DEVELOPMENT OF A TOOL FOR OPTIMIZING VEHICLE ROUTING PROBLEM

HAMANI AMIR

1985

جامعة محمد بوضياف - المسيلة

Université Mohamed Boudiaf - M'sila

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Master

Computer science department
Faculty of Computer science and Mathematics

Supervised by Dr. Lounnas Bilal

2018

Family means nobody gets left behind, or forgotten.

Dedicated to the loving memory of my brother **Hamani Souleyman**.

1988 – 2008

ACKNOWLEDGEMENTS

In The Name of **ALLAH**, The Most Beneficent, The Most Merciful.

All praise belongs to **ALLAH** alone, and blessings and peace be upon the final Prophet.

I am grateful to my parents, who supported me emotionally and financially. I always knew that you believed in me and wanted the best for me. Thank you for teaching me that my job in life was to learn, to be happy, and to know and understand myself; only then could I know and understand others. Thanks from the bottom of my heart dad and mom.

Thanks to my little brother **Sami** who is been there always for me.

Thanks to my fiancee **Batoul Hadj Maati** for her encouragement and support during many hard times, and many thanks to all her family members.

Many thanks to my supervisor **Dr. Lounnas Bilal** for his advices and support.

CONTENTS

LIST OF FIGURES

ACRONYMS

VRP	Vehicle Routing Problem
TSP	Travelling Salesman Problem
JSP	Job Shop Scheduling Problem
SA	Simulated Annealing
TS	Taboo Search
CVRP	Capacitated Vehicle Routing Problem
MDVRP	Multiple Depot Vehicle Routing Problem
PVRP	Periodic Vehicle Routing Problem
SDVRP	Split Delivery Vehile Routing Problem
SVRP	Stochastic Vehicle Routing Problem
VRPB	Vehicle Routing Problem With Backhauls
VRPPD	Vehicle Routing Problem With Pick-up And Delivering
VRPTW	Vehicle Routing Problem With Time Windows
STSP	Symmetric Travelling Salesman Problem
ATSP	Asymmetric Travelling Salesman Problem
MTSP	Multi Travelling Salesman Problem

Part I

DISSERTATION SUMMARY

This part has one chapter of general introduction, the aim of this chapter is to provide a preface of our research interest, our motivation for working on this kind of research issues

GENERAL INTRODUCTION

Combinatorial optimization occupies a very important place in research operational, in discrete mathematics and computer science. Its importance is justified on the one hand by the great difficulty of optimization problems and on the other hand by many practical applications that can be formulated in the form of a combinatorial optimization problem.
A combinatorial optimization problem is to browse the search space (the set of combinations that can be taken by the variables of the problem) in order to extract an optimal solution from a finite set of solutions, often of a very large size such that its exhaustive enumeration is a tedious task. The resolution of a combinatorial optimization problem requires the use of an algorithmic method allowing the maximization or the minimization of one or more objective functions while respecting the constraints posed by the problem.
The vehicle routing problem (VRP) is overwhelmingly described as the problem in which vehicles based on a central depot are required to visit a set of geographically scattered customers in order to fulfill recognized customer demands. The objective is to construct a minimum cost, feasible set of paths one for each vehicle. A path is a series of locations that a vehicle must visit along with the indication of the serve it provides. The vehicle must start and finish its cycle at the depot.
The motivation behind our study of the vehicle routing problem (VRP) came as a result of the multiplying applications in the real world to these problems represent. For this reason, the VRP is a very active field of research and has witnessed an astonishing interaction amidst theory and practice.
VRP is both of theoretical and practical interest (due to its real world applications), which explains the amount of attention given to the VRP by researchers in the past years, and since VRP is an NP-Hard problems as will be ilustrated in the coming chapters.
Our dissertation is divided into two parts, the first one contain a litterateur topics to help the reader understand the contribution of this thesis. The First part contains two chapters, the first chapter is combinatorial optimization, we will discuss the techniques and methods of resolution,which will lead us to two main classes of methods: the exact methods, which generally consist in enumerating, implicitly, all the solutions of the search space and guaranteeing an ideal solution, and the approximated methods which

generally deal with the problems of large size, they do not ensure to find the ideal solution but are effective.
In the second chapter the Vehicle Routing Problem (VRP), we will introduced the VRP by elapsing from the definition to the history to the variants of the VRP and by discussing the origins of it, and some related works of our problem.
The next part is where we talk about our contributions and it contains only one chapter, which we will provide in it a modelization of tool for solving and optimizing the vehicle routing problem.
We conclude our dissertation by a general conclusion and future works.

Part II

LITTERATURE

Two chapters will be provided in this part to help the reader to understand our work. The chapters are Combinatorial Optimization, and Vehicle Routing Problem (VRP). Both of theme will contains definitions, state of the art, methods,...ext.

2 COMBINATORIAL OPTIMIZATION

2.1 INTODUCTION

Combinatorial optimization is an indispensable tool combining various techniques ofdiscrete mathematics and computer science to solve optimization problemscombinatorial of real life. A combinatorial optimization problem is to find the best solution in a discrete set of solutions called together feasible solutions. In general, this together is finite but of very great cardinality . In general, this involves maximizing (maximizing problem) or minimizing (minimization problem) an objective function under certain constraints. The goal is to find an optimal solution in a reasonable execution time[13].

In this chapter we present some of the most representative methods, developed to solve combinatorial optimization problems.

2.2 DEFINITION

Combinatorial optimization is to minimize or maximize a function often called cost function, of one or more variables subject to constraints. Optimization combinatorics occupies a very important place in operations research, in mathematics discrete and in computer science. Its importance is justified on the one hand by the great difficulty of the optimization problems and secondly by many practical applications that can be formulated as a combinatorial optimization problem . even though combinatorial optimization problems are often easy to define, they are usually difficult to solve. Indeed, most of these problems belong to the class of NP-hard problems and so do not currently have an effective algorithmic solution valid for all data [13]. Now we'll see the problem formulation :

$$\begin{gathered} \text{Min } f(x) \\ \text{Subject to } g(x) \leqslant 0 \\ x \in R^n \end{gathered}$$

$x \in R^n \longrightarrow n$ variables, $i = 1, \ldots, n$.

$g : R^n \longrightarrow R^m \longrightarrow m$ constraints, $j = 1, \ldots, m$.

$f : R^n \longrightarrow R^p \longrightarrow$ objective functions, $k = 1, \ldots, p$.

2.2.1 *Resolution of a combinatorial optimization problem*

Solving a combinatorial optimization problem requires the study of three points particulars:

1. The expression of the objective to be optimized.
2. Definition of the set of feasible solutions.
3. The choice of the optimization method to use.

The first two points are problem modeling, the third of its resolution. In order to define the set of feasible solutions, it is necessary to express all the constraints of the problem. This can only be done with good knowledge of the problem under study and its field of application[13].

2.3 COMPLEXITY

The complexity of any problem implies the estimation, in the worst case, of the number of instructions to perform in order to solve an instance of the problem. Algorithmic complexity is used to measure the effectiveness of an algorithm for solving a problem. Indeed, the more the number of instructions is important the more the problem is complex and therefore the algorithmic complexity is proportional to the number of instructions of the problem. We deduce that the more complex the algorithm, the less efficient it is.

Several types of complexity can be distinguished, one cites the constant complexity o(1) which is independent of the size of the problem, the logarithmic complexity o(log (n)), the linear complexity o(n), the quadratic complexity $o(n^2)$, the cubic complexity $o(n^3)$, the polynomial complexity $o(n^p)$, the exponential complexity $o(exp^n)$, and finally the factorial complexity o(n!). According to the criterion of complexity, we distinguish essentially four classes of problems:

- Class P :

  In this class we find all the problems that can be solved, in polynomial time, by a Turing machine. The resolution of any instance of the problems of this class is done in a polynomial time and space hence the class name of polynomial problems or the Class P [29].

- Class P-Complete :

  In this class we find the set of polynomial problems and the set of non-polynomial problems in which any polynomial problem can be reduced to a poly-logarithmic time with or without the use of a deterministic calculating machine [29].

- NP Class :

  This class encompasses the set of problems that can be solved in polynomial time on a Turing machine [29].

- NP-Hard class :

  This class is also called NP-Hard. It brings together all the problems that the complexity is exponential or doubly exponential. This class of problem is distinguished by the difficulty of optimal or exact resolution of large instances, in a polynomial time. In this regard, we are content to find solutions known as good quality in a polynomial time [29].

