1038.0, 535.0, 520.0, 1087.0]], [2, [1450.0, 982.0, 549.0, 1079.0, 506.0], [985.0, 552.0, 1032.0, 5198.0]]]$$

Figure 4.4: Example a chromosome representation

In the example of the chromosome is possible to see that the solution is composed by 3 routes in total, $[1038.0, 535.0, 520.0, 1087.0]$, $[1450.0, 982.0, 549.0, 1079.0, 506.0]$ and $[985.0, 552.0, 1032.0, 5198.0]$, where the first route belongs to the type of vehicle **1** and the other two routes belongs to the type of vehicle **2**. Those routes, that are part of the chromosome, can also be considered genes of the respective chromosome. In this chromosome all information needed to calculate the fitness, and the feasibility is represented. Note that the chromosome is still encoded, so is necessary the table of the different customers with their demands to decode the chromosome. An example table is represented below, table (4.1):

Table 4.1: Example of customers with demands

Store Code	Store Name	Zone Code	Zone Name	Latitude	Longitude	Possible Type Car	EPT (Min)	LPT (Min)	Demand
985	Store 1	2	LISBOA	38.8107	-9.08964	[1,2]	600	720	10
552	Store 2	2	LISBOA	38.83517	-9.15585	[1,2]	540	660	2
1032	Store 3	2	LISBOA	38.81843	-9.17489	[1,2]	600	720	5
1079	Store 3	2	LISBOA	38.77804	-9.22063	[1,2]	570	690	2
1038	Store 4	3	LISBOA	38.70625	-9.29854	[1,2]	450	570	7
535	Store 5	3	LISBOA	38.69772	-9.37164	[1,2]	540	660	5
549	Store 6	3	LISBOA	38.75776	-9.22455	[1,2]	540	660	12
1087	Store 7	3	LISBOA	38.8648	-9.32673	[1,2]	540	660	8
506	Store 8	3	LISBOA	38.79705	-9.33008	[1,2]	720	840	11
1450	Store 9	3	LISBOA	38.77488	-9.33906	[1,2]	450	570	6
982	Store 10	3	LISBOA	38.78901	-9.34013	[1,2]	540	660	3
520	Store 11	3	LISBOA	38.76427	-9.3609	[1,2]	540	660	10
5198	Store 12	1	LISBOA	38.72435	-9.15988	[1,2]	840	960	6
1079	Store13	2	LISBOA	38.778036	-9.220627	[1,2]	570	690	1

With the chromosome and with a table similar to table (4.1) is possible to decode the chromosome. In the example of the chromosome (4.4), the first routes goes from the depot to Store 4, Store 5, Store 11, Store 7 and finally if possible go back again to the depot. The vehicle that does this route will bring a total of 30 pallets to satisfy the demand of all the customers. If the sum of the demands for one route is bigger that the maximum capacity of the vehicle, the excess pallets would have to be left in the depot. With the geographic coordinates is possible to calculate how much the vehicle will travel, this calculation is explained in subsection (3.1.5). With this information is already possible to see if the set of routes are possible, using the restrictions (3.6 until 3.17) and who much it costs using the cost function (3.5).

4.2.2 Reproductive processes

The reproductive processes is based on generating new solutions from the chromosomes of a given population, the new chromosomes are often called offspring and the chromosomes chosen from the population are often called parents. The offspring is going to inherit some of the characteristics of their parent (Baker and Ayechew [65]). From a given population some parents chromosomes are going to be chosen to generate offspring, this selection normally have into account the fitness of the chromosome of the population, the chromosomes with higher fitness are more likely to be chosen to generate new solutions. The process is based on the Darwin's theory (Nazif and Lee [9]), where the specimen with better characteristics are more likely to reproduced and survive. In this case, the characteristic to be preserved and reproduced is the value of the objective function (3.5).

The process of reproduction of new offsprings is divided in 3 sub-processes. The first process, the crossover, will theoretically converge the solution to a local minimum, this will intensify the existing solutions. The second process, the sub-tour reverse, will also intensify the existing solutions. The third process, mutation, have the objective of explore new solutions and diversify the set of already existing specimen, to search for the global minimum and avoid getting stuck in the local minimum. The way to choose the chromosomes used will be explained in the next subsection (4.2.3).

Crossover

The Crossover operation used is based on Ho et al. [67] that is based on the classical order crossover from Cheng and Gen [81]. The steps of the operation can be see bellow:

1. Select two chromosomes from the population;
2. Chose at random a gene, route, of one chromosome;
3. Delete the information contain in the gene on the other chromosome;
4. Insert the gene at random on the receiving chromosome;
5. Change the receiving and giving chromosome and repeat the process.

First of all is necessary to select 2 parents chromosomes in which one of them will receive information and another will give information. For sake of simplification the giving chromosome will be called chromosome **1** and the receiving chromosome will be called chromosome **2**. From chromosome **1** a gene, sub-string, will be chosen at random, similar to a two point crossover. In this case the gene is always one route of a vehicle. After, the information of the gene chosen from chromosome **1** will have to be deleted from chromosome **2** in order to ensure that there is no duplicate information. Finally the gene is inserted in chromosome **2** in a random place, since the gene will always be one route, the important thing is in which type of vehicle the gene is inserted. This process is repeated where chromosome **1** is the receiving chromosome and chromosome **2** is the giving chromosome, so one crossover operation always generate 2 offsprings. Note that the offsprings generated may not be always feasible, for example if the gene that will be inserted have a customer that can only be served by a type one vehicle, and this gene is inserted in a type two vehicle. If in any case the gene chosen from the giving chromosomes also exist in the receiving chromosome, another gene will be chosen to guarantee that a clone is not made. It is possible to see bellow in figure (4.5) an example of a crossover operation:

Step1

P1:[[1,[1038.0, 535.0, 520.0, 1087.0]], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],**[985.0, 552.0, 1032.0, 5198.0]**]]
P2:[[1,[520.0, 552.0, 1087.0],[1450.0, 1079.0, 549.0, 535.0, 982.0]], [2,[985.0, 1038.0, 506.0, 5198.0, 1032.0]]]

Step2

O1:[[1,[520.0,1087.0],[1450.0, 1079.0, 549.0, 535.0, 982.0]], [2,[1038.0, 506.0]]]

Step 3

O1:[[1,[520.0,1087.0],[1450.0, 1079.0, 549.0, 535.0, 982.0]], [2, **[985.0, 552.0, 1032.0, 5198.0]**,[1038.0, 506.0]]]

P=Parent chromosome O=Ofspring chromosme

Figure 4.5: Example of the crossover operation

The crossover operation will generate new solutions that will tend to a local minimum because by adding only one route to the chromosome an improvement of the initial solution is possible, but the improvement of the number of routes will not be very frequent.

Sub-tour reverse

The sub-tour reverse that is used in this GA is based on Nazif and Lee [9], the simplified steps can be seen bellow:

1. Select one chromosome from the population;
2. Chose at random a gene, route, from the chromosome;
3. Chose at random sub-tour of the gene;
4. Invert the sub-tour.

The sub-tour reverse is very similar to the crossover operator. First is necessary to select a chromosome from the population. From the chromosome selected, parent chromosome, a gene is chosen from the chromosome, just like in the crossover the gene is always a route. After, a sub-route of the route is chosen, the sub-route can vary from only two elements to the full route. Finally the sub-route chosen is inverted. The sub-tour reverse operation only need one parent chromosome to generate an offspring and will generate only one offspring. An example of the sub-tour reverse is illustrated bellow.

Step1

P:[[1,[1038.0, 535.0, 520.0, 1087.0]], [2**,[1450.0, 982.0, 549.0, 1079.0, 506.0]**,[985.0, 552.0, 1032.0, 5198.0]]]

Step2

P:[[1,[1038.0, 535.0, 520.0, 1087.0]], [2,[1450.0, **982.0, 549.0, 1079.0**, 506.0],[985.0, 552.0, 1032.0, 5198.0]]]

Step 3

O:[[1,[1038.0, 535.0, 520.0, 1087.0]], [2,[**,** 1450.0, **1079.0, 549.0, 982.0,** 506.0],[985.0, 552.0, 1032.0, 5198.0]]]

P=Parent chromosome O=Ofspring chromosme

Figure 4.6: Example of the sub-tour reverse operation

Due to the fact that the fitness, equation (3.5), is mostly influenced by the number of tours made, the fitness of the new chromosome generated will not diverge much from the fitness of the parent chromosome, this operation will only be able to improve slightly the chromosome selected, improving the path of that route. But this operation also create new solutions that later could be improved by applying other operations.

Mutation

The mutation operation is based in Chand et al. [69], with a small variant, this variation is explain in the end of this subsection. The simplify steps can be seen bellow:

1. Select one chromosome from the population;
2. Chose at random a gene, route, from the chromosome;
3. Chose at random a customer from the gene;
4. Delete the customer in the original chromosome;
5. Insert at random the customer in any part of the chromosome.

First is selected from the population one chromosome that will be used to do the mutation operation. Next a gene of the initial chromosome is chosen at random, and a customer from that gene is also chosen at random. After the customer needs to be deleted from the original chromosome to guarantee that there is no duplication of the information. Finally the customer selected will be insert at random in any part of the chromosome. An example is illustrated in figure (4.7).

Step1

P:[[1,[1038.0, 535.0, **520.0**, 1087.0]], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 552.0, 1032.0, 5198.0]]]

Step2

O:[[1,[1038.0, 535.0,1087.0]], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 552.0, **520.0**,1032.0, 5198.0]]]

P=Parent chromosome   O=Ofspring chromosme

Figure 4.7: Example of the mutation operation

In Chand et al. [69] two customers are also chosen at random and are swapped, but in this problem was decided to chose only one customer due to the feasibility of the problem, since there are many restriction, swapping two customers could lead to more infeasible routes generated. The mutation is the operation that can lead the algorithm to leave a local minimum and improve the current solution. This occurs once more due to the fitness, equation (3.5), being largely influenced by the number of routes. Again to do the mutation is only necessary one parent chromosome and only one offspring is generated.

4.2.3 Selection process

The selection operation is the process that allows to select the chromosomes from a given population, normally this selection is based on the fitness function value, equation (3.5). The Selection process can be divided in two, just like it is possible to be seen in (4.1), the selection of the next population and the selection of the parents chromosomes that are used to generate new offsprings. Both selections are based on Ho et al. [67], where a roulette wheel operation is used. The roulette wheel operation was first introduced in 1989 in Goldberg [82]. This method is a probabilistic algorithm where a ”roulette wheel” has a size proportional to the fitness for each chromosome of the population. The size of the wheel for each chromosome, this is the probability of a chromosome be chosen, is calculated using:

$$P_{sel_i} = \frac{1/f_i}{\sum_{i=1}^{Npop} 1/f_i} \tag{4.2}$$

Where f_i is the fitness value of the chromosome that is get from equation (3.5).

The first selection operation, selection new population, in made by having a set of solutions where a chromosome is chosen using the roulette wheel operation at a time until the number of chromosomes chosen is equal to the desired population size, always guaranteed that clones (identical solution) are not possible to be chosen. This set of solutions can appear in two ways, in the initialization section (4.1), where there is not yet a population set, and at the end of each generation where the last population and the offsprings generated in that generation are merged to generate the population set. In this selection an extra step is used before using the roulette wheel operation, this step is called elitism where some of the best solution of the population go directly to the new population without going through the roulette wheel operation. This guarantees that the best chromosomes always remain in the population ensuring that the best solution is not lost.

The second selection operation, selection parents, is made always after the genetic operation is chosen. The genetic operations are chosen with a fixed probability picked in the begging of the GA and are made until the number of feasible offsprings generated are equal to the desired number of new solution. After the genetic operation is chosen, the roulette wheel operation is used to chose which is the chromosome, or chromosomes, selected to be used in the genetic operation.

Using this selecting allows to intensify the current solution since most of the improvements are made in the chromosomes of the population with the best fitness but also allow to explore new solution spaces by trying to improve solutions that have a not so good fitness. So using this operation the best solution are more likely to reproduced and survive, improving of the current solution, but the other solution have also the possibility to reproduce and survive, exploring the solution space.

