Site-Dependent Vehicle Routing Problem with Hard Time Windows

Gui Xavier Pereira Carvalho
gui.carvalho@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

November 2019

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,

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 [1].

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 [1], 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.

The company 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 main contribution were the design of two algorithm that are able to improve the routes made and we believe that those algorithms are also able 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. Outline

In the first section a small introduction to the problem and the objectives are introduced. In section 2 a theoretical introduction to the VRP, to the variables of the VRP needed to define the problem, and to the algorithms used to solve the problem is made. In section 3 the different characteristics of the problem are introduced followed by a mathematical formulation.In section 4 the methods used to solve the problem are explained in detail. In section 5 the baselines are presented and a comparison with the results obtain using the algorithms is done. Finally in section 6 the conclusions reached with this work are stated.

2. Literature review

.

The Vehicle Routing Problem (VRP) was first introduced in the literature by Dantzig and Ramser in 1959 [2], 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).

In the classic VRP each customer have a deterministic and know 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 problem in the field of operations research, and since the VRP is an NP-hard problem [3], 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.

2.1. 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 [4].

Solomon [5] was the first to introduced the VRPTW in the literature in 1987, and used a variation of the Clarke and Wright [6] 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 [7], and in hard constraints here the vehicle must arrive before the LPT for the solution to be feasible [8]. The mathematical formulation for the VRPTW can be found in Cordeau et al. [8].

A vast literature can be found about the VRPTW due to being "an important problem occurring in many distribution systems" [9].

2.2. Heterogeneous Fleet Vehicle Routing Problem (HFVRP)

A variation of the VRP, is the Heterogeneous Fleet Vehicle Routing Problem (HFVRP). The HFVRP have the difference that the fleet is heterogeneous this is, the fleet is composed by vehicles that are allowed to have different costs and capacities [10]. 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) [10]. The main objective of the HVRP is the minimization of the total routing cost [11].

The HFVRP is considered a "rich" VRP, more similar to real-life problem [10], HFVRP increase flexibility in distribution planning [12] and fleets in the industrial sector are rarely homogeneous [12]. There are different variations for the Heterogeneous problem, those are enunciated in Baldacci et al. [10]. 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. [11] in 1984, and the HFVRP with limited fleet, also know as Heterogeneous Vehicle Routing Problem (HVRP) introduced by Taillard [13] in 1999. The complete mathematical formulation can found in Baldacci et al. [10].

2.3. 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 [9]. Normally the exploring is done by applying k-exchange, replacing k edges by another k edges. In Bräysy and Gendreau [9] an extensive research on the Local Search, mostly apply to the VRPTW, can be found.

2.4. Genetic Algorithm (GA)

The Genetic Algorithm (GA) was developed in the $60'$s by Holland et al. [14] and is a stochastic optimization technique. The GA is based on natural evolution following Darwin's theory [15], 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 [16]. First of all to start the GA is necessary to represent each individual by a string (chromosome), which is one of the critical issues [17]. 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 parentsthe crossover can be compared to a Local Search (LS) [18] and sometimes the crossover is replaced by a LS, this hybridization is called Memetic [19]. The mutation operation is the one that normally avoids the convergence to a local minimum by maintaining the diversity [14]. 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 GA shows to be a competitive method to solve large combinatorial problems in terms of time and solution quality [16] and can be a simple and effective method [19]. The GA is also an algorithm that that is well suited for multi objective optimizations problems.

3. Problem formulation

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

3.1. Logistcis

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

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. 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 necessary if the customer does not have one of his own, that is, if a customer does not have this platform the vehicle that visits this customer must have this component.

The type of subcontracting that is going to be implemented is the Full Truck Load, where the company pays the full truck. Choosing the this hypotheses to do the design it is guaranteed that the price per pallet is going to be minimize regardless of the type of tariff chosen.

3.2. Customers

The customers that need to be supply by the company are the different stores of Worten, and not only, that are divided all over the country, the demand of the customer are 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 of Portugal, that can be obtain by its Georeference (latitude, longitude), and they are divide in 7 zones and the price of the vehicle depend on the zone that each trucks goes.

Other two things that are important to know about the customers are that a time window is defined in 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, counted 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.

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 merge 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 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. This merge have a positive point, due to the different types of customer belong to the same Group, Sonae, some of the customers are in the same place, that allows to reduce the number of customer from 713 to 491 customers, if two or more customers are in the same place it can be consider only one customer that have a demand equal to the sum of all demands.In conclusion this method allows to increase the mean occupation rate and therefore the price pay to Luís Simões decreases.