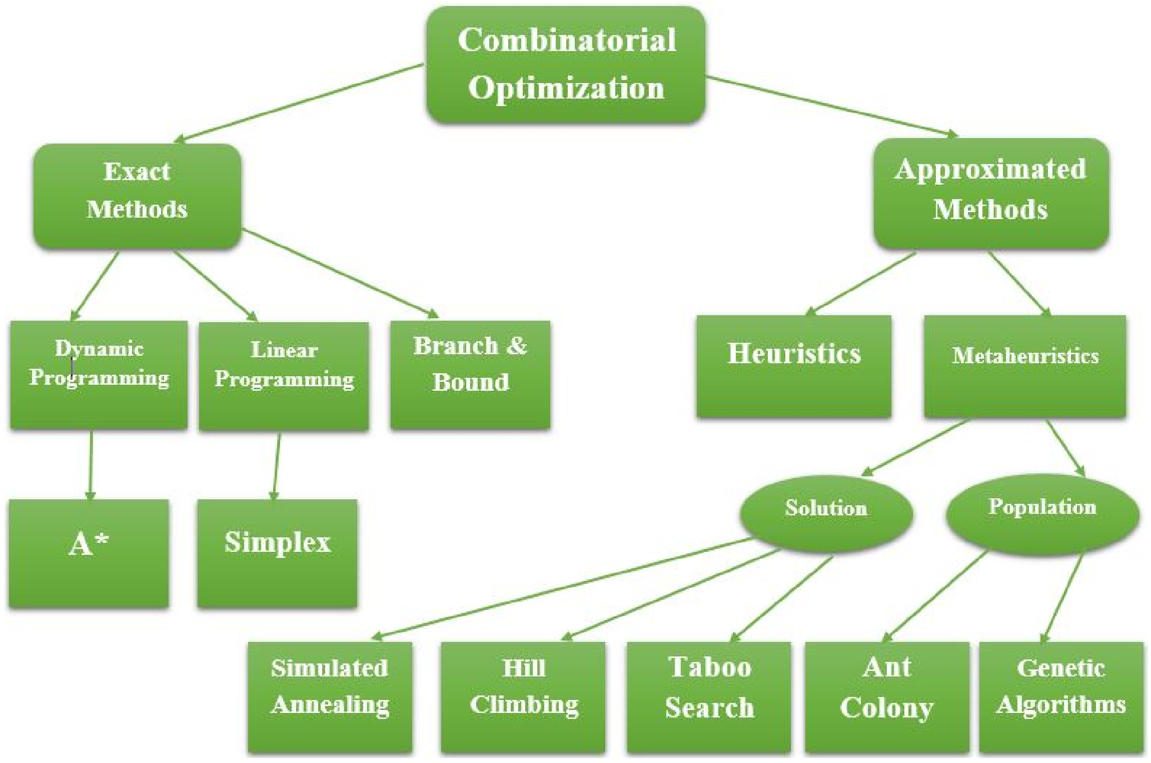
2.4 METHODS OF COMBINATORIAL OPTIMIZATION

Optimization methods can be divided into two broad classes of methods for problem solving:

- The exact methods.
- The approximate methods.

As shown in the Figure 1 Below :

Figure 1: Classification Of Combinatorial Optimization Methods



2.4.1 *The exact methods*

The exact methods rely on the use of algorithms that finds at least one solution for a problem.

As we saw in the Figure 1 we can list the following methods :

2.4.1.1 *Branch and Bound*

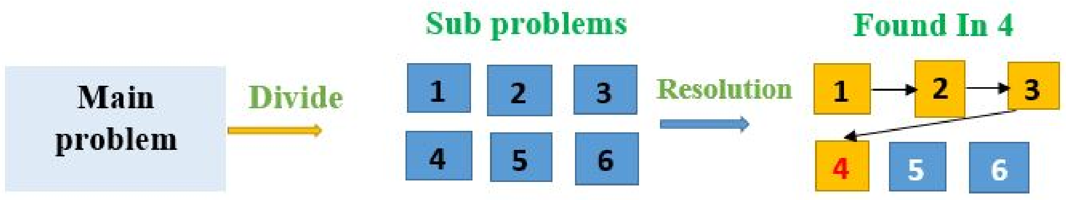
The branch and bound method consists in enumerating these solutions in an intelligent way in that, by using certain properties of the problem in question, this technique succeeds in eliminating partial solutions which do not lead to to the solution we are looking for. As a result, the desired solution is often achieved in reasonable time. Of course, in the worst case, we always fall back on the explicit elimination of all the solutions of the problem.

To do this, this method has a function that allows to put a terminal on some solutions to either exclude or maintain them as potential solutions. Of course, the performance of a method of branch and bound depends, among others , the quality of this function (its ability to exclude early partial solutions)[13].

2.4.1.2 *The dynamic programming*

The term dynamic programming was first used in 1940 by Richard Bellman to describe problems where one needs to find the best decisions one after another. The main concept of dynamic programming is straight-forward as we'll see in the next Figure 2. We divide a problem into smaller nested subproblems, and then combine the solutions to reach an overall solution. This concept is known as the principle of optimality[24].

Figure 2: The Main Concept Of Dynamic Programming



2.4.1.3 *Other exact methods*

Apart from the branch and bound and dynamic programming methods described above, there are many different exact approaches that exist, and use the specificities of the problem addressed to solve the optimization problem. This is the case of the simplex algorithm[38].

In another hand, there is other exact methods that are specific to the problem under study, such as Johnson's algorithm for scheduling [16].

2.4.2 *The approximate methods*

Approximate resolution methods are generally used where exact methods fail. Indeed, an exact resolution requires traversing the entire search space, which becomes unworkable when one wants to solve big problems. In this case, a partial execution of the exact algorithm rarely makes it possible to obtain a good quality solution. Approximate resolution methods have been developed to quickly provide good quality but not optimal solutions.

2.4.2.1 *The heuristics*

As opposed to exact methods, which guarantee to give an optimum solution of the problem, heuristic methods only attempt to yield a good, but not necessarily optimum solution. Nevertheless, the time taken by an exact method to find an optimum solution to a difficult problem, if indeed such a method exists, is in a much greater order of magnitude than the heuristic one (sometimes taking so long that in many cases it is inapplicable). Thus we often resort to heuristic methods to solve real optimization problems[23].
In addition to the need to find good solutions of difficult problems in reasonable time, there are other reasons for using heuristic methods, among which we want to highlight:

1. No method for solving the problem to optimality is known.
2. Although there is an exact method to solve the problem, it cannot be used on the available hardware.
3. The heuristic method is more flexible than the exact method, allowing, for example, the incorporation of conditions that are difficult to model.
4. The heuristic method is used as part of a global procedure that guarantees to find the optimum solution of a problem.

A good heuristic algorithm should fulfil the following properties :

- A solution can be obtained with reasonable computational effort.
- The solution should be near optimal (with high probability).
- The likelihood for obtaining a bad solution (far from optimal) should be low.

2.4.2.2 *Metaheuristics*

Fred Glover and al [19]. Define the metaheuristics, in their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space. Over time, these methods have also come to include any procedures that employ strategies for overcoming the trap of local optimality in complex solution spaces, especially those procedures that utilize one or more neighborhood structures as a means of defining admissible moves to transition from one solution to another, or to build or destroy solutions in constructive and destructive processes. The degree to which neighborhoods are exploited varies according to the type of procedure. In the case of certain population-based procedures, such as genetic al- rithms, neighborhoods are implicitly (and somewhat restrictively) defined by reference to replacing components of one solution with those of another, by variously chosen rules of exchange popularly given the name of crossover. In other population-based methods, based on the notion of path relinking, neighborhood structures are used in their full generality, including constructive and destructive neighborhoods as well as those for transitioning between (complete) solutions. Certain hybrids of classical evolutionary approaches, which link them with local search, also use neighborhood structures more fully, though apart from the combination process itself.

Metaheuristics which are subdivided into two subclasses :

- Neighborhood methods.
- The evolutionary methods.

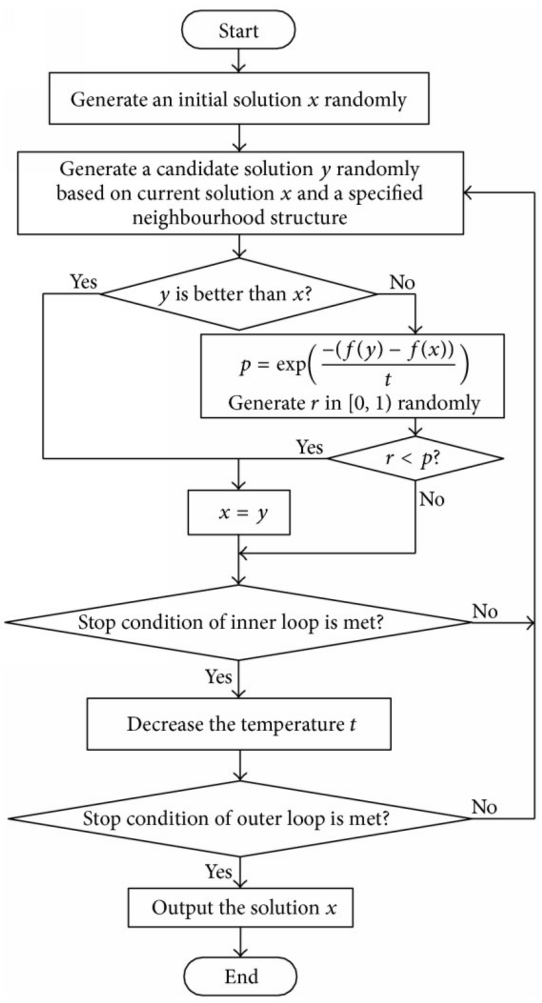
1. **Neighborhood methods :** These methods start from an initial solution (obtained exactly, or by drawing random) and gradually move away from it, to achieve a trajectory, a path progressive in the space of solutions [13].In this category, we rank:

   - **Simulated Annealing**.
   - **The Tabou method** : the local search term is more and more used for qualify these methods.

   a) **Simulated Annealing** :
   Simulated annealing (SA) was introduced by (Kirkpatrik et al., 1983)[41] and (Cerny 1985)[6] as a normal local search method, using a strategy to avoid local minima. This metaheuristic is based on a technique used for a long time by metallurgists who, to obtain a faultless alloy, alternating cycles of reheating (or annealing) and slow cooling of metals.
   Simulated annealing is based on work done by (Metropolis et al., 1953)[25], who have been able to describe the evolution of a thermodynamic system.
   The main idea of simulated annealing as proposed by Metropolis in 1953 is simulate the behavior of the material in the annealing process very widely used in metallurgy. The goal is to reach a state of thermodynamic equilibrium, this state of equilibrium (where the energy is minimal) represents - in the simulated annealing method - the solution optimal of a problem; The energy of the system will be calculated by a cost function (or objective function). The method will try to find the optimal solution by optimizing an objective function, for this, a fictitious temperature parameter has been added by Kirkpatrick, Gelatt and Vecchi. Basically the

principle is to generate successively configurations from an initial solution So and an initial temperature To which will decrease throughout the process until reaching a final temperature or steady state (global optimum)[13].
The Figure 3 will explain the main idea.