4.3 Local Search

The Local Search (LS) is an heuristic based on improving the current solution iteratively by exploring the neighboring space. The LS that is going to be used to explore the neighboring space is an adaptation of the crossover of Chand et al. [69]. The steps are enumerated bellow:

1. Initialize a chromosome;
2. Chose the smallest possible gene, route, of the chromosome;
3. Insert the gene chosen on a list, this gene cannot be selected again;
4. Delete the information contained in the gene in the chromosome;
5. Insert every customer of the gene is the best possible place in the chromosome;
6. If the new chromosome is better than the old one, update the chromosome;
7. Repeat set 2 to 6 until, stopping criteria is met.

The first thing that is necessary to do, to run the algorithm is to initialize one chromosome. This initialization can be done in the same way the GA is initialized, where the algorithm present in section (4.1) is run the same number of times as the desired number of chromosomes and the best chromosome is chosen to initialize the LS. The initialization can also be done by running the GA and choosing the best chromosome from the last generation just like in flow char (4.1), where the LS is the improvement.

After the smallest possible route, the one that serves the least customers, of the chromosome is chosen. This is done because the bigger the route, the more likely it is to be a good route with good efficiency. Next the route chosen is inserted in a list and the routes from that list cannot be chosen again, this is done to avoid infinite cycles. Then the customers that are contain in the gene chosen are going to be insert in the place where the fitness of the chromosome is minimized always guaranteeing that the chromosome is feasible, once more using equation (3.5). If a customer cannot be inserted on any of the existing routes, a new one is created with that customer. Note that the best place for each customer individually do not ensures automatically a minimization of the fitness of the chromosome, so if there is not an improvement on the fitness the update of the chromosome is not done and the next smaller route is chosen. Finally the algorithm stops when there is no improvement of chromosome for 5 consecutive iterations. An example of one iteration of the LS can be seen in figure (4.8).

Step1

C:[[1,**[1038.0, 535.0, 520.0, 1087.0]**], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 552.0, 1032.0, 5198.0]]]

Step2

C:[[1], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 552.0, 1032.0, 5198.0]]] R:**[1038.0, 535.0, 520.0, 1087.0]**

Step3

C:[[1], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, **1038.0**, 552.0, 1032.0, 5198.0]]] R:**[535.0, 520.0, 1087.0]**

Step4

C:[[1, **[535.0]**], [2,[1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 1038.0, 552.0, 1032.0, 5198.0]]] R:**[520.0, 1087.0]**

Step5

C:[[1, [535.0]], [2,[**520.0,** 1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 1038.0, 552.0, 1032.0, 5198.0]]] R:**[1087.0]**

Step6

C:[[1, [535.0, **1087.0**]], [2,[520.0, 1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 1038.0, 552.0, 1032.0, 5198.0]]]

Step7

C:[[1, **[535.0, 1087.0]**], [2,[520.0, 1450.0, 982.0, 549.0, 1079.0, 506.0],[985.0, 1038.0, 552.0, 1032.0, 5198.0]]]

C=Chromosome   R=Route

Figure 4.8: Example of one iteration of the Local Search

4.4 Hybrid Genetic Algorithm (HGA)

The Hybrid Genetic Algorithm (HGA) is an hybridization between the two algorithms that were presented in this chapter, the Genetic Algorithm (GA) that was introduced in section (4.2), and the Local Search (LS) that was introduced in section (4.3). This hybridization was made because when the parameterization of the proposed algorithm at the beginning of chapter (4), figure (4.1) was made, it was noticed that the LS was able to improve the solution in a way that the GA was not able because its operations do not directly take into account the objective function, unlike LS. It is possible to notice this difference of evolution later in figure (5.2). It was also possible to notice that the LS takes big computational time so if the reproductive process is replaced by the LS the algorithm will not be able to evolve. For the reasons given above a new HGA is proposed where the reproduction process varies between the GA reproduction process, present in subsection (4.2.2), and between an LS variation where only one iteration of the LS is done to generate a new offspring, like in figure (4.8), from step 1 to step 6. The flowchart of the new proposed algorithm is represented in figure (4.9).

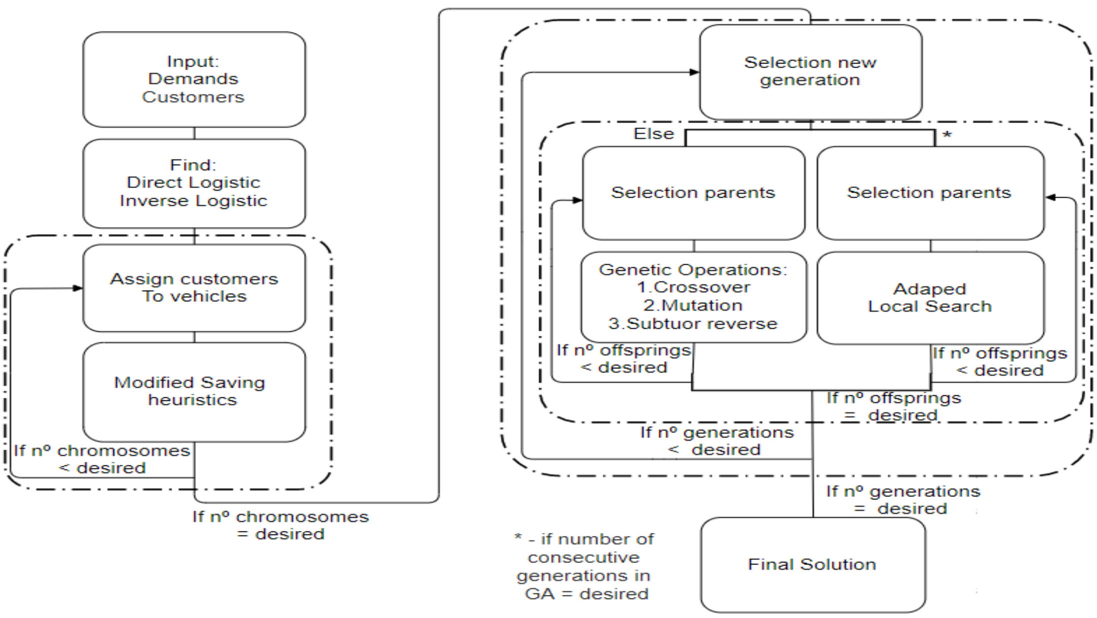


Figure 4.9: Flowchart of the final proposed algorithm

Since the only thing that changes in this hybridization with respect to the initially proposed algorithm (4.1) is the manner in which new offsprings are generated, it can concluded that all of the previously stated characteristics still could be considered, those characteristics are in the beginning of chapter (4) and in the beginning of section (4.2). Other thing that is also noticeable in the flowchart (4.9) in relation to the flowchart (4.1) is that the improvement is no longer used, this is due to the fact that by the end of the HGA the solution is already very close to the optimum, which causes very little flexibility so the LS is no longer able to improve the current best solution. Those results appear in section (5.4).

Chapter 5

Results

In this chapter the results of the theoretical models that were presented in chapter (4) will be tested and compared with the baseline solution that were given and used by the company. First will be introduced the problem that is going to be solved and the baseline solution. After will be introduced how to code the algorithms and how the different modules of the code are divided. Next the cumin towards the Hybrid Genetic Algorithm (HGA) and its sensibility analysis and parameterization will be done to find what are the best parameters to be used in this concrete problem. In the end of the chapter the final results given by the different methods will be compared with the baseline solution.

5.1 Problem Data

The baseline solutions are routes planed for weeks 6, 24 and 50. These solutions are only composed of direct logistics since no concrete data on reverse logistics has been provided. The demands of the aforementioned weeks are based on forecasts of previous years and weeks, but even so those demands were heavily analysed by the subcontracted company, and the routes of those weeks were build for those demand. In short, if the forecast demands were real, the routes that will be analyzed next would be the routes made by the subcontractor, those routes are the baseline solution.

The forecast was made and analyzed in three different weeks, due to the fact that the demands of the customers are influenced by seasonality, just like it was possible to see in subsection (3.1.2), so the year was divided in 3 periods and a typical week of each period was analyzed. The first period, weak 6, is the period of low volume of demands, the second period, weak 24, is the period of medium volume of demands, and finally the third period, weak 50, is the period where a high volume of demands is considered. The weeks have been analyzed Monday through Friday only, as no weekend deliveries will be made.

It is going to be possible to see in section (5.2) that some of the solutions given by the subcontractor are not feasible according to the formulation made in section (3.3), and with reality. This happens because the subcontracted first does a bulk analysis with good efficiencies and later some routes need to be adjusted. The main infeasibility incurred is the number of pallets that a vehicle can take, there are some routes that have more than 50 pallets which is clearly non feasible since a vehicle can take at maximum 33 pallets and leave at maximum 2 pallets at the depot. To circumvent this unfeasibility two different type of solutions will be made. The first approach is saying that all the pallets that can not be carried by the vehicle are going to be left at the depot and a price for each pallet is paid, this approximation will be called Base1. The second approach is saying that if a vehicle carries mores that 35 pallets, 33+2, another vehicle is going to be necessary to bring the remaining pallets, this approximation will be called Base2. Other thing that is going to me made in the baseline solutions is to assume that all routes are closed ones just like it was explained in subsection (3.1.5). This approximation is done because the data given by the company only have the customers that are supply by a particular vehicle and not the order that those customers are supplied. To make sure that the fitness is not influenced by non-existent wait times a close route is assume to all vehicles. Note that this approximation will always give a better solution than the real one.

5.2 Data Analysis

In this section the data provided will be analyzed, first will be analyzed what are the characteristics of the problems that needs to be solved and optimized during the three type weeks. Then, the routes that were provided by the company, the baseline solutions, will be analyzed in the end the section. The first day of the first typical week is going to be analyse in detail.

In the first day of week 6, Monday, there are 212 different customers that need to be supply that have a total demand of 757 pallets, so each customer has on average a demand of 3.57 pallets. If it is considered that each vehicle can bring a total of 33 pallets, there are necessary at least 23 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, where two pallets can be left in the depot, there are necessary at least 22 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers just like it is explained in subsection (3.1.2) we end with a total of 97 customers with an average demand of 7.8 pallets. The analysis of the other days is summarized in the table (5.1).

If we analyse table (5.1) is possible to notice that the total demand of the customers is bigger in week 50 and is smaller in week 6, just like expected. It is also possible to see that the mean demand of the customers also changes in the same way in the different weeks. Finally it can be seen that doing the cluster substantially reduces the number of customers which increases demand media.

Now If we analyse the baseline solution for Monday of week 6, it is possible to notice that 29 vehicles are used with an average efficiency of 0.79. Note, the way to calculate the efficiency is explain in the subsection (3.1.5). The baseline solution is the same using first or the second approximation because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The fitness of this solution using the objective function is 13775€, leaving 1 pallet on the depot. The analysis of the other days is summarized in the table (5.2).