3.3. Pallets on the depot

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 allows large savings because if some pallets are left on the depot is possible to avoid paying extra vehicles. The pallets that are left on the depot are going to be store again, and are going to be shipped in the next possible delivery along with the demand of that delivery.

The way to model this strategy, is to say that the vehicles can delivery a capacity bigger than the 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 have a maximum demand of 35 pallets is feasible, where 33 pallets can be supply and two must stay at the depot.

3.4. Vehicles

The vehicles that are used to supply the demands of the customers of Worten, and not only, 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 subcontracted the vehicles that have a maximum capacity of 33 pallets . 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 (1).

$$\eta = \frac{Number\, pallets\, delivered}{Maximum\, capacity} \tag{1}$$

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 going to be used an approximation that the distance between two customer it's straight and later a correction factor is apply because the vehicles need to travel on routes and the routes are not always straight. The straight distance is going to be calculated using the geographic coordinate of each customer, and with:

$$a = sin^2(\Delta\phi/2) + cos(\phi1).cos(\phi2).sin(\Delta\lambda/2) \tag{2}$$

$$c = 2.atan2(\sqrt{a}, \sqrt{1-a}) \tag{3}$$

$$d = R.c \tag{4}$$

Where ϕ is the latitude, λ is the longitude, R is earth's radius. With the straight distance calculated is apply a factor of 1.23, this factor was find comparing the straight distance with the real distance of the routes.

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, this fee will be 50% of the vehicle value.

3.5. Mathematical Formulation

The formulation is going to be adapted from Cordeau et al. [8] where a VRPTW is present, from Baldacci et al. [10] where a HVRP is present and from Toth and Vigo [20] where a classic CVRP formulation is present.

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) $i, j \in N$ 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 will be used a fleet that is composed by m different vehicle where, $k \in V = \{1...m\}$. All Decision variables and parameters that are necessary to do 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 that is present in (5) and have the objective of minimize both the subcontracted cars, the number of pallets that are left in the depot, the number of open routes and finally the number of stop's. The smaller the combination between the those factors the better is the objective function. A cost is added for every type of car that is subcontract, for every stop made, if a route is open instead of close and finally for every pallets that should be supply and was left in the depot.

$$\begin{gathered} Min(\sum_{k=1}^{m}\sum_{j=1}^{n}X_{0j}^kC_k + \sum_{i=1}^{n}a_iC_{pal} + \\ \sum_{j=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{m}(X_{0j}^k - X_{i0}^k)O_k + \sum_{j=1}^{n}\sum_{i=0}^{n}\sum_{k=1}^{m}X_{ij}^kC_{stop}) \end{gathered} \tag{5}$$

$$\sum_{i=0}^{n}\sum_{k=1}^{m}X_{ij}^k = 1 \quad j = 1...n \tag{6}$$

$$\sum_{j=0}^{n}X_{0j}^k = 1 \quad k = ...m \tag{7}$$

$$\sum_{j=1}^{n}\sum_{i=0}^{n}l_jX_{ij}^k \leq Q_k \quad k = 1...m \tag{8}$$

$$\sum_{k=1}^{m}\sum_{i=0}^{n}Y_{ij}^k = d_j - a_j = lj \quad j = 1...n \quad Y_{ij}^k \leq Q_k \tag{9}$$

$$\sum_{j=1}^{n}\sum_{i=0}^{n}a_jX_{ij}^k \leq Pk \quad k = 1...m \tag{10}$$

$$\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{i=0}^{n}a_jX_{ij}^k \leq Pt \tag{11}$$

$$X_{ij}^k(B_i^k + t_{ij}^k - B_j^k) \leq 0 \tag{12}$$

$$e_i \leq \sum_{k=0}^{m}B_i^k \leq L_i \quad i = 0...n \tag{13}$$

$$\sum_{i=0}^{n}\sum_{j=0}^{n}X_{ij}^kt_{ij}^k \leq 540 \quad k = 1...m \tag{14}$$

$$\begin{gathered} \sum_{i=0}^{n}\sum_{j=0}^{n}X_{ij}^kt_{ij}^k + \sum_{i=0}^{n}\sum_{j=1}^{n}X_{ij}^kTf \\ + \sum_{i=0}^{n}\sum_{j=1}^{n}X_{ij}^kTvl_j \leq 780 \quad k = 1...m \end{gathered} \tag{15}$$