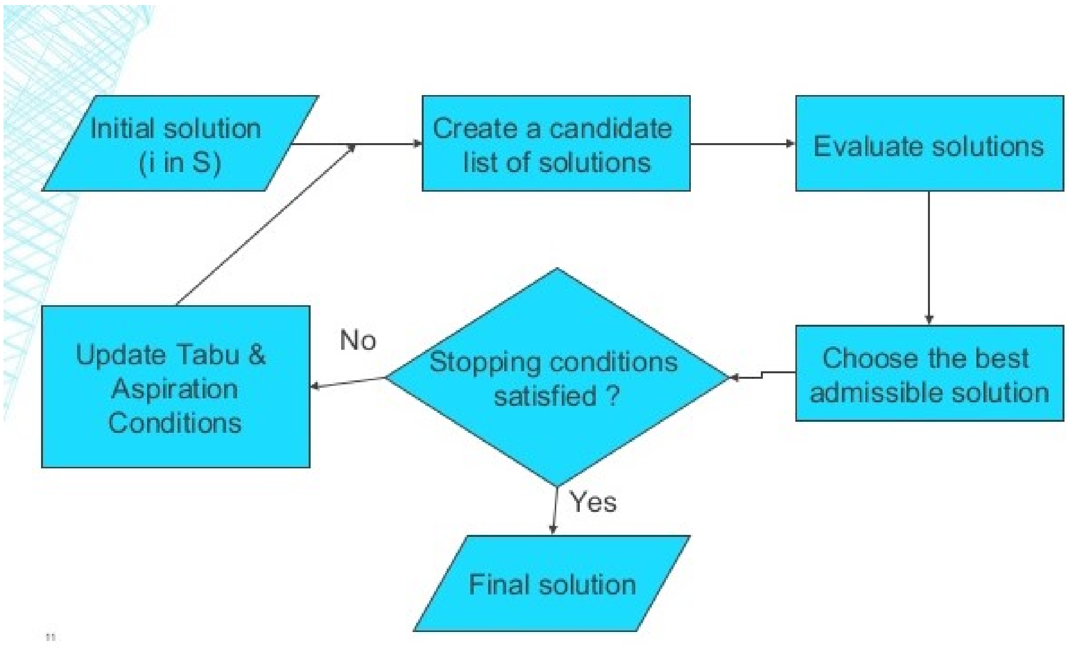
Figure 3: Flowchart Of Simulated Annealing Algorithm [36]


Start
Generate an initial solution x randomly
Generate a candidate solution y randomly based on current solution x and a specified neighbourhood structure
y is better than x?
Yes
No
$p = \exp\left(\frac{-(f(y) - f(x))}{t}\right)$
Generate r in $[0, 1)$ randomly
$r < p$?
Yes
No
$x = y$
Stop condition of inner loop is met?
No
Yes
Decrease the temperature t
Stop condition of outer loop is met?
No
Yes
Output the solution x
End

b) **Taboo Search** :
Building upon some of his previous work, Fred Glover proposed a new approach, which he called tabu search, to allow local search methods to overcome local optima . In fact, many elements of this first TS proposal and some elements of later TS elaborations were introduced in , including short-term memory to prevent the reversal of recent moves, and longer term frequency memory to reinforce attractive components. The basic principle of TS is to pursue LS whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously vis-

ited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search, a key idea that can be linked to artificial intelligence concepts. It is also important to remark that Glover did not see TS as a proper heuristic, but rather as a metaheuristic, i.e., a general strategy for guiding and controlling inner heuristics specifically tailored to the problems at hand[21].
The next Figure 4 will explain the process of the Taboo Search algoithm.

Figure 4: Flowchart Of Taboo Search [32]


Initial solution
(i in S)
Create a candidate list of solutions
Evaluate solutions
Update Tabu & Aspiration Conditions
No
Stopping conditions satisfied ?
Choose the best admissible solution
Yes
Final solution

Taboo search enhances the performance of local search by relaxing its basic rule. First, at each step worsening moves can be accepted if no improving move is available (like when the search is stuck at a strict local minimum). In addition, prohibitions (henceforth the term taboo) are introduced to discourage the search from coming back to previously-visited solutions.

2. **The evolutionary methods :** These methods are distinguished from those already studied by the fact that they operate on a population of solutions, for this they are often called population-based methods. Some of them have principles inspired by the genetics and behavior of insects. The complexity of these two biological phenomena has served as a model for ever more sophisticated algorithms over the last twenty years [13]. We have retained only by genetic algorithms.

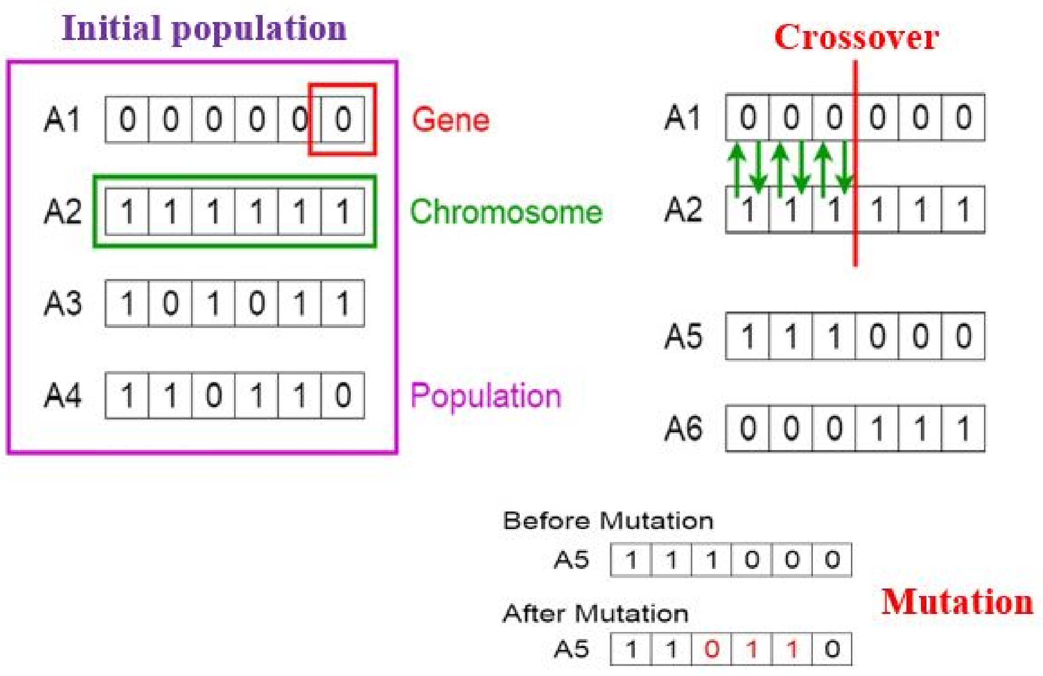
2.1.**Genetic Algorithms :**

Genetic algorithms are a type of optimization algorithm, meaning they are used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function. Genetic algorithms represent one branch of the field of study called evolutionary computation , in that they imitate the biological processes of reproduction and natural selection to solve for the 'fittest' solutions . Like in evolution, many of a genetic algorithm's processes are random, however this optimization technique allows one to set the level of randomization and the level of control .

These algorithms are far more powerful and efficient than random search and exhaustive search algorithms , yet require no extra information about the given problem. This feature allows them to find solutions to problems that other optimization methods cannot handle due to a lack of continuity, derivatives, linearity, or other features[5].

The Figure 5 will show us what's a genetic algorithm

Figure 5: Genetic Algorithm [22]



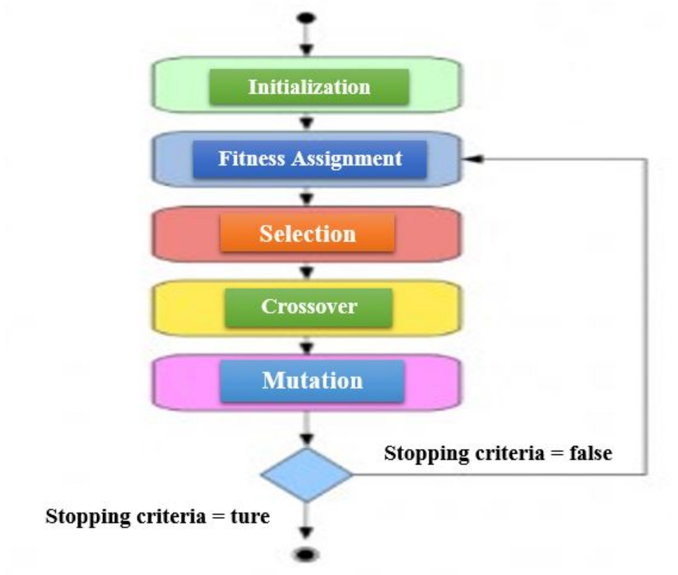
2.1.1.**Notion of Natural Selection :**

The process of natural selection starts with the selection of fittest individuals from a population. They produce offspring which inherit the characteristics of the parents and will be added to the next generation. If parents have better fitness, their offspring will be better than parents and have a better chance at surviving. This process keeps on iterating and at the end, a generation with the fittest individuals will be found.
This notion can be applied for a search problem. We consider a set of solutions for a problem and select the set of best ones out of them[22].
Here we consider that the genetic algorithms have five (05) phases which are :

- Initial population.
- Fitness function.
- Selection.
- Crossover.
- Mutation.

The working of a genetic algorithm is also derived from biology, which is as shown in the Figure 6.