Table 5.1: Characteristics of the days of the week

		Number of customers	Total demand	Mean demand	Necessary vehicles (33 pallets)	Necessary vehicles (35 pallets)	Number of Clusters	Mean demand cluster
Week 6	Monday	212	757	3.57	23	22	97	7.8
	Tuesday	124	472	3.8	15	14	63	7.49
	Wednesday	213	772	3.62	24	23	100	7.72
	Thursday	141	525	3.72	16	15	66	7.95
	Friday	196	692	3.53	21	20	92	7.52
Week 24	Monday	169	801	4.74	25	23	104	7.7
	Tuesday	128	596	4.65	19	18	73	8.16
	Wednesday	233	982	4.21	30	29	115	8.54
	Thursday	117	547	4.68	17	16	65	8.42
	Friday	132	667	4.05	21	20	81	8.23
Week 50	Monday	160	1056	6.6	32	31	93	11.35
	Tuesday	142	972	6.85	30	28	76	12.79
	Wednesday	195	1211	6.21	37	35	99	12.23
	Thursday	138	910	6.59	28	26	76	11.97
	Friday	148	1017	6.87	31	30	91	11.18

Table 5.2: Analysis of the baseline solutions

		Base 1 fitness	Base 1 efficiency	Base 1 pallets depot	Base 1 number cars	Base 2 fitness	Base 2 efficiency	Base 2 pallets depot	Base 2 number cars
Week 6	Monday	13775	0.792	1	29	13775	0.792	1	29
	Tuesday	7375	0.841	0	17	7375	0.841	0	17
	Wednesday	14225	0.800	4	29	14225	0.800	4	29
	Thursday	8155	0.836	1	19	8155	0.836	1	19
	Friday	12835	0.800	5	26	12835	0.800	5	26
Week 24	Monday	15525	0.782	88	28	16820	0.662	3	37
	Tuesday	10145	0.806	64	20	11320	0.644	1	28
	Wednesday	21804	0.803	171	31	20850	0.638	2	47
	Thursday	9420	0.781	57	19	11050	0.611	2	27
	Friday	13950	0.686	68	27	15310	0.602	1	34
Week 50	Monday	16530	0.887	52	35	17955	0.789	9	41
	Tuesday	12995	0.911	40	31	13820	0.807	13	36
	Wednesday	18600	0.876	69	40	20625	0.737	9	50
	Thursday	12905	0.912	59	29	13540	0.753	13	37
	Friday	16745	0.810	11	38	17405	0.777	2	40

In the table (5.2) is possible to note that the fitness of the solutions is almost always smaller when the first approximation is used, this is when all the extra pallets are left in the depot. Using this approximation some times leads to a solution where around 10% of the demands are left in the depot, which clearly cannot be done. Finally is possible to notice that when the second approximation is used the number of cars used increases substantially and consequently the efficiency decrease but the final solution looks much more similar to the reality.

5.3 Brief programming explanation

The algorithms presented in chapter (4) was programmed in Python and run on a computer with a Intel Core i7-4790 processor with 3.6GHz and a RAM of 8GB. The code used is divided in 4 big parts, the preprocessing, the initialization the Hybrid Genetic Algorithm (HGA) and finally the Improvement.

The preprocessing has the main objective of receiving the demands from a file, in this case in a excel, and transform it in a way that the program is able to use all the relevant information given. An input excel table with the direct logistics and the inverse logistics (if necessary), and a excel table with the characteristics of all possible customers are read, then a data crossing is done ending with a similar table as (4.1) for the linehauls (direct logistics) and other from the backhauls if necessary (inverse logistics). Note, the crossing only adds the demand to the customers table and deletes the customers that have no demands.

In the initialization the heuristic, that is present in (4.1), is run the same number of times as the desired number of chromosomes for the initial population, where on each run is generated one and only one chromosome. The initialization begins to receive in each iteration the table with all the relevant information from the customers and assign each customer to a type of vehicle having into account the price of the vehicles. After, for each type of vehicle, the Saving heuristics is run and the set of routes for that type of car is made. With this an initial chromosome is made.

The HGA begins to receive the initial chromosomes and using the selection process presented in subsection (4.2.3), the number of chromosomes equal to the size of the population intended is selected. This is a parameter that needs to be set up. After the HGA is run during some iterations until it reach the stopping criteria, which is also a set up parameter. In each iteration a predefined number of offsprings, set up parameter, will be generated. The offsprings will be generated using the reproductive process presented in (4.2.2) and Local search present in (4.3). Each reproductive processes have a predefined probability of occurring. After generating the desired number of offsprings, the offsprings and the population are merged and a sort is made according the fitness of the chromosomes. This sort is necessary to do the elitism process where a predefined number of chromosomes. This parameter needs to be set up, and it will pass directly to the population of the next generation.

The Improvement receives the best chromosome from the HGA and a Local Search (LS) can be made in order to try to improved the solution. The LS used can be seen in (4.3). Note that the LS can be fully independent of the HGA, to be able to run the LS is only necessary a solution to be improved, and the HGA can also be run without the LS.

5.4 Parameters configurations

5.4.1 First Proposed algorithm

To be able to used the proposed algorithm, represented in figure (4.1), applied to our specific problem is necessary to do the configuration of the necessary parameters, and to know how the variation of the parameters influences the result. The parameters that are needed to be configured so that the Genetic

Algorithm (GA) can be used are, the probability of using each of the different reproductive processes, the number of chromosomes used on the population and the number of offspring generated in each generation, the number of chromosomes that are part of the elitism, and the stopping criterion. It is not possible to vary all the parameters at the same time, that would be quite a large number of solutions that would have to be tested. For that reason the parameters will be set up independently but always using the parameters that where set up before. To set up the parameters, is necessary to know what is the order of the parameters that are going to be tested first.

The first parameter that is going to be set up is the probability of the different reproductive processes to occur, this is the most important parameter to be configured because this parameter will influence the others largely, due to this being the parameter that changes the way the offspings are generated. After it will be set up the size of the population used and the number of offspirngs generated in each iteration. These parameters will influence the way the algorithm evolves. Then the set up of the number of chromosomes affected by the elitism process is done, and finally the stopping criterion.

All the sensibility analysis and the parameter configuration is going to be made in the data set of week 24 Wednesday, this data-set was chosen because is the data-set that is in the mean of the year and is the one that have the most customers, so will be the day that the algorithm need more time to run. This data-set have a total of 233 customers to serve 982 palettes, since each vehicle load up to 33 pallets it takes at least 30 cars to supply all the customers.

The first thing that was done, to gain some sensibility of the algorithm was to change the probability from 1/6 to 1/6 in all of the different reproductive processes. Since we have 3 parameters all the possible case are:

Table 5.3: Probability variation of Sub-tour reverse

Probability Crossover / Probability Mutation	0	1/6	2/6	3/6	4/6	5/6	6/6
0	6/6	5/6	4/6	3/6	2/6	1/6	0
1/6	5/6	4/6	3/6	2/6	1/6	0	-
2/6	4/6	3/6	2/6	1/6	0	-	-
3/6	3/6	2/6	1/6	0	-	-	-
4/6	2/6	1/6	0	-	-	-	-
5/6	1/6	0	-	-	-	-	-
6/6	0	-	-	-	-	-	-

As it is possible to see in table (5.3) that the probability of the 3 reproductive processes must be equal to 1, this is, each offspring will be created by one of those three processes. If we choose for example a probability of 3/6 for the mutation and a probability of 1/6 for the probability of the crossover, obligatorily the probability of the sub-tour reverse must me equal to 2/6.

To run the initial solution a population size of 50 chromosomes was chosen which will generate 50 offspirngs in each generation, 5 chromosomes were chosen to be part of the elitism process. The Local

Search (LS) was used after the GA to see if the improvement helps achieves a better solution. The stopping criteria was chosen to be 2 hours where 1h45m were spend on the GA and 15m were spend on the LS. Since the objective is to run the algorithm in around 2 hours. The algorithm was run 3 times for each parameters since those algorithm depends on probability and is initialization dependent. Ideally the algorithm should be run endless times and the mean of the solution should be analyzed, but since a big computation time exist the algorithm run only 3 times for each set of parameters. Once more is going to be stressed that this first run is only to gain some sensibility of the algorithm. Having the results in the table (5.4).

Table 5.4: Fitness obtain in the first test with the variation of the probability of the reproductive process

	Crossover Probability													
	0		1/6		2/6		3/6		4/6		5/6		6/6	
Mutation Probability	Min	Mean	Min	Mean	Min	Mean	Min	Mean	Min	Mean	Min	Mean	Min	Mean
0	29113	30208	30127	30894	30223	30826	30155	30853	30463	31038	30460	30829	30050	31047
1/6	22043	22561	22780	23579	21797	22558	22621	23361	22579	22944	21373	22898	-	-
2/6	20645	21206	21648	21735	20963	21593	19714	21325	21333	21700	-	-	-	-
3/6	18902	19687	18930	19988	19551	20203	20638	21341	-	-	-	-	-	-
4/6	18758	19086	19455	20231	19010	20064	-	-	-	-	-	-	-	-
5/6	19381	19478	19088	19717	-	-	-	-	-	-	-	-	-	-
6/6	**18309**	**18957**	-	-	-	-	-	-	-	-	-	-	-	-

If we fixe the probability of the mutation and if we vary the probability of crossover, and the probability of the sub-reverse indirectly, it is obtained the graph (5.1).

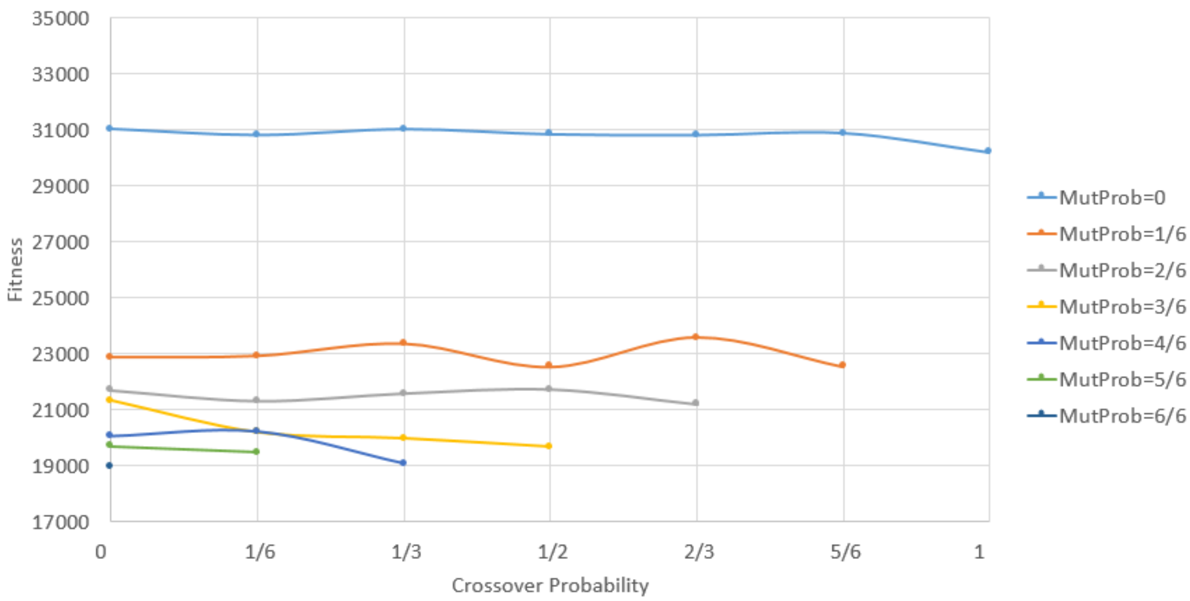


Figure 5.1: Graphical representation of the average solution fitness of table 5.4

In table (5.4) is possible to notice that the best parameters both for the minimum and for the mean are when the probability of the mutation operation is 1, this is, 100% of the chromosomes are created using the mutation operation. if the results are analyzed using the graph (5.1) is easy to see that the increasing of the probability of the mutation improves the results and the increasing of the probability of the crossover does not change the solution very much. Note that those results are only valid if the algorithm stops after 2 hours. If the best solution obtained is analyzed we have a final fitness of 18309 €, using 38 vehicles with a mean efficiency of 0.783 and leaving 0 pallets at the depot. The evolution of the best solution is shown in figure (5.2).