$$\sum_{i=0}^{n}X_{ij}^k - \sum_{i=0}^{n}X_{ij}^kD_j^k = 0 \quad k = 1...m \quad j = 1...n \tag{16}$$

$$X_{ij}^k \in \{0,1\} \tag{17}$$

Equation (6) guarantees that a costumer can be visit once and only once. In equation (7) is guarantee that the vehicles can do only one route per day. Expression (8) ensures that the capacity of the vehicle is respected. In equation (9) is guarantee that the delivery to the customer must be equal to the demand of the customer less the number of pallets that are left in the depot. Equations (10) and (11) ensures that the number of pallets in the depot is smaller than a constant. Equation (12) and (13) guarantee schedule feasibility with respect to time considerations. Equation (14) and (15) guarantees that the legislation is fulfil, the total driving time must and the total working time must be smaller that 9h and 13h respectively. In equation (16) is ensured that only specific vehicles can supply the customers. Finally in (17) X_{ij}^k must be integer.

4. Implementation

To solve the problem is used a Hybrid Genetic Algorithm (HGA), using a modify saving heuristic to initialize the algorithm. In figure (1) is possible to see the flowchart of the algorithm.

The Genetic Algorithm (GA) was used due to be able to give competitive results in terms of time and solution quality [16], and also because it can be a relative simple but effective method [19]. Once more is going to be stressed that the final algorithm should be fast and give good results due to the computation 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 [19]. Finally the GA is well suited for multiple objective optimization [17].

To do the initialization is going to be used a variation of the Clarke and Wright [6]. Ho et al. [18] 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.

4.1. Hybrid Genetic Algorithm (HGA)

The Hybrid Genetic Algorithm (HGA) is an hybridization between two algorithms, the Genetic Algorithm (GA) that was introduced in subsection (4.3), and the Local Search

(LS) that was introduced in subsection (4.4). This hybridization was made because 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. 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 the reproduction process varies between the GA reproduction process, and between an LS variation where only one iteration of the LS is done to generate a new offspring.

Since the only thing that changes in this hybridization with respect to the GA is the manner in which new offsprings are generated, we can conclude that all of the previously stated characteristics still could be considered, those characteristics are in the begging of section (4) and in the begging of subsection (4.3).

4.2. Initialization

Since it is going to be used a Genetic Algorithm, is necessary to initialize the population. Normally the population is initialize randomly [21] but can also be initialize with a mixed population, where both random and structured individuals are part of the initial solution. In the VRP literature most cases uses the second method [16], in Ho et al.[18] a comparison is done between using a random initialization and using a initialization using a build heuristic and it's concluded that the second hypothesis achieve a better performance than the randomly initialization. It is also important to refer that when the population is initialized randomly, a repair procedure is normally necessary in order to obtain feasible solution. An algorithm base on Clarke and Wright [6] was implemented due to be one of the best construction algorithms [9]. The pseudo algorithm can be seen bellow, algorithm (2).

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 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 and, 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 assign to one type of car, an adaptation of the Clarke and Wright [6] 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 [6]. The "Saving" used in this algorithm is calculated having 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 less likely those customer 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 [5]. To calculate the "Saving" is used:

$$S_{ij} = \frac{1}{C_1 + C_2},\ C_1 = d_{ij},\ C_2 = l_j - l_i,\qquad j = 1...n,\ i = 1...n. \tag{18}$$

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 costumer 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 wight 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:

-Both customer are "not assign", in this case customer j is going to be insert on the route of customer i, customer i will not be able to be chose again, and both customer pass from "not assign" to "assign", if and only if the solution is feasible;

- One of the customer is "assign" and the other is "not assign", in this case the "not assign" customer is insert in the end of the route of the "assign", the "assign" customer will not be able to be chose again, and the "not assign" pass to be an "assign" one, again if and only if the solution is feasible;

- Both customer 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, multiple 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. 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.

4.3. Genetic Algorithm

The Genetic Algorithm (GA) is a stochastic optimization technique that is used to solve big combinatorial problems. The GA was used due to getting fast and quality solutions and also due to be well suited for multiple objective optimization [17].