Figure 6: The Working Of A Genetic Algorithm [22]

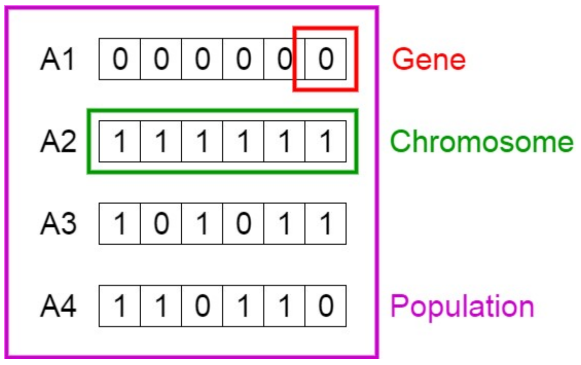


So, let us try to understand the steps one by one.

a) **Initial Population :**

The process begins with a set of individuals which is called a **Population**. Each individual is a solution to the problem you want to solve.
An individual is characterized by a set of parameters (variables) known as **Genes**. Genes are joined into a string to form a **Chromosome** (solution).
In a genetic algorithm, the set of genes of an individual is represented using a string, in terms of an alphabet. Usually, binary values are used (string of 1s and 0s). We say that we encode the genes in a chromosome[22]. As it shown in the Figure 7 below.

Figure 7: Population, Chromosomes And Genes [22]



b) **Fitness Function :**

The fitness function determines how fit an individual is (the ability of an individual to compete with other individuals). It gives a **fitness score** to each individual. The probability that an individual will be selected for reproduction is based on its fitness score[22].

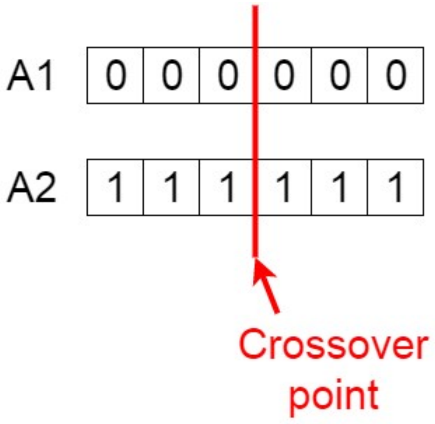
c) **Selection :**

The idea of selection phase is to select the fittest individuals and let them pass their genes to the next generation.
Two pairs of individuals **(parents)** are selected based on their fitness scores. Individuals with high fitness have more chance to be selected for reproduction[22].

d) **Crossover :**

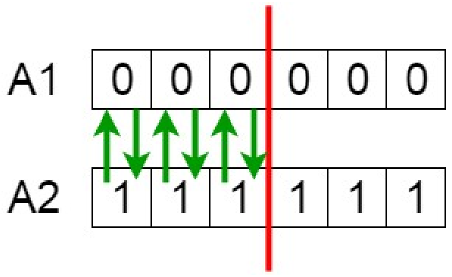
Crossover is the most significant phase in a genetic algorithm. For each pair of parents to be mated, a **crossover point** is chosen at random from within the genes[22]. For example, consider the crossover point to be 3 as shown below in Figure 8.

Figure 8: Crossover Point [22]



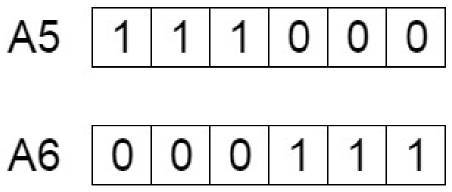
Individual Chromosomes are created by exchanging the genes of parents among themselves until the crossover point is reached.

Figure 9: Exchanging Genes Among Parents [22]



The new individual chromosomes are added to the population.

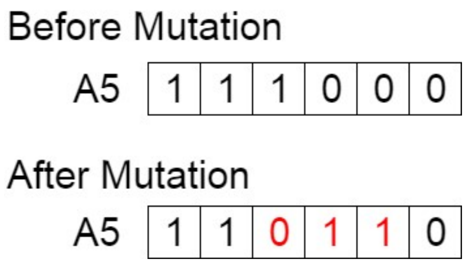
Figure 10: New Individual Chromosomes [22]



e) **Mutation :**

In certain new offspring formed, some of their genes can be subjected to a mutation with a low random probability. This implies that some of the bits in the bit string can be flipped[22].As it shown below in Figure 11.

Figure 11: Before And After Mutation [22]



Mutation occurs to maintain diversity within the population and prevent premature convergence.

f) **Termination :**

The algorithm terminates if the population has converged (does not produce offspring which are significantly different from the previous generation). Then it is said that the genetic algorithm has provided a set of solutions to our problem[22].

g) **Comments :**

The population has a fixed size. As new generations are formed, individuals with least fitness die, providing space for new offspring.

The sequence of phases is repeated to produce individuals in each new generation which are better than the previous generation[22].

2.5 CONCLUSION

In this chapter we have presented the combinatorial optimization and the classification of methods of resolution .several methods of heuristic resolution and metaheuristics. It is necessary to use heuristics to find good approximate solutions. So to solve a problem one must choose the appropriate methods that can be adapted to the type of the problem. Besides our main objective from this chapter is to present the techniques of resolution of the vehicle routing problem . In the next chapter we'll present our main problem which is the vehicle routing problem.

3 VEHICLE ROUTING PROBLEM

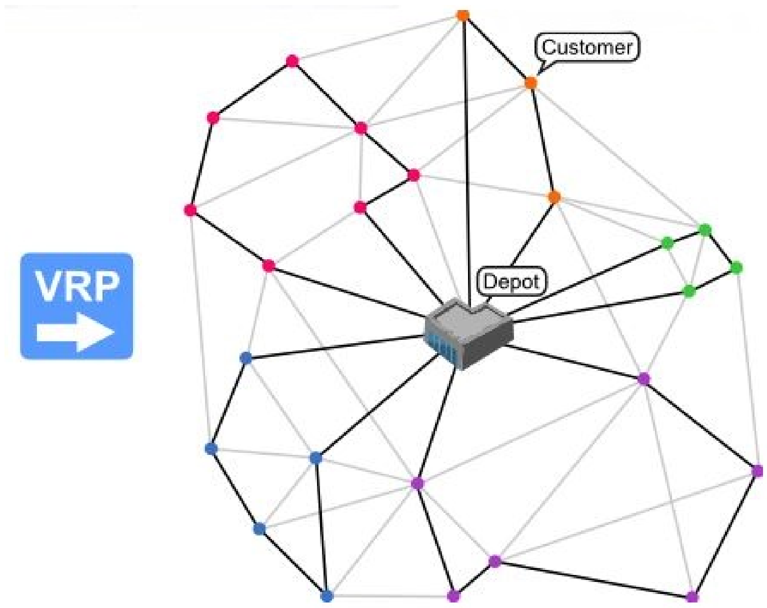
3.1 INTRODUCTION

Vehicle Routing Problem can be defined as the problem of designing routes for delivery vehicles (of known capacities) which are to operate from a single depot to supply a set of customers with known locations and known demands for a certain commodity. Routes for the vehicles are designed to minimize some objective such as the total distance travelled or total cost. The VRP is NP Hard problem , meaning that the computational effort required solving this problem increases exponentially with the problem size[35].
Vehicle Routing Problems have very important applications in the area of distribution management. VRP is both of theoretical and practical interest (due to its real world applications), which explains the amount of attention given to the VRP by researchers in the past years, and since VRP is an NP-Hard problems [14].
we will try to make a study of this problem in general, in order to adapt the methods of resolution to this problem in the coming chapters.

3.2 DEFINITION

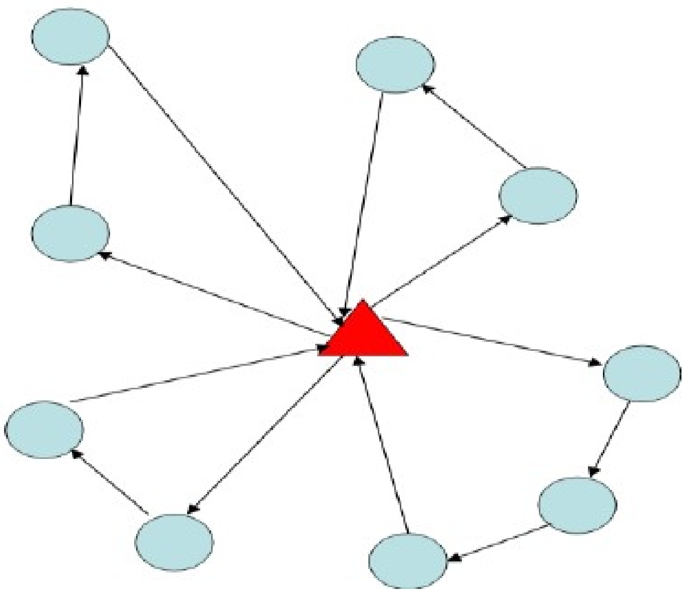
The Vehicle Routing Problem (VRP) is a generic name given to a set of problems in which set of routes for a fleet of vehicles based at one or several depots are to be formed for servicing the customers dispersed geographically. In the Vehicle Routing Problem (VRP), multiple vehicles leave from a single location a ('depot') and must return to that location after completing their assigned tours. The objective of the VRP is to form a route with lowest cost to serve all customers. Generally, distribution or collection of goods from customers to depot is called as VRP or Vehicle Scheduling Problem. In particular, the solution of a VRP calls for the determination of a set of routes, each performed by a single vehicle that starts and ends at its own depot, such that all the requirements of the customers are fulfilled, with some operational constraints and the global transportation cost is minimized. The operational constraints can be a vehicle capacity, route length, precedence relation between customers, etc[35]. we'll see the next Figure 12:

Figure 12: Solution Of A VRP [28]