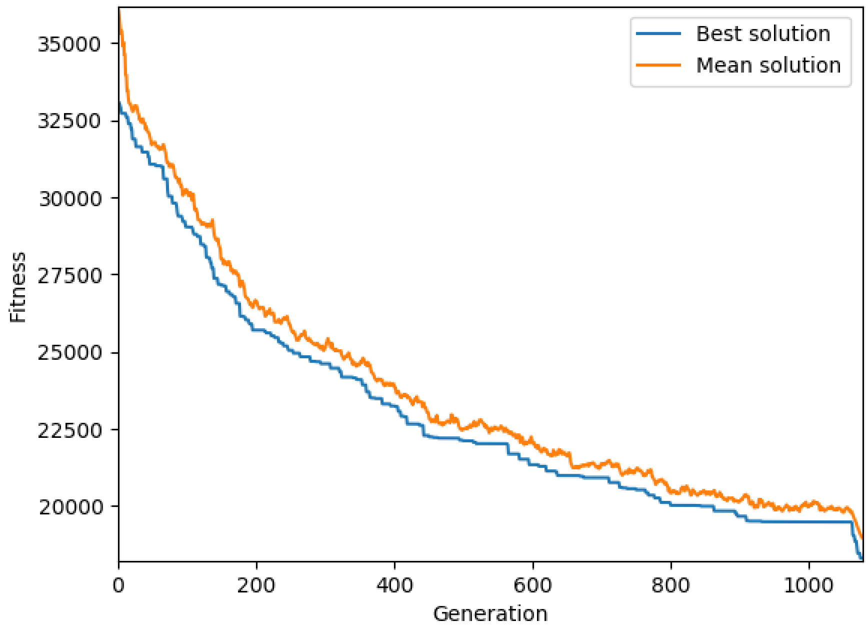


Figure 5.2: Evolution of the best solution obtain in the first test

Is easy to see in table (5.4) that using this algorithm with very low parameters configurations already achieves a solution, better that the solution of the baselines, in this data-set, with an efficiency almost as good. In figure (5.2) is possible to see the best solution in each generation and the mean solution in each generation, this is, the sum of the fitness of all solutions of the population divided by the size of the population. It is also possible to see in figure (5.2) that the part of the GA of the proposed algorithm, introduced in (4.1), has already begun to converge. Finally it was also possible to see that even though the LS takes a long time to run, the use of the LS at the end of the algorithm may be advantageous.

5.4.2 Hybrid Genetic Algorithm

With the results obtain in table (5.4) and in graph (5.1) the conclusion was reached that it is only necessary to run combinations where the probability of the mutation operation is bigger than 0.5, this happens because with probabilities less than 0.5 for the mutation the algorithm is not able to evolve. Another conclusion was that the LS greatly improves the solution in a way that the GA is not able to, but also uses a big computation time. For that reason the algorithm will be run ranging between GA and adaptation of LS. It was opted to run 5 generation of the adaptation after every 550 generation of GA. The adaptation of the LS is very similar to the LS that was presented in section (4.3) but only 1 iteration of the algorithm is done to generate a new ofspring and only the 5 chromosomes with the best fitness can be selected, just like it was explain in section (4.4). Each generation of the adapted LS will generate 5 offsprings, and each generation takes about 1 minutes so in each period of the LS the algorithm will take around 5 minutes. The GA will continue to generate 50 offsprings per generation, and each generation takes around 1.5 seconds so in each period of the GA the algorithm will take around 15 minutes. This new algorithm is called Hybrid Genetic Algorithm (HGA). This division causes about 3/4 of the time to be spent in GA and the remaining 1/4 in the LS. Note that it is not possible to transform the LS in one of the reproductive processes like it was done in Chand et al. [69] because each generation would take too much time, which would make the algorithm not able to evolve, so a compromise was reached. With this was run the HGA with the probability of the mutation operation ranging from 0.5 to 1 from 1/6 to 1/6, whit a stop criteria of a maximum of 8h or 500 generations without evolution, the results are in table (5.5):

Table 5.5: Fitness obtain with the variation of the probability of the reproductive process

	Crossover Probability							
	0		1/6		2/6		3/6	
Mutation Probability	Min	Mean	Min	Mean	Min	Mean	Min	Mean
3/6	15922	16063	15875	16233	**15803**	**16025**	16282	16301
4/6	15875	16037	16164	16186	16090	16296	-	-
5/6	15958	16340	16504	16557	-	-	-	-
6/6	15971	16494	-	-	-	-	-	-

If once more the probability of the mutation is fixed and if the probability of crossover is varied, and the probability of the sub-reverse indirectly, it is obtain the next graph:

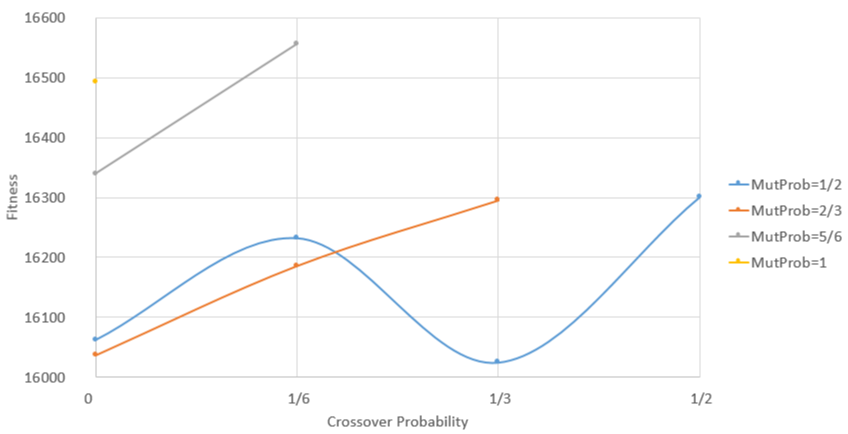


Figure 5.3: Graphical representation of the average solution fitness of table 5.5

In table (5.5) is possible to see that the best result, again both for the minimum fitness and the mean fitness, is when the probability of the mutation is 3/6, the probability of the crossover is 2/6 so the probability of the sub-tour must be equal to 1/6. If the graph (5.3) is analysed, it can be seen that the variation of the parameters is not so straightforward as in graph (5.1), in this graph seems that the less the probability of the mutation the better the results and the increasing of the crossover the worst the results. Even so it is possible to see that the best result is in between the values chosen so it was chosen and fixed those values for the probability of the different reproductive processes. One thing that is also possible to see is that all the results are better than the ones in the first test. This is due to the utilization of the LS in between generation and also due to changing the stopping criteria, so the new strategy chosen seems to achieve better results. If the best solution is analysed, it is obtain a fitness of 15803€, with a total of 32 routes that have a mean efficiency of 0.9299 and leaves 0 pallets on the depot, as we saw in the begging of section (5.4) the optimal solution in theory need to have at least 30 vehicles so this solution only uses two more vehicles, so it is possible to conclude that the solution is not too far from the optimal solution. If the evolution of the best solution is analysed, it is obtain:

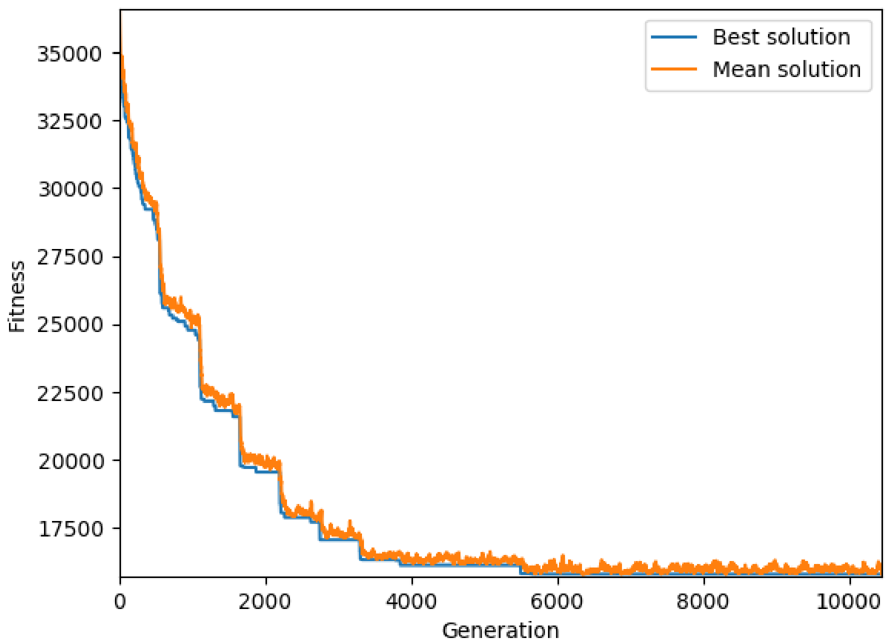


Figure 5.4: Evolution of the best solution obtain in the reproductive process parameterization

In figure (5.4) is easy to distinguish the difference from when the adaptive LS is used and when the GA is used, in the Adaptive LS a great step towards the final solution is given and in the GA the evolution is much slower. Once again this difference in evolution is due to the fact that LS directly takes into account the objective function while GA does not. It is also possible to notice that after the generation 8000 the algorithm is not able to evolve anymore. In the end when the LS is used the solution is not able to evolve anymore, this happens because the solution is already very close to the optimal and the flexibility is basically nil, and the LS has no ability to leave its respective local minimum. For the reasons stated before the algorithm from now on will be run with only 8300 iteration and the LS is not going to be used in the end of the HGA. With this, the algorithm proposed in section (4.4) is reached.

Now with the probability of using each of the different reproductive processes fixed, is going to be chosen what is the best size population and what should be the number of offsprings generated in each iteration. The selection of the size of the population and number of offspings should not be selected independently because in theory the population size is the possible search space, this is, the number of solutions that can be exploited, and the number of offsprings is how long is the search in this space. For the reason mention before the number of offsprings that will be generated will be depend on the size of the population, the number of offsprings can be 0.5, 1, 1.5 or 2 times the size of the population. Population size will range from 50 to 200 with an interval of 50 in 50. The fitness obtained with the variation of those parameters appear in the table (5.6).

Table 5.6: Fitness obtained with the variation of the size of the population and the number of offsprings

	Offsprings/Population							
	0.5		1		1.5		2	
Population	Min	Mean	Min	Mean	Min	Mean	Min	Mean
50	15654	**15846**	16119	16299	16359	16604	16194	16387
100	**15607**	15984	15885	16069	15924	16044.7	15968.0	16261.0
150	16129	16342.7	16323	16463	15726	16067	15729	16157
200	16008	16145	15735	16260	-	-	-	-

In table (5.6) is possible to see that the the best mean solution is when the algorithm uses a size of 50 individuals and generate 25 offsprings per generation, but is also possible to notice that the minimum solution found was when the algorithm uses a population of 100 individuals and generate 50 offspings per generation. The most important parameter of choice is the mean because the algorithm is only going to be run 1 time, with this, the parameters set of (50, 0.5) was chosen. One thing that is not mentioned in the table (5.6) is the running time of the algorithm with the different parameters. The more offsprings generated the higher the computational time required by the algorithm. The set of parameters (200, 1.5), (200, 2) where not run since the set of parameters (200, 1) took about 20h to run, so increasing the number of offsprings would take even longer and the algorithm would not able to be run with the expected time. One thing that was expected to happen, that didn't happen, was the improvement of the solution with the increasing of the size population, because there was more search space to found new solutions. This probably did not happen due to the maximum number of generations used, but a weighting between the final result and the running time is necessary. The final parameters chosen for the size of the population and the number of offsprings generated was (50,0.5), because a good solution occurs and the algorithm runs in reasonable computing time, 4,5h.

The last set of parameters that are necessary to test are the number of chromosome that are used in the elitism process. If a large number of chromosomes are used, the solution theoretical will converge too quickly because less bad solution are explored, but if a small number of chromosomes are used the algorithm may always be looking for new solutions and never converge. Note that the selection process of the new generation always exists so the best solution will always tend to survive more than the other, and the algorithm will theoretically converge, but the best solution can also be lost in the process. The algorithm was run with a variation in the number of chromosomes used in the elitism process between 0% and 20% of size of the population. The algorithm was run 5 times for each parameter set, because this will be the last parameter set and because there are a small number of parameters that need to be tested, which makes the algorithm capable of executing a larger number of times. The results obtained are represented in table (5.7).