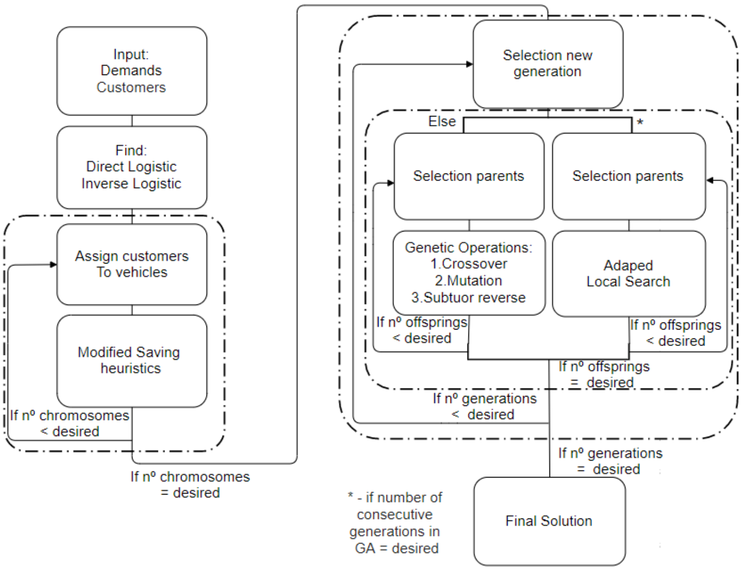

Figure 1: Caption

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

4.3.1 Gene representation

Since the GA will be used, is necessary to find a way to represent all the relevant information in the chromosome firstly, this representation is one of the critical issue when developing the GA [17]. The chromosome representation was based and similar to the chromosome of Chad et al. [22], the generic way to represent the chromosome is show bellow (3):

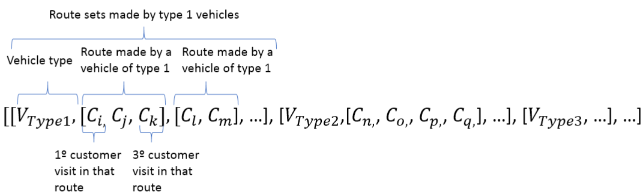

Figure 3: Generic representation of the chromosome

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 supply. So in this case the customers $C_{i..m}$ are the customers that are supply by vehicles of the type 1 were $C_{i..k}$ are supply by one vehicle and $C_{l..m}$ are supply by another vehicle of type 1. One simple example of a chromosome is represented bellow:

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

Figure 4: 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 in table (1).

With the chromosome and with a table similar to table (1) is possible to decode the chromosome. In the example of the chromosome (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 do 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. With this information is already possible to see if the set of routes are possible, using the restrictions (equation (6) until (17)) and who much it cost using the cost function (5).

4.3.2 Reproductive processes

The reproductive processes is based on generating new solutions from the chromosomes of a given population, the new chromosomes are often called offsprings and the chromosomes chosen form the population are often called parents. The offspring is going to inherit some of the characteristics of their parents [16]. From a given population some parents chromosomes are going to be chosen to generate offsrings, 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 [15], 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 objective function (5).

Table 1: Example of customers with demands

Code Store	Name Store	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

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 (4.3.6).

4.3.3 Crossover

The Crossover operation used is based on Ho et al. [18] . The steps of the operation can be see bellow:

- Select two chromosomes from the population;
- Chose at random a gene, route, of one chromosome;
- Delete the information contain in the gene on the other chromosome;
- Insert the gene at random on the receiving chromosome;
- 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.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.

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.

4.3.4 Sub-tour reverse

The sub-tour reverse that is used in this GA is based on Nazif and Lee [15], the simplify steps can be seen bellow:

- Select one chromosome from the population;
- Chose at random a gene, route, from the chromosome;
- Chose at random sub-tour of the gene;
- 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.

Due to the fact that the fitness, equation (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.

4.3.5 Mutation

The mutation operation is based in Chad et al. [22], The simplify steps can be seen bellow:

- Select one chromosome from the population;
- Chose at random a gene, route, from the chromosome;
- Chose at random a customer from the gene;
- Delete the customer in the original chromosome;
- 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.

The mutation is the operation that can lead the algorithm to leave an local minimum and improve the current solution. This occurs once more due to the fitness, equation (5), be largely influenced by the number of routes. Again to do the mutation is only necessary one parent chromosome and only one offspring is generated.