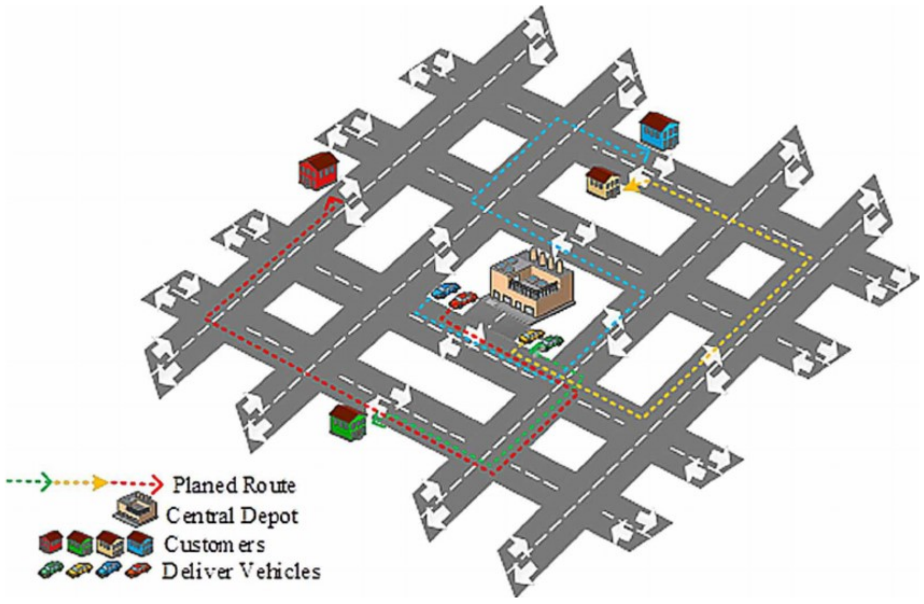
Let 's have a look to another Figure 13 the vehicle routing problem :

Figure 13: Vehicle Routing Problem [15]



Here's another Figure 14 about the VRP for the seek of a good explanation :

Figure 14: The Traditional VRP [18]



3.2.1 *History*

The vehicle routing problem (VRP) consists of designing least cost delivery routes through a set of geographically scattered customers, subject to a number of side constraints. This problem holds a central place in distribution management and is faced on a daily basis by tens of thousands of carriers worldwide.
The problem arises in several forms because of the variety of constraints encountered in practice. For over 50 years, the VRP has attracted the attention of a large part of the operational research community. This is due partly to the economic importance of the problem, but also to the methodological challenges it poses. For example, the traveling salesman problem (TSP), which is a special case of the VRP, can now be solved for thousands and even tens of thousands of vertices. In contrast, the VRP is much more difficult to solve [42].

3.2.2 *Types of VRP*

There are a lot of VRP types which are the following :

- **Capacitated VRP** : CVRP is a VRP in which a fixed fleet of delivery vehicles of uniform capacity must service known customer demands for a single commodity from a common depot at minimum transit cost. That is, CVRP is like VRP with the additional constraint that every vehicles must have uniform capacity of a single commodity [26].

- **Multiple Depot VRP** : In a MDVRP, the number and locations of the depots are predetermined. Each depot is large enough to store all the products ordered by the customers. Each vehicle starts and finishes at the same depot. The location and demand of each customer is also known in advance and each customer is visited by a vehicle exactly once [8].

- **Periodic VRP** : Planning of the routes occurs over a time period of several days (time horizon) in which customers are visited with different frequencies. For instance, supermarkets might request service every day, while for a small butcher one collection a week suffices. The resulting problem is a variant of the Periodic Vehicle Routing Problem (PVRP). As in the traditional Vehicle Routing Problem (VRP), customer locations each with a certain demand function are given, as well as a set of capacitated vehicles. In addition, the PVRP has a horizon, say T days, and there is a frequency for each customer stating how often within this T-day period this customer must be visited. A solution to the PVRP consists of T sets of routes that jointly satisfy the demand constraints and the frequency constraints. The objective is to minimize the sum of the costs of all routes over the planning horizon. Obviously, this problem is at least as hard as the VRP [34].

- **Split Delivery VRP** : In the Split Delivery Vehicle Routing Problem (SDVRP) a fleet of capacitated homogeneous vehicles is available to serve a set of customers. Each customer can be visited more than once, contrary to what is usually assumed in the classical Vehicle Routing Problem (VRP), and the demand of each customer may be greater than the vehicle capacity. Each vehicle has to start and end its tour at the same depot. The problem consists in finding a set of vehicle routes that serve all the customers such that the sum of the quantities delivered in each tour does not exceed the capacity of a vehicle and the total distance traveled is minimized [2].

- **Stochastic VRP** : SVRP is also a generic name given to VRP that considers one or more parameters as stochastic or unknown during the planning horizon. The SVRP refers to a family of problems, that combine the characteristics of stochastic and integer programs, and are often regarded as computationally intractable .The random parameters may be the presence of customers, the nature of customer demand at a given location, the time such as service time, or travel time or windows time . Most common variants of SVRP are VRP with stochastic customer demands. SVRP also addresses other areas such as unknown knowledge of the road conditions, passengers shift and working hours, heterogeneous vehicles, split customer demand, number and size of vehicle, and others [10].

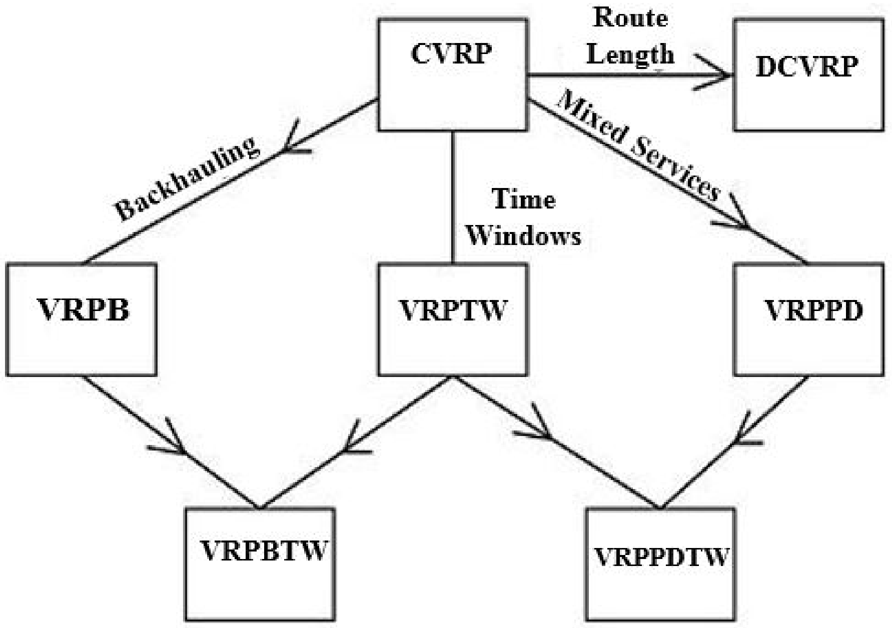
- **VRP with Backhauls** : The VRP with Backhauls (VRPB) [11], which considers delivery and collection points. Linehaul customers are sites which have a demand of goods, thus a delivery has to be made from the depot. Backhaul customers are points where a quantity of goods has to be collected and taken to the depot. A practical example of this customer partition is that of the grocery industry, where supermarkets and shops are the linehaul customers and grocery suppliers are the backhaul customers.

- **VRP with Pick-Up and Delivering** : The Vehicle Routing Problem with Pick-up and Delivering (VRPPD) is a VRP in which the possibility that customers return some commodities is contemplated. So in VRPPD itâs needed to take into account that the goods that customers return to the deliver vehicle must fit into it. This restriction make the planning problem more difficult and can lead to bad utilization of the vehicles capacities, increased travel distances or a need for more vehicles.
  Hence, it is usually to consider restricted situations where all delivery demands start from the depot and all pick-up demands shall be brought back to the depot, so there are no interchanges of goods between the customers. Another alternative is relaxing the restriction that all customers have to be visited exactly once. Another usual simplification is to consider that every vehicle must deliver all the commodities before picking up any goods [27].

- **VRP with Time Windows** : The vehicle routing problem with time windows (VRPTW), which is a generalization of the VRP where the service at any customer starts within a given time interval, called a time window. Time windows are called soft when they can be considered nonbiding for a penalty cost. They are hard when they cannot be violated, i.e., if a vehicle arrives too early at a customer, it must wait until the time window opens; and it is not allowed to arrive late [37].

Here, in this next Figure 15, we can see the variants of the vehicle routing problem :

Figure 15: Variants Of VRP [44]



3.3 FORMULATION OF THE PROBLEM

In this section we define the problem under study [4]:

Customers:

The problem is given by a set of customers $N = 1, 2, ..., n$, residing at n different locations. Every pair of locations (i,j), where i,j $\in$ N, and $i \neq j$, is, associated with a travel time t_{ij} and a distance traveled d_{ij} that are symmetrical $(t_{ij} = t_{ji} and d_{ij} = d_{ji})$.Denote by q_i , $i = 1, 2, ..., n$; the demand at point i . The central depot is denoted by 0.

Fleet of vehicles:

The customers are served from one depot with a homogeneous and limited fleet. The vehicles leave and return to the depot. There is a set V of vehicles , $V = 1, ., m$ with different capacities. The capacity of each vehicle $k \in V$ is represented by a_k.

We let $R_i = (r_i(1), .., r_i(n_i))$, denote the route for vehicle i ,where $r_i(j)$ is the index of the jth customer visited and n_i is the number of customers in the route.We assume that every, route finishes at the depot,i.e $r_i(n_i + 1) = 0$.

Time windows :

Each customer $i \in N$, has a time windows,i.e. an interval$[e_i, l_i]$, where $e_i \leqslant l_i$,which corresponds respectively, to the earliest and latest time to start to service customer i. Let s_i be the service time at customer i.

Split deliveries :

The demand of a customer may be fulfilled by more than one vehicle. This occurs in all cases where some demand exceeds the vehicle capacity, but can also turn out to be cost effective in other cases. The decision variables of the model are:

${x_{ij}}^k = 1$,if j is supplied after i by vehicle. 0 , Otherwise.

${x_i}^k$= moment at which service begins at customer i by vehicle k, i=1,..,n ; k=1,..,m.

${y_i}^k$=fraction of customer demand i delivered by vehicle k.