Table 5.7: Fitness obtain with the variation of the number of chromosomes in the elitism process

Nº chromosomes in elitims process											
0		1		3		5		7		10	
Min	Mean	Min	Mean	Min	Mean	Min	Mean	Min	Mean	Min	Mean
15862	16320.4	16002	16237.4	15987	16420.6	**15267**	**16056**	15654	16326	15857	16215.8

The plot of the variation of the mean value of the fitness obtain by vary the number of chromosomes on the elitism process is represented in figure (5.5).

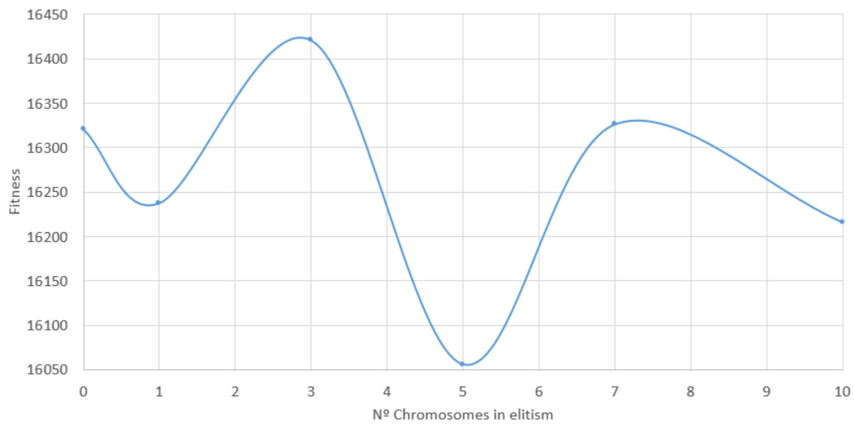


Figure 5.5: Graphical representation of the mean solutions of table 5.7

Both in table (5.7) and in figure (5.5) is possible to notice that the algorithm gives better results when the number of chromosomes on the elitims process is equal to 5. Just like it was explained before this happens because if the number of chromosomes used is to small the best solutions may be lost and the exploration of those solution may not occur, but if the number of chromosomes used it to large a big intensification of those solution may occur and the search space is less sought after. Other thing that is also possible to notice is that the best mean solution in (5.7) is worst that the best mean solution in (5.6). This happen because as it was said before this algorithm is based in probabilities and initialization which causes some variations of the results. Also in table (5.7) the number of experiments was higher than the ones of table (5.6) which makes the result less influenced by each solution and gives a more real value. The evolution of best solution obtain can be seen in figure (5.6).

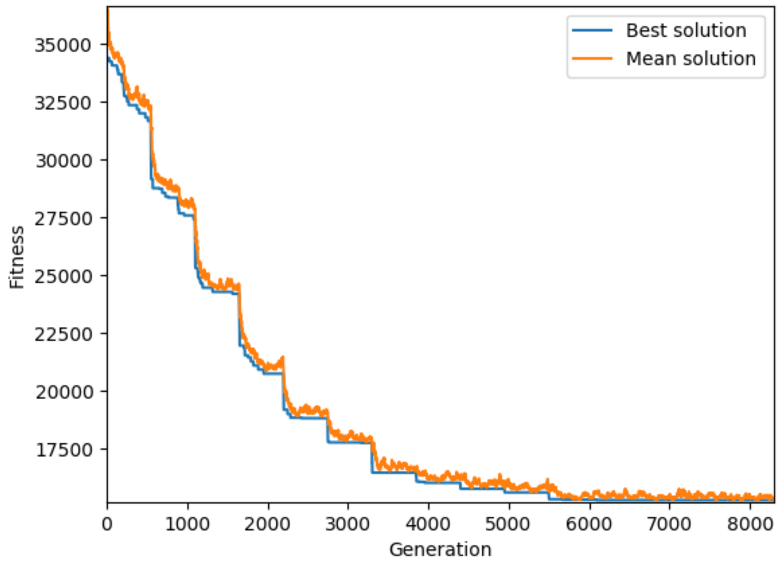


Figure 5.6: Evolution of the best solution obtain in the elitism parameterization

The best solution obtained has a fitness value of 15267€, using 31 vehicles to supply all the customers with an mean efficiency of 0.96 and leaving 0 pallets on the depot. If we compare this solution with the baseline solution, presented in (5.2) the fitness obtain is 30% less using the first approximation and leaving less 171 pallets on the depot, using the same number of vehicles. If we compare the solution with the second approximation, the fitness is around less 0.27% and uses less 16 vehicles. In conclusion, is possible to see that for this specific day the improvements using the algorithm are very relevant. Finally if we analyze figure (5.6), is possible to see that the convergence of the results already happen when the algorithm stops. It is also possible to see that the mean solution is always oscillating even when there are no more improvement in the best solution, this indicates that the algorithm is still looking for new solution and those solution are not equal to the best ones. Note that the solutions obtain in this parameterization are equal to the solution of the final algorithm, since the all the parameters are the same, so the the total analysis done earlier is equivalent to the analysis made using the final algorithm. A validation parameter that has not yet been analyzed is the variation of the algorithm, using the standard deviation in the 5 runs with the correct parameters is obtain 3.03%, this means that if the algorithm is run 5 times it has an average deviation of 3.03%.

In this section it was conclude that the best parameters to be used in the algorithm for this specific problem are the ranging between the adaptive LS and the GA as a reproductive process. For the reproductive process in the GA a probability of 0.5 for the mutation, a probability of 0.3334 for the crossover and a probability of 0.1666 for the sub-tour is used. A population size of 50 chromosomes and a generation of 25 offsprings per generation. For the adaptive LS a generation of 5 chromosomes per generation using only the best 5 chromosomes of the population. A total of 5 chromosomes in the elitism process and finally a stopping criteria of 8300 iterations.

5.5 Local Search analysis

In this section is going to analysed the behavior of the Local Search (LS) algorithm that was presented in section (4.3). Once more the algorithm will be use the data set of Wednesday of week 24, again is going to be used this data set because just like it is possible o see in (5.2) is the day that have the bigger number of customers but also and more important is in the weak where the demands of the customers is average. The results obtained if the algorithm is run during 5 times, are in table (5.8).

Table 5.8: Fitness obtain using the Local Search

Min	Mean
16410	17425

Now if the evolution of the best solution using the LS is analysed, figure (5.7).

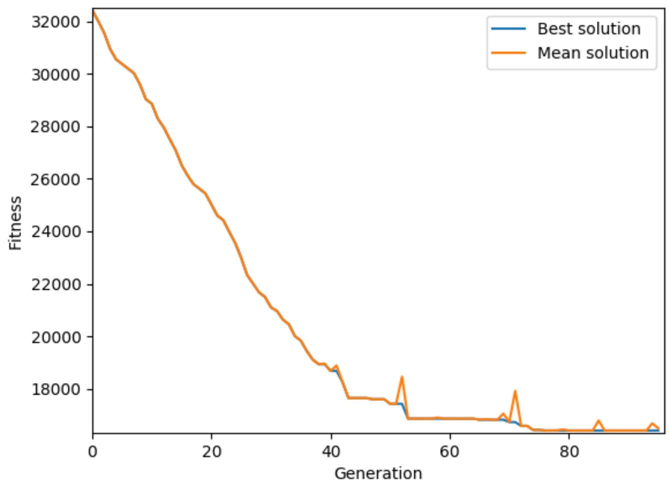


Figure 5.7: Evolution of the best solution obtain using the Local Search

In table (5.8) and in the figure (5.7) is possible to see that the best solution obtained has a fitness of 16410€. This solution uses 32 vehicles with a mean efficiency of 0.928 to supply all the customers leaving two pallets in the depot. Comparing this solution with the baseline solutions is possible to see that using this algorithm in this data set the fitness is less 25% that the baseline solution using the first approximation with less 169 pallets in the depot, and less 22% using when the second approximation is used. Now if we analyse the evolution of best solution obtain using the LS, that is present in figure (5.7), we can see that in the beginning the solution evolves very fast and after some time the evolution begins to converge. This happens because in the beginning of the algorithm the solution is very poor and the system still have a big flexibility to improvements. Over time the system starts to lose its flexibility and the improvement begins to stagnate. Now if we look at the number of generation used to achieve a final result we see that the number is much less that the number of iterations necessary in HGA, also is possible to see that the stopping criteria is appropriate because the LS stops when the solution already converged. Finally if we compare the best solution obtain with the solution obtain in each iteration, we can see that in the beginning the solutions are the same but with time the algorithm begins to make the solution worse, so the LS does not guarantee always an improvement just like it was explain in section (4.3).

5.6 Final Results

In this section is going to be run the Genetic Algorithm (GA) ranging with the Local Search (LS), also called Hybrid Genetic Algorithm (HGA), and the the LS only 1 time for each data set, is also going to be transform the data sets in their clusters just like it was explained in subsection (3.1.2) and in subsection (5.2), and both algorithms will be run again 1 time only for each data set. The results of the fitness obtain are in table (5.9).

Table 5.9: Fitness of the final results in €

		Base1	Base2	HGA	HGA cluster	LS	LS cluster
Week 6	Monday	13775	13775	13271	**10962**	15168	11585
	Tuesday	7375	7375	7598	**6286**	8772	6298
	Wednesday	14225	14225	13574	12324	14438	**12181**
	Thursday	8155	8155	8420	**6574**	9089	6804
	Friday	12835	12835	12576	**9850**	13521	11501
Week 24	Monday	15525	16820	13209	**12086**	15211	13758
	Tuesday	10145	11320	8760	**7959**	9174	8483
	Wednesday	21804	20850	16180	**13993**	17466	15538
	Thursday	9420	11050	8520	**7263**	9186	7993
	Friday	13950	15310	10848	**9707**	11898	10075
Week 50	Monday	16530	17955	15006	**14398**	16718	15285
	Tuesday	12995	13820	13003	**11614**	13253	11789
	Wednesday	18600	20625	16601	**15190**	17812	16238
	Thursday	12905	13540	12523	11572	13342	**11417**
	Friday	16745	17405	14940	**13786**	16496	14971
	Mean	13665.6	14337.33	12335.27	**10904.27**	13436.27	11594.4

In table (5.9) is possible to notice that even if the algorithms are run only 1 time, all the algorithms gives almost always a better solution that both baseline solution. The best results are obtained when there is a cluster of the customers, this means that doing the cluster of the customers will result in a better solution. If the HGA and the HGA cluster are compared it is possible to see that clustering the customers lead always to better results, the same thing happens if the the LS and the LS cluster are compared. Other thing that is also easy to see is that the HGA almost always obtains better results than the LS, both with and without cluster. This happens because the HGA has more flexibility than the simple LS, if the LS gets stuck in a local minimal it is not able to get out of it unlike the HGA that is design to be able to get out of local minimal. One thing that is not possible to be see in the table is the computational time, but just like it was expect the time in running the HGA is around 3 to 4 hours and when the clusters are applied the time reduces in around 1 hour. The computational time of the LS is around 15 minutes and when the clustering is applied an almost immediate solution is obtain.

The number of cars used is a very important aspect due to the higher cost being associated with the vehicle cost. If we look at the number of cars that must be subcontract to the subcontractor is obtain table (5.10).

Table 5.10: Number of cars in the final results

		Base1	Base2	HGA	HGA cluster	LS	LS cluster
Week 6	Monday	29	29	26	**24**	28	**24**
	Tuesday	17	17	16	**15**	19	**15**
	Wednesday	29	29	26	**24**	27	25
	Thursday	19	19	18	**17**	18	**17**
	Friday	26	26	23	**22**	24	24
Week 24	Monday	28	37	**26**	**26**	28	28
	Tuesday	20	28	**19**	20	**19**	20
	Wednesday	**31**	47	32	**31**	33	33
	Thursday	19	27	**18**	**18**	**18**	**18**
	Friday	27	34	22	**21**	24	**21**
Week 50	Monday	35	41	**33**	**33**	**33**	34
	Tuesday	31	36	**30**	31	31	32
	Wednesday	40	50	**37**	38	38	40
	Thursday	29	37	**28**	29	29	29
	Friday	38	40	**32**	**32**	**32**	33
	Mean	27.87	33.13	25.73	**25.4**	26.73	26.2

In table (5.10) is possible to see that the number of vehicles used is not always minimum in the best solution. This happens because the objective function (3.5) depends not only on the number of vehicles but also on the way the routes are made, so the best solutions does not have to mean the minimum number of vehicle. Even so, is possible to see that in the first two weeks the best solutions have also a minimum number of vehicles. In week 50 is possible to see that this phenomenon does not occur, probably because in week 50 the mean number of pallets per customer is very high this makes the algorithms to lose flexibility in the number of vehicles used, but is able to compensate otherwise. In conclusion the minimum number of vehicles do not guarantee the minimum fitness but even so the HGA cluster still gave the minimum mean number of vehicles. Like it was expect the baseline solution using the second approximation gave the worst results. The LS algorithms always gives a bigger number of vehicles than the HGA because the LS is not completely capable of reducing the number of routes, which concludes that HGA brings together the best things about the GA and the LS.