4.3.6 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 (5). The selection are based on Ho et al. [18], where a roulette wheel operation is used. The roulette wheel operation was first introduced in 1989 in Goldberg [23], 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:

$$P_{sel_i} = \frac{f_i}{\sum_{i=1}^{Npop} f_i} \tag{19}$$

Where f_i is the fitness value of the chromosome that is get from equation (5). 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 set 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. Using this selecting allows intensify the current solution due to 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 of reproduce and survive, exploring the solution space.

4.4. Local Search

The Local Search (LS) is an heuristic base 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. [22]. The steps are enumerated bellow:

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

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 HGA is initialized, where the algorithm present in section (4.2) is run the same number of times as the desired number of chromosomes and the best chromosome is chosen to initialize the LS.

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 minimize always guaranteeing that the chromosome is feasible, once more using equation (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.

5. Results

In this section the results of the theoretical models that were present 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 problems that going to be solve and the baseline solution used by the company. In the end of the section the final results given by the different methods will be compared with the baseline solution.

5.1. Problem Data

The baseline solutions is constituted by the routes that are expected to me made during weeks 6, 24 and 50. The characteristics of those weeks can be seen in table (2). The demands of the aforementioned weeks are based on forecasts of previous years and weeks, but even so those demands were heavily analyed 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.

Two different type of baseline 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 close ones just like it was explain in subsection (3.4). 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 supply. 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. Parameters

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 muta-

Table 2: 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

tion, 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 ofsprings 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.

5.3. Final results

In this section is going to be run the Hybrid Genetic Algorithm (HGA), and the the LS only 1 time for each data set, also is going to be transform the data sets in their clusters, and both algorithms will be run again 1 time only for each data set. The results of the fitness obtain are in table (3).

In table (3) 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 obtain 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 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 obtain better results than the LS, both with and without cluster, this happen because the HGA have more flexibility than the simple LS, if the LS get stuck in a local minimal he 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.

6. 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 achieve 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 a shorter computational time than the hybrid algorithm. Transforming the customers into their clusters allowed improving the overall results from an improvement of almost 10% to an improvement of over 20%, although this transformation increase the number of pallets left in the depot.

The used of the Local Search in the Genetic Algorithm, the Hybrid Genetic Algorithm, greatly improved the solution due to the LS having into account the objective function and not purely the randomness, but also requires a longer computational time. The hybrid algorithm implemented gave the best results because since 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. We believe 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 several parameter calibrations which requires a long preprocessing before the algorithm is ready to use.

Table 3: Fitness of the final results

		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

References

[1] Arild Hoff, Henrik Andersson, Marielle Christiansen, Geir Hasle, and Arne Løkketangen. Industrial aspects and literature survey: Fleet composition and routing. *Computers & Operations Research*, 37(12):2041–2061, 2010.

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

[3] Jan Karel Lenstra and AHG Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

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

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

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

[7] Éric Taillard, Philippe Badeau, Michel Gendreau, François Guertin, and Jean-Yves Potvin. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, 31(2):170–186, 1997.

[8] Jean-François Cordeau, Guy Desaulniers, Jacques Desrosiers, Marius M Solomon, and François Soumis. The vehicle routing problem, chapter vrp with time windows. *SIAM, Society for Industrial and Applied Mathematics, Philadelphia, USA*, pages 157–193, 2000.

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

[10] Roberto Baldacci, Maria Battarra, and Daniele Vigo. Routing a heterogeneous fleet of vehicles. In *The vehicle routing problem: latest advances and new challenges*, pages 3–27. Springer, 2008.

[11] Bruce Golden, Arjang Assad, Larry Levy, and Filip Gheysens. The fleet size and mix vehicle routing problem. *Computers & Operations Research*, 11(1):49–66, 1984.

[12] Puca Huachi Vaz Penna, Anand Subramanian, and Luiz Satoru Ochi. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232, 2013.

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

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

[15] Habibeh Nazif and Lai Soon Lee. Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, 36(5):2110–2117, 2012.

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

[17] SP Anbuudayasankar, Koh Ganesh, SC Lenny Koh, and Yves 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.

[18] William Ho, George TS Ho, Ping Ji, and Henry CW Lau. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4):548–557, 2008.

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

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

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

[22] Padmabati Chand, Bhabani Sankar Prasad Mishra, and Satchidananda Dehuri. A multi objective genetic algorithm for solving vehicle routing problem. *International Journal of Information Technology and Knowledge Management*, 2(2):503–506, 2010.

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