The objective function can be written as follows:

$$\text{Min}\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=1}^{m} d_{ij}\, {x_{ij}}^k.$$

The model constraints are:

$$\sum_{j=1}^{n} x_{0j}^k = 1 \text{ ; k=1,...,m.} \qquad (1)$$

Constraint (1) guarantees that each vehicle will leave the depot and arrive at determined customer.

$$\sum_{i=0}^{n} x_{ip}^k - \sum_{j=0}^{n} x_{pj}^k = 0 \text{ ; p=0,..,n ; k=1,..,m.} \qquad (2)$$

Constraint (2) is about entrance and exit flows, guarantees that each vehicle will leave a determined customer and arrive back to the depot.

$$\sum_{k=1}^{m} y_i^k = 1 \text{ ; i=1,...,n.} \qquad (3)$$

Constraint (3) guarantees that the total demand of each customer will be fulfilled.

$$\sum_{i=1}^{n} q_i\, y_i^k \leqslant a_k \text{ ; k=1,...,m .} \qquad (4)$$

Constraint (4) guarantees that the vehicle capacity will not be exceed.

$$y_i^k \leqslant \sum_{j=0}^{n} q_{ij}^k \text{ ; i=1,..,n ; k=1,..,m .} \qquad (5)$$

Constraint (5) guarantees that the demand of each customer will only be fulfilled if a determined vehicle goes by that place. We can notice that , adding to constraint (5) the sum of all vehicles and combining to constraint (3) we have the constraint $\sum_{k=1}^{m} \sum_{i=0}^{n}$ $\geqslant 1$; j=0,..n , which guarantees that each vertex will be visited at least once by at least one vehicle.

$$b_i^k + d_i + t_{ij} - m_{ij} \ (1 - x_{ij}^k) \leqslant b_j^k. \text{ i=1,..,n . j=1,..,n . k=1,..,m .} \qquad (6)$$

Constraint (6) sets a minimum time for beginning the service of customer j in a determinedroute and also guarantees that there will be no sub tours. The constant m_{ij} is a large enough number for instance :

$$m_{ij} = l_i + t_{ij} - e_j \text{ .}$$
$$e_i \leqslant b_i^k \leqslant l_i \text{ ; i=1,..n .} \qquad (7)$$

Constraint (7) guarantees that all customers will be served within their time windows.

$${y_i}^k \geqslant 0 \text{ ; i=1,..,n. ; k=1,..,m .}$$
$${b_i}^k \geqslant 0 \text{ ; i=1,..,n. ; k=1,..,m .} \qquad (8)$$

Constraint (8) guarantees that the decision variables ${y_i}^k$ and ${b_i}^k$ are positive .

$${x_{ij}}^k \in 0 \ , \ 1 \text{ ; i=0,..,n ; j=0,..,n ; k=1,..,m.} \qquad (9)$$

Finally constraint (9) guarantees the decision variables x_{ij}^k to be binary.

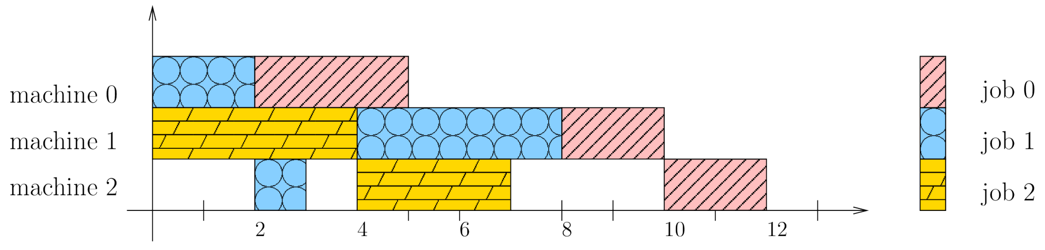
3.4 SIMILAIR PROBLEMS

There's a lot of similar problems to VRP's , one of them is the Job Shop Scheduling. Despite a number of similarities, vehicle routing problems and scheduling problems are typically solved with different techniques.

3.4.1 *Job Shop Scheduling*

An nXm job shop scheduling problem (JSP) consists of n jobs and m resources. Each job is a set of m completely ordered activities, where each activity has a duration for which it must execute and a resource which it must execute on. The total ordering defines a set of precedence constraints, meaning that no activity can begin execution until the activity that immediately precedes it in the complete ordering has finished execution. Each of the m activities in a single job requires exclusive use of one of the m resources defined in the problem. No activities that require the same resource can overlap in their execution and once an activity is started it must be executed for its entire duration (i.e. no pre-emption is allowed). The job shop scheduling decision problem is to decide if all activities can be scheduled, given for each job a release date of 0 and a due date of the desired makespan D, while respecting the resource and precedence constraints. The job shop scheduling decision problem is also NP-complete [30].
we can see an example in the Figure 16 shown below :

Figure 16: The Job Shop Scheduling [40]



3.4.1.1 *Mutual Reformulations*

In [30],two reformulations are presented :

- A reformulation of VRP as a flexible shop problem with transition time.
- A reformulation of JSP as a VRP.

We start by reformulate the VRP into JSP as follows:

Each vehicle is represented as a resource, and each customer visit as an activity. The distance between a pair of visits corresponds to a transition time between respective activities. Each activity can be performed on any resource, and is constrained to start execution within the time window defined in the original VRP. Each activity has a demand

for a secondary resource, corresponding to a visit's demand within a vehicle. For each resource R there are two special activities **Start(R)** and **End(R)**. Activities **Start(R)** and **End(R)** must be performed on resource R. **Start(R)** must be the first activity performed on R and **End(R)** must be the last. The transition time between **Start(R)** and any other activity **Ai** corresponds to the distance between the depot and the i th visit. Similarly the transition time between **End(R)** and **Ai** corresponds to the distance between the depot and the i th visit. The processing time of **Start(R)** and **End(R)** is zero. We associate a consumable secondary resource with every (primary) resource to model the capacity of vehicles.
Consequently a sequence of activities on a resource corresponds to a vehicle's tour in the VRP. In the resultant scheduling problem each job consists of only one activity, each activity can be performed on any resource, and there are transition times between each pair of activities. The problem is then to minimize the sum of transition times on all machines and maybe also to minimize the number of resources used.

We reformulate a JSP into a VRP as follows :

We have for each resource, a vehicle, and for each activity, a customer visit. The visits have a duration the same as that of the corresponding activities. Each visit can be made only by the vehicles corresponding to the set of resources for the activity. Any ordering between activities in a job results in precedence constraints between visits. Transition times between activities correspond to travel distances between visits. The deadline D imposes time windows on visits. Assuming we have m resources, and therefore m vehicles, we have 2m dummy visits corresponding to the departing and returning visits to the depot. A vehicle's tour corresponds to a schedule on a resource. For the nXm JSP we have a VRP with m(n+2) visits and m vehicles. Each visit can be performed only by one vehicle. Since there are no transition times in the pure JSP, there are no travel distances between visits, but visits have durations corresponding to those of the activities.
There are precedence constraints between those visits corresponding to activities in a job. The decision problem is then to find an ordering of visits on vehicles that respects the precedence constraints and time windows.

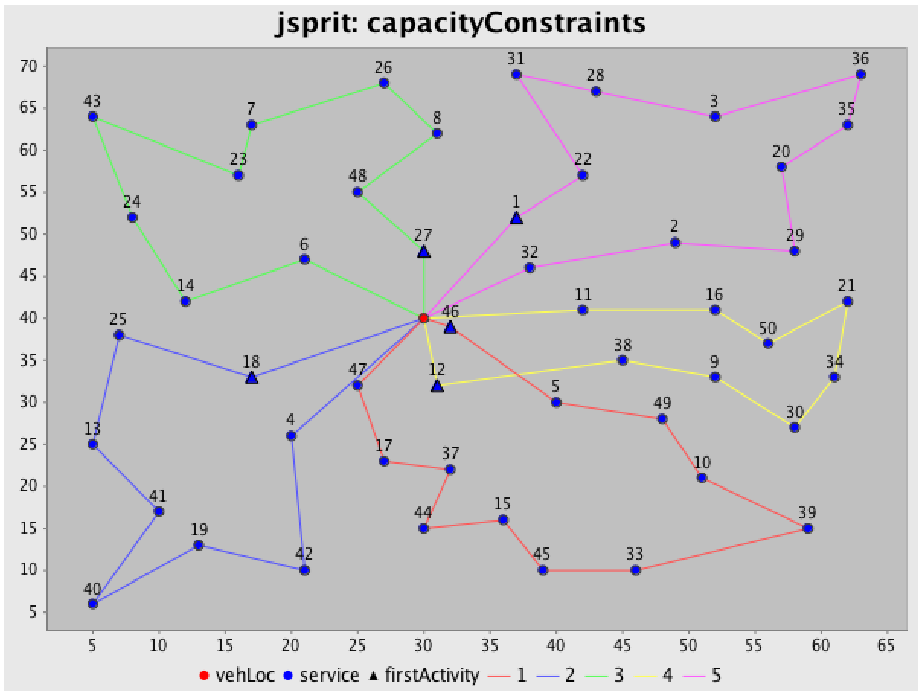
3.5 SOFTWARES OF VRP

In this section we 'll talk about the free softwares for solving Vehicle Routing Problem which are :

3.5.1 *Jsprit*

Is a Java based toolkit for solving rich traveling salesman (TSP) and vehicle routing problems (VRP). It is lightweight, flexible and easy-to-use, and based on a single all-purpose meta-heuristic [17]. In The Figure 17 we'll see an example about the Jsprit :