Now if the mean efficient of the vehicles is analyzed, with this we have the table (5.11). The way to calculate the mean efficient is explain in subsection (3.1.5).

Table 5.11: Mean efficiency in the final results

		Base1	Base2	HGA	HGA cluster	LS	LS cluster
Week 6	Monday	0.792	0.792	0.882	**0.955**	0.817	0.94
	Tuesday	0.841	0.841	0.894	0.941	0.752	**0.943**
	Wednesday	0.800	0.800	0.900	**0.965**	0.86	0.924
	Thursday	0.836	0.836	0.883	**0.932**	0.879	0.9251
	Friday	0.800	0.800	0.906	**0.953**	0.870	0.870
Week 24	Monday	0.782	0.662	0.933	**0.939**	0.857	0.874
	Tuesday	0.806	0.644	**0.950**	0.903	0.938	0.892
	Wednesday	0.803	0.638	0.928	**0.958**	0.896	0.891
	Thursday	0.781	0.611	0.917	**0.919**	**0.919**	0.908
	Friday	0.686	0.602	0.915	0.952	0.842	**0.955**
Week 50	Monday	0.887	0.789	**0.969**	0.954	0.963	0.932
	Tuesday	0.911	0.807	**0.980**	0.942	0.944	0.9059
	Wednesday	0.876	0.737	**0.986**	0.954	0.957	0.903
	Thursday	0.912	0.753	**0.978**	0.935	0.943	0.932
	Friday	0.81	0.777	0.931	**0.954**	0.952	0.92
	Mean	0.822	0.739	0.930	**0.944**	0.893	0.914

The analysis of table (5.11) is very similar to the analysis made before for the table (5.10). This is because the mean efficiency of the vehicle is greatly influenced by the number of vehicles used. The other parameter that influence the mean efficiency is the number of pallets that are left on the depot. From the table (5.10) and (5.11) is possible to see that the mean efficiency is always smaller when the number of vehicles used it's minimal. Again is possible to notice that the HGA cluster not always give the best mean efficiency even so is the algorithm that gives the minimum on the mean of all data sets. Finally like before is possible to see that the the worst results are obtained in the baseline solution when the second approximation is used, and the LS gives normally worst results that the HGA for the same reason previously stated.

If we analyse the last relevant parameter, the number of pallets that are left on the depot, the table (5.12) is obtain.

Table 5.12: Number of pallets left on the depot in the final results

		Base1	Base2	HGA	HGA cluster	LS	LS cluster
Week 6	Monday	1	1	**0**	1	2	10
	Tuesday	**0**	**0**	**0**	6	**0**	5
	Wednesday	4	4	**0**	8	5	7
	Thursday	1	1	**0**	2	3	6
	Friday	5	5	4	**0**	3	3
Week 24	Monday	88	3	**0**	6	9	4
	Tuesday	64	1	**0**	**0**	8	7
	Wednesday	171	2	**1**	2	6	12
	Thursday	57	2	2	**1**	**1**	9
	Friday	68	1	3	7	**0**	5
Week 50	Monday	52	9	**1**	14	6	6
	Tuesday	40	13	**2**	3	4	6
	Wednesday	69	9	**7**	12	9	13
	Thursday	59	13	6	9	**5**	11
	Friday	11	2	**1**	8	10	12
	Mean	46	4.4	**1.8**	5.27	4.73	7.73

Now if we analyse table (5.12), that has the amount of pallets that are left in the depot is possible to see that the algorithm that leave on average less pallets on the depot is the HGA followed by the baseline solution when the second approximation is used. This happens because the data set without the cluster are more flexible and are able to give good results without leaving many pallets in the depot and the HGA is the algorithm that can better do this weighting. One thing that can be see in the number of pallets in the depot, is that the baseline solution using the first approximation leaves much more pallets that all the other. This indicates that probably the company is not able to use this approximation to deliver the demands of the customers and the second approximation must to be used, or a compromise between the two. Finally is possible to notice that once more the HGA leaves on average less pallets that the LS and that in week 50 all the algorithms leave a lot more pallets than in the other two weeks, again this happen because the demand of the customers on week 50 is higher than in the other weeks.

Chapter 6

Conclusions

6.1 Conclusions

In this master thesis was developed two algorithms that solved a real problem proposed by the company Worten with the restrictions based on its own logistics. This problem can also be called Site Dependent Vehicle Routing Problem with Hard Time Windows (SDVRPHTW). The algorithms developed achieves better results than the solutions developed by the company. The Hybrid Genetic Algorithm is the one that achieves better results and the Local Search is able to find competitive results in those data sets in a shorter computational time than the hybrid algorithm. Transforming the customers into their clusters shifted the overall results from an improvement of almost 10% to an improvement of over 20%, although this transformation increases the number of pallets left in the depot.

The use of the Local Search in the Genetic Algorithm, the Hybrid Genetic Algorithm, greatly improved the solution due to the LS taking into account the objective function and not purely the randomness, but also requires a longer computational time. The hybrid algorithm implemented gave the good results because it's a population algorithm is able to not only improve the best individual, but also explore new solution using the other individuals, which allows great flexibility and has a good behavior along with very restrictive and highly non feasible problems. It is believed that this hybrid algorithm is able to give competitive results in other vehicle routing problems variants and those type of algorithms should be further investigated in the literature.

The hybrid algorithm has the problem that it needs intensive parameter calibrations, which requier a long preprocessing before the algorithm is ready to use.

6.2 Future work

The proposed approach can be improvement in many different ways. Even if the algorithm is prepared to take into account independent pickups this part of the algorithm was not able to be tested and compared due to the lack of information, so a validation of the of the reverse logistics is necessary to be able to use the this part in the algorithm. Other aspect that can also be improved in the reverse logistics is to do the collection and delivery in the same stop and check if there is a significant improvement. This strategy may reduce the number of used cars if it is necessary to deliver and collect to the same customers. Other aspect that is important to improve in the algorithm is to replace distance estimates with actual distances using an API. This approach will make the feasibility of the routes more real. Even so using estimates of distances, it is easy to prove that using these algorithms will improve the baseline solutions. A cluster to the zones could also be applied, this would decrease the number of non feasible solutions generated because two clients from different zones may not be able to be on the same route, which would make the running of the algorithm more efficient. The final improvement would be the necessity of improving the computational time used, since the algorithm needs to run between 2/3 hours daily, this improvement could be made after by a code optimization expert or by parallelizing the algorithm.

This algorithm can also be used to assist in deciding each customer's delivery windows, that is, since the algorithm is able to provide near optimal routes, the customer input parameters, such as delivery windows, can be changed and checked which are the best entry parameters that optimize the overall fitness. This change in the algorithm can greatly improve the total cost of the transportation.

Bibliography

[1] G. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[2] A. Hoff, H. Andersson, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010.

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

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

[5] J. K. Lenstra and A. R. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11 (2):221–227, 1981.

[6] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo. Vehicle routing. *Handbooks in operations research and management science*, 14:367–428, 2007.

[7] P. Toth and D. Vigo. Branch-and-bound algorithms for the capacitated vrp. In *The vehicle routing problem*, pages 29–51. SIAM, 2002.

[8] P. Toth and D. Vigo. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(1-3):487–512, 2002.

[9] H. Nazif and L. S. Lee. Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, 36(5):2110–2117, 2012.

[10] P. Augerat, J.-M. Belenguer, E. Benavent, A. Corbéran, and D. Naddef. Separating capacity constraints in the cvrp using tabu search. *European Journal of Operational Research*, 106(2-3):546–557, 1998.

[11] A. Wren and A. Holliday. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*, 23(3):333–344, 1972.

[12] R. Liu, Z. Jiang, and N. Geng. A hybrid genetic algorithm for the multi-depot open vehicle routing problem. *OR spectrum*, 36(2):401–421, 2014.

[13] J. Renaud, G. Laporte, and F. F. Boctor. A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3):229–235, 1996.

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

[15] E. J. Beltrami and L. D. Bodin. Networks and vehicle routing for municipal waste collection. *Networks*, 4(1):65–94, 1974.

[16] R. Russell and W. Igo. An assignment routing problem. *Networks*, 9(1):1–17, 1979.

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

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

[19] C. Archetti and M. G. Speranza. The split delivery vehicle routing problem: a survey. In *The vehicle routing problem: Latest advances and new challenges*, pages 103–122. Springer, 2008.

[20] M. Dror and P. Trudeau. Savings by split delivery routing. *Transportation Science*, 23(2):141–145, 1989.

[21] C. Archetti and M. G. Speranza. Vehicle routing problems with split deliveries. *International transactions in operational research*, 19(1-2):3–22, 2012.

[22] C. Archetti, M. G. Speranza, and A. Hertz. A tabu search algorithm for the split delivery vehicle routing problem. *Transportation science*, 40(1):64–73, 2006.

[23] C. Archetti, N. Bianchessi, and M. G. Speranza. Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*, 238(3):685–698, 2014.

[24] W. R. Stewart Jr and B. L. Golden. Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research*, 14(4):371–385, 1983.

[25] M. Gendreau, G. Laporte, and R. Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

[26] F. A. Tillman. The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3):192–204, 1969.

[27] M. Reimann. Analyzing a vehicle routing problem with stochastic demand using ant colony optimization. *Advanced OR and AI methods in Transportation, Poznan Technical University, Poznan*, pages 764–769, 2005.

[28] G. Laporte, F. Louveaux, and H. Mercure. The vehicle routing problem with stochastic travel times. *Transportation science*, 26(3):161–170, 1992.

[29] M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42(1):39–51, 1989.

[30] P. Toth and D. Vigo. Vrp with backhauls. In *The vehicle routing problem*, pages 195–224. SIAM, 2002.

[31] E. E. Zachariadis and C. T. Kiranoudis. An effective local search approach for the vehicle routing problem with backhauls. *Expert Systems with Applications*, 39(3):3174–3184, 2012.

[32] I. Deif and L. Bodin. Extension of the clarke and wright algorithm for solving the vehicle routing problem with backhauling. In *Proceedings of the Babson conference on software uses in transportation and logistics management*, pages 75–96. Babson Park, MA, 1984.

[33] P. Toth and D. Vigo. An exact algorithm for the vehicle routing problem with backhauls. *Transportation science*, 31(4):372–385, 1997.

[34] Y. Gajpal and P. L. Abad. Multi-ant colony system (macs) for a vehicle routing problem with backhauls. *European Journal of Operational Research*, 196(1):102–117, 2009.

[35] G. Desaulniers, J. Desrosiers, A. Erdmann, M. M. Solomon, and F. Soumis. *The VRP with pickup and delivery*. Montréal: Groupe d'études et de recherche en analyse des décisions, 2000.

[36] H. Min. The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5):377–386, 1989.

[37] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31, 2007.

[38] S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft*, 58(1):21–51, 2008.

[39] J. Dethloff. Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1):79–96, 2001.

[40] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation science*, 39(1):119–139, 2005.

[41] M. Gendreau and C. D. Tarantilis. *Solving large-scale vehicle routing problems with time windows: The state-of-the-art*. Cirrelt Montreal, QC, Canada, 2010.