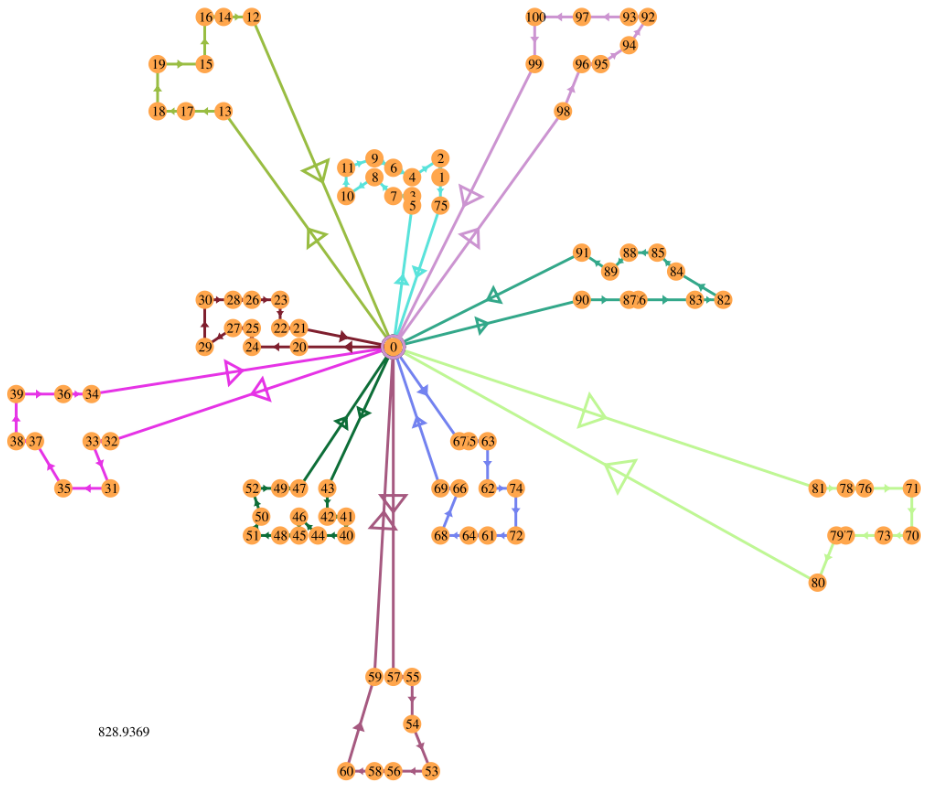
Figure 17: An Example About Jsprit [17]



3.5.2 *Open-VRP*

Open VRP is a framework to model and solve VRP-like problems for students, academics, businesses and hobbyist alike. This framework allows for quick implementation of simple TSP/VRP problems to more complicated VRPTW, PDPTW, MDCPVRPPDTW, or however cool you want to sound [43]. An example about the Open-VRP in the Figure 18 below :

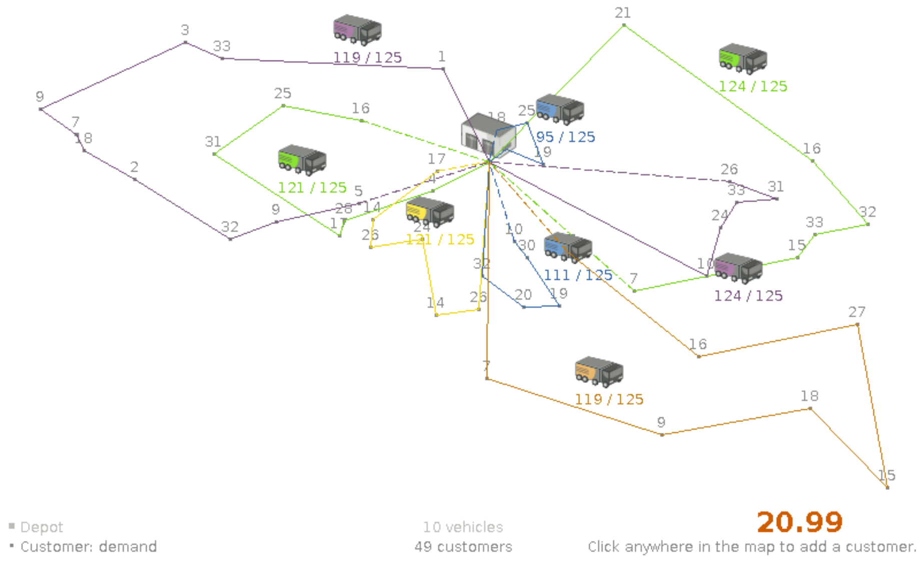
Figure 18: An Example About Open-VRP [43]



3.5.3 *OptaPlanner*

OptaPlanner is a lightweight, embeddable constraint satisfaction engine which optimizes planning problems. It solves use cases such as: VRP and Jsp etc [39]. The Following Figure 19 will show us a good example about the OptaPlanner :

Figure 19: An Example About OptaPlanner [39]



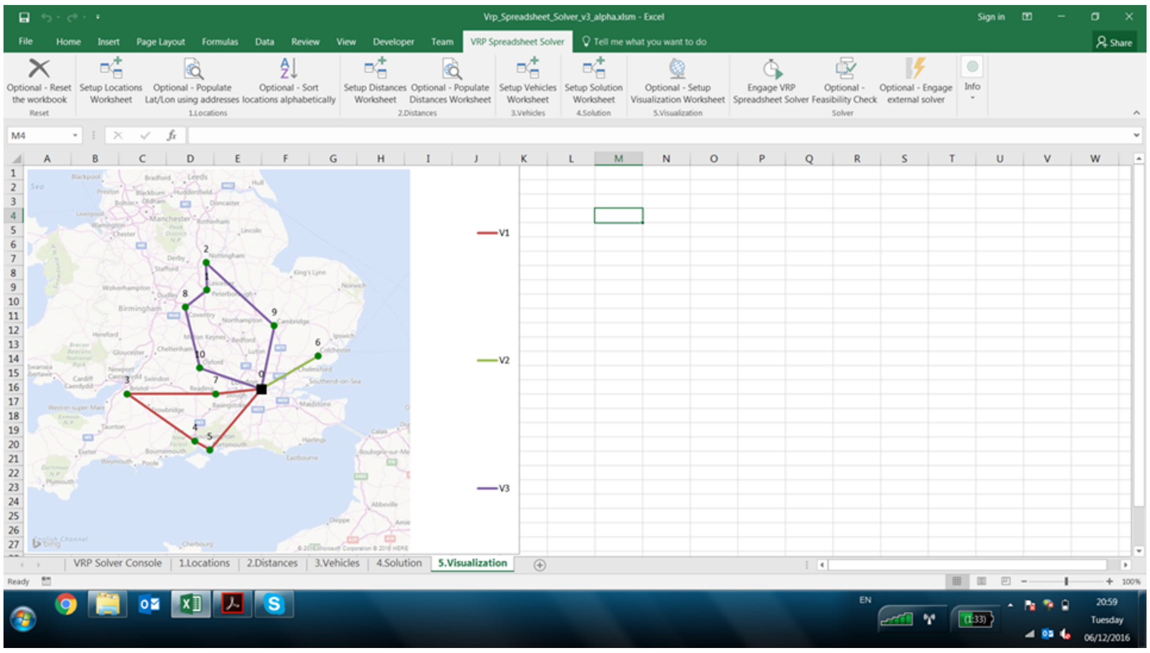
3.5.4 *Symphony*

Open-source solver for mixed-integer linear programs (MILPs) with support for VRPs [7].

3.5.5 *VRP Spreadsheet Solver*

Is an open source unified platform for representing, solving, and visualising the results of Vehicle Routing Problems (VRPs). It unifies Excel, public GIS (Geographic Information Systems) and metaheuristics. It can solve Vehicle Routing Problems with up to 200 customers [46]. The Figure 20 gives us an example about VRP Spreadsheet Solver :

Figure 20: An Example About VRP Spreasheet Solver [46]


Vrp_Spreadsheet_Solver_v3_alpha.xlsm - Excel
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Developer
Team
VRP Spreadsheet Solver
Tell me what you want to do
Optional - Reset the workbook
Setup Locations Worksheet
Optional - Populate Lat/Lon using addresses
Optional - Sort locations alphabetically
Setup Distances Worksheet
Optional - Populate Distances Worksheet
Setup Vehicles Worksheet
Setup Solution Worksheet
Optional - Setup Visualization Worksheet
Engage VRP Spreadsheet Solver
Optional - Feasibility Check
Optional - Engage external solver
Info
V1
V2
V3
VRP Solver Console
1.Locations
2.Distances
3.Vehicles
4.Solution
5.Visualization

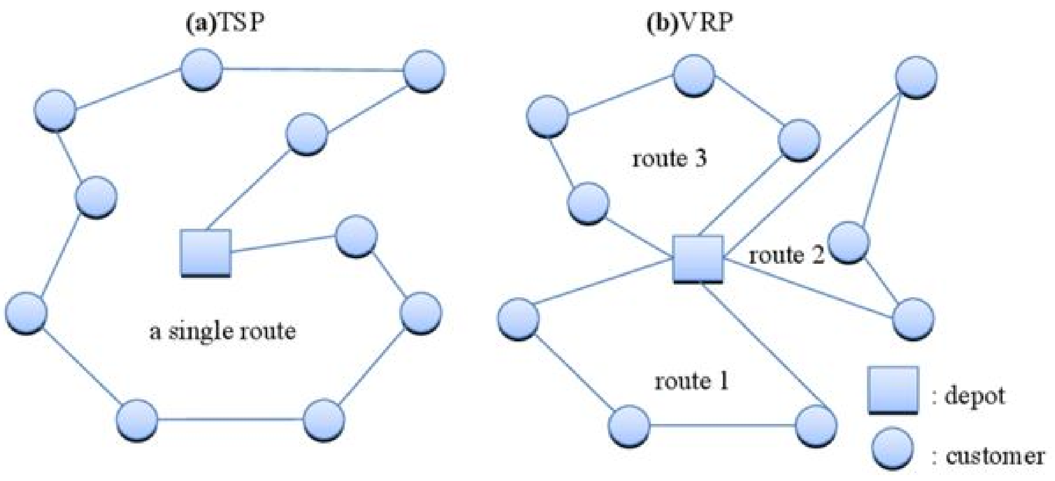
3.6 THE ORIGINS OF VRP

One of the simplest, but still NP-hard, routing problems is probably the traveling salesman problem (TSP).