[42] J. Homberger and H. Gehring. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*, 162(1):220–238, 2005.

[43] M. M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[44] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.

[45] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. The vehicle routing problem, chapter vrp with time windows. *SIAM, Society for Industrial and Applied Mathematics, Philadelphia, USA*, pages 157–193, 2000.

[46] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation science*, 39(1):104–118, 2005.

[47] O. Bräysy. Fast local searches for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 40(4):319–330, 2002.

[48] R. Baldacci, M. Battarra, and D. Vigo. Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges*, pages 3–27. Springer, 2008.

[49] B. Golden, A. Assad, L. Levy, and F. Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

[50] P. H. V. Penna, A. Subramanian, and L. S. Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232, 2013.

[51] É. D. Taillard. A heuristic column generation method for the heterogeneous fleet vrp. *RAIRO-Operations Research*, 33(1):1–14, 1999.

[52] F. Semet and E. Taillard. Solving real-life vehicle routing problems efficiently using tabu search. *Annals of Operations research*, 41(4):469–488, 1993.

[53] G. B. Dantzig. *Linear programming and extensions*. Princeton university press, 1998.

[54] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer, 2010.

[55] J. Clausen. Branch and bound algorithms-principles and examples. *Department of Computer Science, University of Copenhagen*, pages 1–30, 1999.

[56] F. S. Hillier and G. J. Lieberman. *Introduction to operations research*. McGraw-Hill Science, Engineering & Mathematics, 1995.

[57] G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, 1983.

[58] D. Naddef and G. Rinaldi. Branch-and-cut algorithms for the capacitated vrp. In *The vehicle routing problem*, pages 53–84. SIAM, 2002.

[59] S. Raff. Routing and scheduling of vehicles and crews: The state of the art. *Computers & Operations Research*, 10(2):63–211, 1983.

[60] P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK*, 1997.

[61] J.-Y. Potvin and J.-M. Rousseau. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

[62] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & operations research*, 13(5):533–549, 1986.

[63] M. Gendreau, J.-Y. Potvin, O. Bräumlaysy, G. Hasle, and A. Løkketangen. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. In *The vehicle routing problem: latest advances and new challenges*, pages 143–169. Springer, 2008.

[64] J. H. Holland et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[65] B. M. Baker and M. Ayechew. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800, 2003.

[66] S. Anbuudayasankar, K. Ganesh, S. L. Koh, and Y. Ducq. Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. *Expert Systems with Applications*, 39(3):2296–2305, 2012.

[67] W. Ho, G. T. Ho, P. Ji, and H. C. Lau. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557, 2008.

[68] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[69] P. Chand, B. S. P. Mishra, and S. Dehuri. A multi objective genetic algorithm for solving vehicle routing problem. *International Journal of Information Technology and Knowledge Management*, 2 (2):503–506, 2010.

[70] A. S. Tasan and M. Gen. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. *Computers & Industrial Engineering*, 62(3):755–761, 2012.

[71] F. Glover. Tabu search—part i. *ORSA Journal on computing*, 1(3):190–206, 1989.

[72] J.-F. Cordeau and G. Laporte. A tabu search algorithm for the site dependent vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 39(3):292–298, 2001.

[73] M. Dorigo, V. Maniezzo, A. Colorni, et al. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics*, 26(1):29–41, 1996.

[74] S. Balseiro, I. Loiseau, and J. Ramonet. An ant colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. *Computers & Operations Research*, 38(6):954–966, 2011.

[75] J. Tang, Y. Ma, J. Guan, and C. Yan. A max–min ant system for the split delivery weighted vehicle routing problem. *Expert Systems with Applications*, 40(18):7468–7477, 2013.

[76] C. B. Kalayci and C. Kaya. An ant colony system empowered variable neighborhood search algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, 66:163–175, 2016.

[77] Calculate distance, bearing and more between Latitude/Longitude points. `https://www.movable-type.co.uk/scripts/latlong.html`, 2019. [Online; accessed 23-September-2019].

[78] Regulation (EC) No 561/2006. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02006R0561-20150302`, 2015. [Online; accessed 23-September-2019].

[79] E. Zare-Reisabadi and S. H. Mirmohammadi. Site dependent vehicle routing problem with soft time window: Modeling and solution approach. *Computers & Industrial Engineering*, 90:177–185, 2015.

[80] D. Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

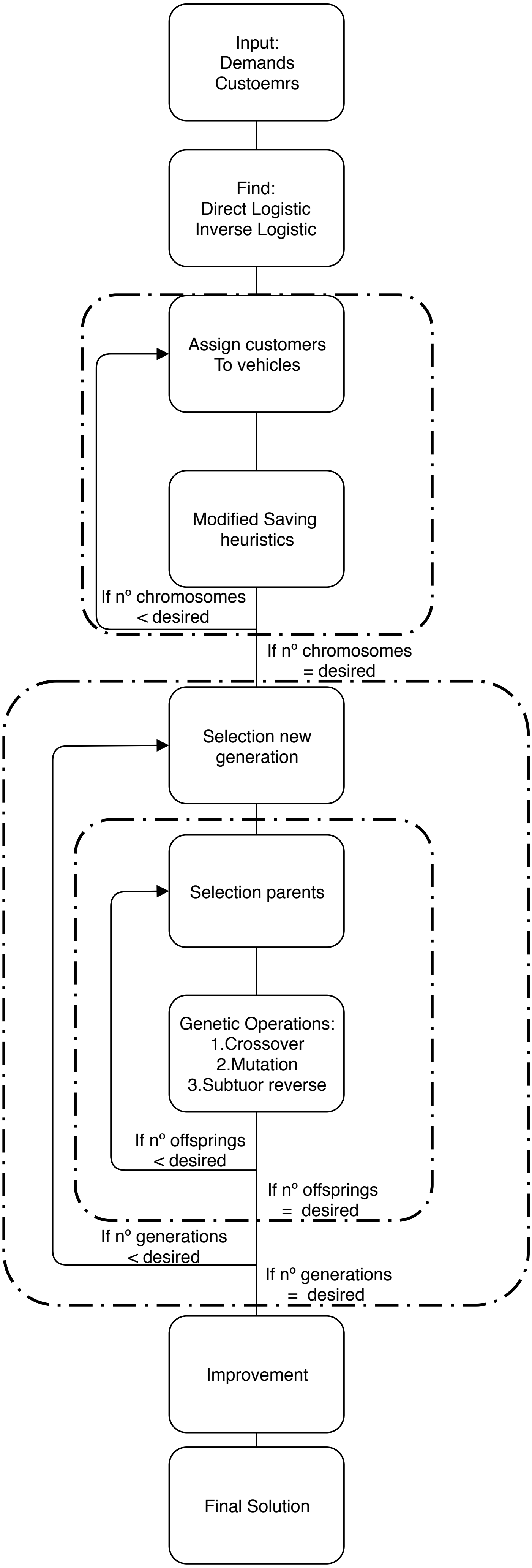
[81] R. Cheng and M. Gen. *Genetic algorithms and engineering design*. John Wiley, 1997.

[82] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989. ISBN 0201157675.

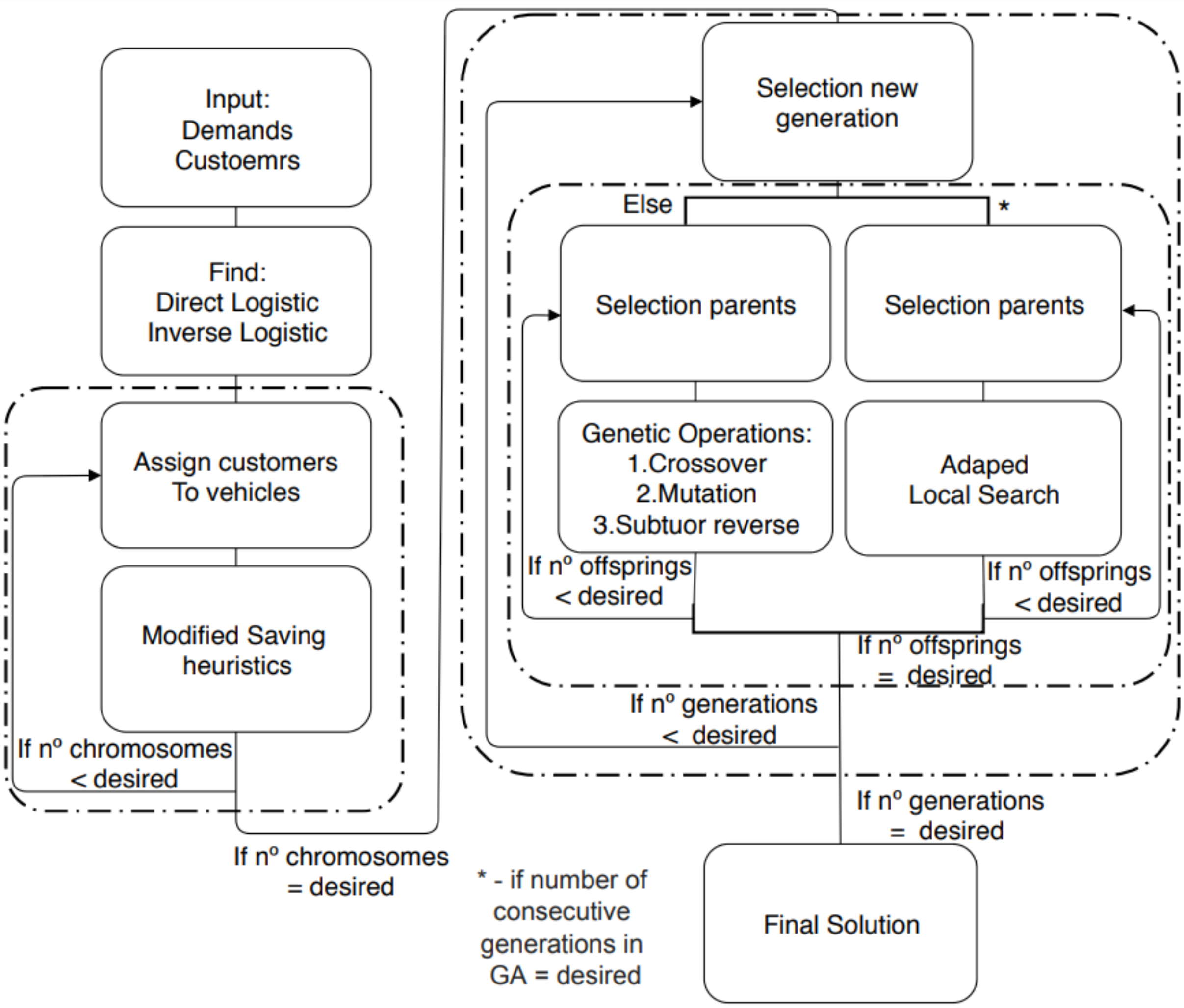
Appendix A

apendix

A.1 Flowchart of the first proposed algorithm


Input:
Demands
Custoemrs
Find:
Direct Logistic
Inverse Logistic
Assign customers
To vehicles
Modified Saving
heuristics
If nº chromosomes
< desired
If nº chromosomes
= desired
Selection new
generation
Selection parents
Genetic Operations:
1.Crossover
2.Mutation
3.Subtuor reverse
If nº offsprings
< desired
If nº offsprings
= desired
If nº generations
< desired
If nº generations
= desired
Improvement
Final Solution

A.2 Flowchart of the last proposed algorithm



A.3 Extended pseudo algorithm of the initial construct algorithm

Algorithm 1 Initial construct algorithm

INPUT: Customers with demands
OUTPUT: Routes

```
for each customer do
    Calculate the probability of a customer be assigned to a feasible vehicle
    Assigned the customer to a vehicle
end for

for Each type of vehicle do
    see assigned customers
    initialize all customers with "not assigned"
    For all customers do an initial route with only the depot and that customer
    For every pair of customers calculate the "Saivings" of join the routes
    while any Saiving>0 do
        Find max(Saiving) → pair of customers
        if Both customers are "not assigned" then
            Try to assign the second customer to the route of the first customer
            if route=feasible then
                Assign second customer to the route of the first customer and delete the route of second customer

                pass both customer form "not assigned" to "assigned"
                pass all "Saiving" of customer 1 to negative ones
            else
                max(Saiving) → negative
            end if
        else if 1 customers is "not assigned" and 1 customers is "assigned" then
            Try Assign "not assigned" customer to the route of the "assigned" customer
            if route=feasible then
                Assign "not assigned" customer to the route of the "assigned" customer and delete route of the
                "not assigned" customer
                pass all Saiving of customer "assigned" to negative ones
                pass "not assigned" customer "assigned"
            else
                max(Saiving) → negative
            end if
        else if Both customers are "assigned" then
            max(Saiving) → negative
        end if
    end while
    Return:Routes
end for
```

Figure A.1: psedocode2

A.4 Initialization illustration small example

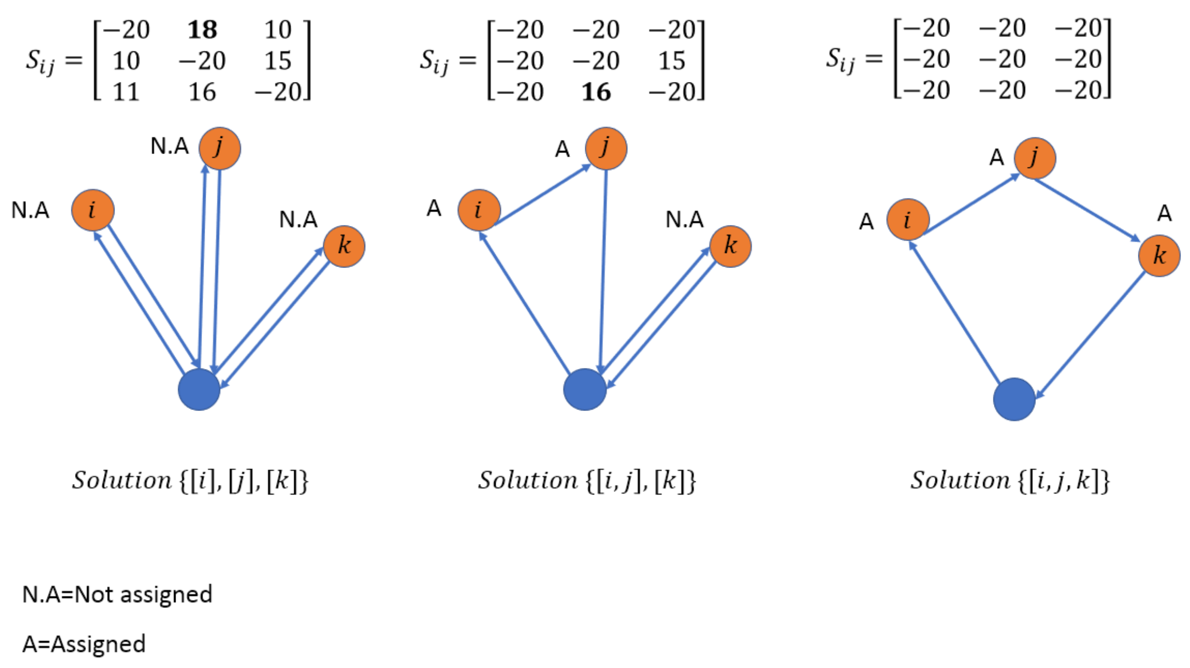


Figure A.2: itialization example

A.5 Extended baseline solution analysis

A.5.1 Week 6

The week 6 just like it was explain before is the a typical week where the demand of the clients is not very high, this is there are not many pellets per clients and also not many customers that need to be supply. The characteristics of the days and the baseline solution are analyse next day by day.

Monday

In the first day of week 6, Monday, there are 212 different customers that need to be supply that have a total demand of 757 pallets, so each customer has on average a demand of 3.57 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 23 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, where two pallets can be left in the depot, are necessary at least 22 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers just like it is explain in subsection (3.1.2) we end with a total of 97 customers with an average a demand of 7.8 pallets

If we analyse the baseline solution for Monday of week 6, it is possible to notice that there are used 29 vehicles with an average efficiency of 0.79. The baseline solution is the same using first or the second the approximation because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The fit of this solution using the objective function is 13775.

Tuesday

In the second day of week 6, Tuesday, there are 124 different customers that need to be supply that have a total demand of 472 pallets, so each customer has on average a demand of 3.8 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 15 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, are necessary at least 14 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 63 customers with average a demand of 7.49 pallets

If we analyse the baseline solution for Tuesday of week 6, it is possible to notice that there are used 17 vehicles with an average efficiency of 0.841. The baseline solution is the same using first or the second the approximation once more because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The final fit is 7375.

Wednesday

In the third day of week 6, Wednesday, there are 213 different customers that need to be supply that have a total demand of 772 pallets, so each customer has on average a demand of 3.62 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 24 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, are necessary at least 23 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 100 customers with average a demand of 7.72 pallets

If we analyse the baseline solution for Wednesday of week 6, it is possible to notice that there are used 29 vehicles with an average efficiency of 0.8025. The baseline solution is the same using first or the second the approximation once more because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The final fit is 14225.

Thursday

In the fourth day of week 6,Thursday, there are 141 different customers that need to be supply that have a total demand of 525 pallets, so each customer has on average a demand of 3.72 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 16 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, are necessary at least 15 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 66 customers with average a demand of 7.95 pallets

If we analyse the baseline solution for Thursday of week 6, it is possible to notice that there are used 19 vehicles with an average efficiency of 0.836. The baseline solution is the same using first or the second the approximation once more because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The final fit is 8155.

Friday

In the last day of week 6,Friday, there are 196 different customers that need to be supply that have a total demand of 692 pallets, so each customer has on average a demand of 3.53 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 21 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 20 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 92 customers with average a demand of 7.52 pallets

If we analyse the baseline solution for Friday of week 6, it is possible to notice that there are used 26 vehicles with an average efficiency of 0.80. The baseline solution is the same using first or the second the approximation once more because all the vehicles carry a total demand lower than the maximum capacity of the vehicle. The final fit is 12835.

A.5.2 Week 24

The week 24 just like it was explain before is the a typical week where the volume of demands of the customers is not to high or to low , this is there is a mediated number of customers each with a median volume. The characteristics of the days and the baseline solution are analyse next day by day.

Monday

In the first day of week 24,Monday, there are 169 different customers that need to be supply that have a total demand of 801 pallets, so each customer has on average a demand of 4.74 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 25 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 23 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 104 customers with average a demand of 7.7 pallets

If we analyse the baseline solution for Monday of week 24 using the first option, it is possible to notice that there are used 28 vehicles with an average efficiency of 0.782 and a final fit of 15525. If solution is analyse using the second option,there are used 37 vehicles with an average efficiency of 0.661 and a final fit of 16820

Tuesday

In the second day of week 24,Tuesday, there are 128 different customers that need to be supply that have a total demand of 596 pallets, so each customer has on average a demand of 4.65 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 19 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 18 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 73 customers with average a demand of 8.16 pallets

If we analyse the baseline solution for Tuesday of week 24 using the first option, it is possible to notice that there are used 20 vehicles with an average efficiency of 0.644 and a final fit of 10145. If solution is analyse using the second option,there are used 28 vehicles with an average efficiency of 0.661 and a final fit of 11320

Wednesday

In the third day of week 24,Wednesday, there are 233 different customers that need to be supply that have a total demand of 982 pallets, so each customer has on average a demand of 4.21 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 30 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 29 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 115 customers with average a demand of 8.54 pallets

If we analyse the baseline solution for Wednesday of week 24 using the first option, it is possible to notice that there are used 31 vehicles with an average efficiency of 0.802 and a final fit of 18745. If solution is analyse using the second option,there are used 47 vehicles with an average efficiency of 0.638 and a final fit of 20850.

Thursday

In the fourth day of week 24,Thursday, there are 117 different customers that need to be supply that have a total demand of 547 pallets, so each customer has on average a demand of 4.68 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 17 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 16 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 65 customers with average a demand of 8.42 pallets

If we analyse the baseline solution for Thursday of week 24 using the first option, it is possible to notice that there are used 19 vehicles with an average efficiency of 0.781 and a final fit of 9420. If solution is analyse using the second option,there are used 27 vehicles with an average efficiency of 0.612 and a final fit of 11050.

Friday

In the last day of week 24,Friday, there are 132 different customers that need to be supply that have a total demand of 667 pallets, so each customer has on average a demand of 4.05 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 21 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 20 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 81 customers with average a demand of 8.23 pallets

If we analyse the baseline solution for Friday of week 24 using the first option, it is possible to notice that there are used 27 vehicles with an average efficiency of 0.683 and a final fit of 13950. If solution is analyse using the second option,there are used 34 vehicles with an average efficiency of 0.602 and a final fit of 15310.

A.5.3 Week 50

The week 50 the demand of the clients is very high, this is there are many pellets per clients and also there are many customers that need to be supply. The characteristics of the days and the baseline solution are analyse next day by day.

Monday

In the first day of week 50,Monday, there are 160 different customers that need to be supply that have a total demand of 1056 pallets, so each customer has on average a demand of 6.6 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 32 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 31 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 93 customers with average a demand of 11.35 pallets

If we analyse the baseline solution for Monday of week 50 using the first option, it is possible to notice that there are used 35 vehicles with an average efficiency of 0.887 and a final fit of 16530. If solution is analyse using the second option,there are used 41 vehicles with an average efficiency of 0.789 and a final fit of 17955.

Tuesday

In the second day of week 50,Tuesday, there are 142 different customers that need to be supply that have a total demand of 972 pallets, so each customer has on average a demand of 6.85 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 30 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 28 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 76 customers with average a demand of 12.79 pallets

If we analyse the baseline solution for Tuesday of week 50 using the first option, it is possible to notice that there are used 31 vehicles with an average efficiency of 0.911 and a final fit of 12995. If solution is analyse using the second option,there are used 36 vehicles with an average efficiency of 0.807 and a final fit of 13820.

Wednesday

In the third day of week 50,Wednesday, there are 195 different customers that need to be supply that have a total demand of 1211 pallets, so each customer has on average a demand of 6.21 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 37 vehicle to supply all the demands to the customers. If is considered that each vehicle can ”bring” a total of 35 pallets, are necessary at least 35 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 99 customers with average a demand of 12.23

If we analyse the baseline solution for Wednesday of week 50 using the first option, it is possible to notice that there are used 40 vehicles with an average efficiency of 0.876 and a final fit of 18600. If solution is analyse using the second option,there are used 50 vehicles with an average efficiency of 0.74 and a final fit of 20625.

Thursday

In the fourth day of week 50,Thursday, there are 138 different customers that need to be supply that have a total demand of 910 pallets, so each customer has on average a demand of 6.59 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 28 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, are necessary at least 26 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 76 customers with average a demand of 11.97

If we analyse the baseline solution for Thursday of week 50 using the first option, it is possible to notice that there are used 29 vehicles with an average efficiency of 0.912 and a final fit of 12905. If solution is analyse using the second option,there are used 37 vehicles with an average efficiency of 0.753 and a final fit of 13540.

Friday

In the last day of week 50,Friday, there are 148 different customers that need to be supply that have a total demand of 1017 pallets, so each customer has on average a demand of 6.87 pallets. If it is considered that each vehicle can bring a total of 33 pallets, are necessary at least 31 vehicle to supply all the demands to the customers. If is considered that each vehicle can "bring" a total of 35 pallets, are necessary at least 30 vehicles to supply all the customers where some of the pallets will not be delivered that day. If we cluster the customers we end with 91 customers with average a demand of 11.18

If we analyse the baseline solution for Friday of week 50 using the first option, it is possible to notice that there are used 38 vehicles with an average efficiency of 0.81 and a final fit of 16745. If solution is analyse using the second option,there are used 40 vehicles with an average efficiency of 0.777 and a final fit of 17405.