3.6.1 *Definition*

Given a set of cities and the cost of travel (or distance) between each possible pairs, the TSP, is to find the best possible way of visiting all the cities and returning to the starting point that minimize the travel cost (or travel distance) [31]. The next Figure 21 will show us the illustration of the traveling salesman problem (TSP) and vehicle route problem (VRP) route patterns.

Figure 21: Illustration Of The Traveling Salesman Problem (TSP) And Vehicle Route Problem (VRP) Route Patterns [20]



3.6.2 *History*

The traveling salesman problem (TSP) were studied in the 18th century by a mathematician from Ireland named Sir William Rowam Hamilton and by the British mathematician named Thomas Penyngton Kirkman. Detailed discussion about the work of Hamilton Kirkman can be seen from the book titled Graph Theory . It is believed that the general form of the TSP have been first studied by Kalr Menger in Vienna and Harvard. The problem was later promoted by Hassler, Whitney and Merrill at Princeton [31].

3.6.3 *Classification*

Broadly speaking, the problem of Travelling Salesman is classified as:

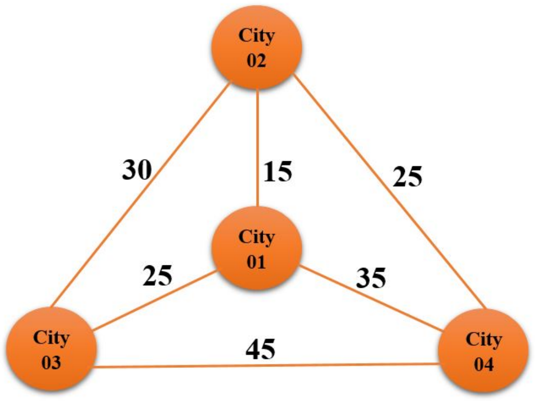
1. Symmetric travelling salesman problem (sTSP).
2. Asymmetric travelling salesman problem (aTSP).
3. Multi travelling salesman problem (mTSP).

This section presents description about these three widely studied TSP.

3.6.3.1 *Symmetric TSP*

In the symmetric TSP, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of possible solutions [45]. The next Figure 22 will give us an explanation :

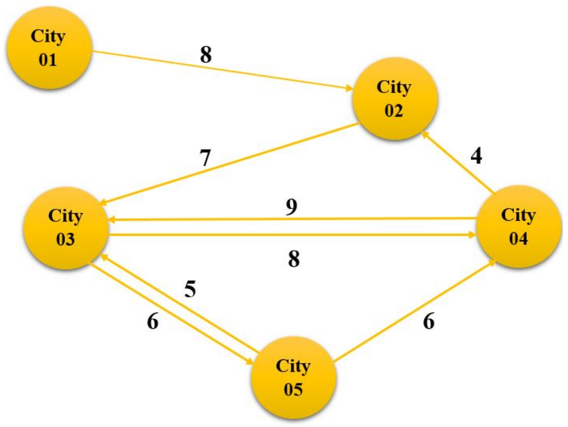
Figure 22: Symmetric TSP



3.6.3.2 *Asymmetric TSP*

In the asymmetric TSP, paths may not exist in both directions or the distances might be different, forming a directed graph. Traffic collisions, one-way streets, and airfares for cities with different departure and arrival fees are examples of how this symmetry could break down [45]. The Figure 23 is an example about aTSP :

Figure 23: Asymmetric TSP



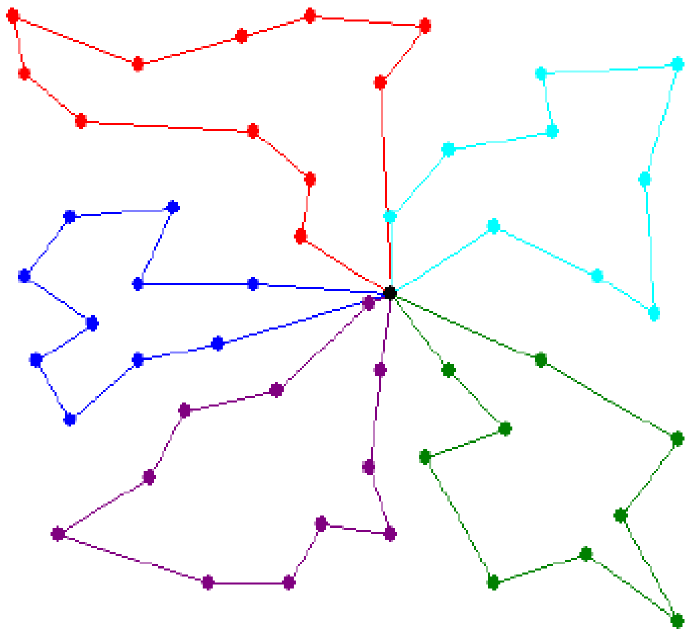
3.6.3.3 *Multi TSP*

In [31], the mTSP is defined as : In a given set of nodes, let there are m salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the mTSP consists of finding tours for all m salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting all nodes is minimized. The cost metric can be defined in terms of distance, time, etc.
Possible variations of the problem are as follows :

- **Single vs. multiple depots** : In the single depot, all salesmen finish their tours at a single point while in multiple depots the salesmen can either return to their initial depot or can return to any depot keeping the initial number of salesmen at each depot remains the same after the travel.
- **Number of salesmen** : The number of salesman in the problem can be fixed or a bounded variable.
- **Cost** : When the number of salesmen is not fixed, then each salesman usually has an associated fixed cost incurring whenever this salesman is used. In this case, the minimizing the requirements of salesman also becomes an objective.
- **Timeframe** : Here, some nodes need to be visited in a particular time periods that are called time windows which is an extension of the mTSP, and referred as multiple traveling salesman problem with specified timeframe (mTSPTW). The application of mTSPTW can be very well seen in the aircraft scheduling problems.
- **Other constraints** : Constraints can be on the number of nodes each salesman can visits, maximum or minimum distance a salesman travels or any other constraints. The mTSP is generally treated as a relaxed vehicle routing problems (VRP) where there is no restrictions on capacity. Hence, the formulations and solution methods for the VRP are also equally valid and true for the mTSP if a large capacity is assigned to the salesmen (or vehicles). However, when there is a single salesman, then the mTSP reduces to the TSP.

The following Figure 24 it is an example about the multi TSP with a single depot :

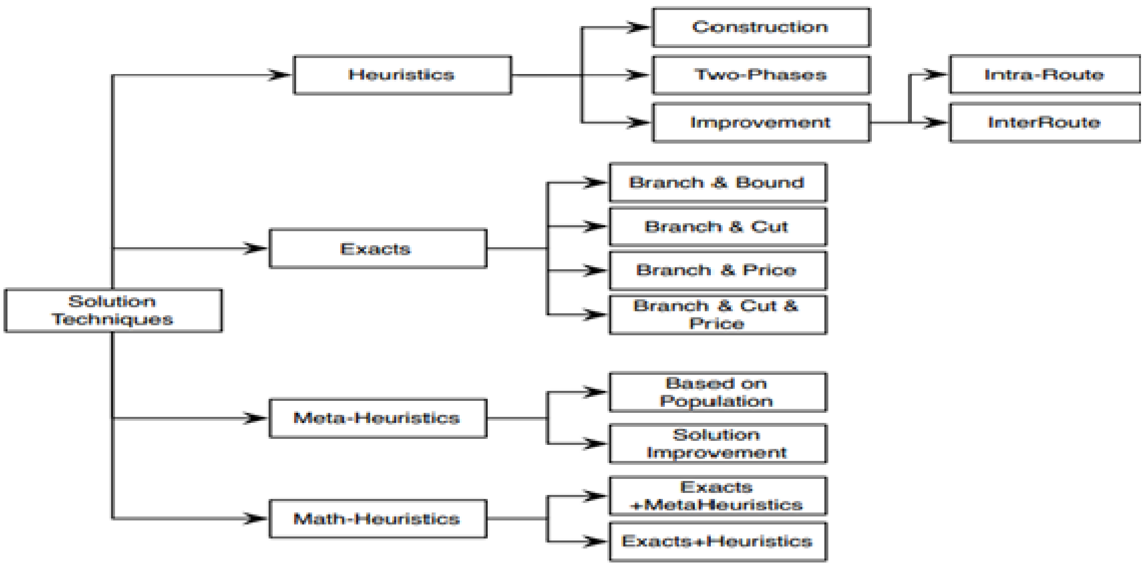
Figure 24: Multi TSP With A Single Depot [31]



3.7 RELATED WORKS

By related works we mean work that proposes a method we discussed in the previous chapter to solve the VRP, that's mean the solution techniques that solve the problem. Solution techniques can be classified as depicted in the next Figure 25 :

Figure 25: VRP Solution Techniques [9]



3.7.1 *Solving the VRP using a parallel Branch and Bound algorithm*

G.H. Dastghaibifard and al [12],propose in their research a new parallel algorithm. Instead of using a shared-memory they use a multi-computer environment. As well they use a decentralized load balancing method. Might as well shows that by revising existed algorithms can archive a good performance and lowers communication between processes. Their algorithm is implemented to solve the famous Capacitated Vehicle Routing Problem (CVRP). Finally, the experimental results show that the algorithm have a great efficiency.

3.7.2 *Solving the VRP using an improved Tabu Search approach*

Hongmei Jia and al [14], they designed a new Tabu Search by introducing mutation and mixed local searching tactics for overcoming the weaknesses of the current TS. As well after comparing the Improved Tabu Search with other algorithms, the excellent performance of the Improved Tabu Search is shown by the qualities of the improved algorithm o the VRP whether the size of the problem is big or small, and the stability of the algorithm. Finally the the convergent speed is fast, the calculating efficiency is high.

3.7.3 *Solving the VRP using Genetic Algorithm*

Abdul Kadar Muhammad Masum and al [1], showing that the genetic algorithms have provide a very interesting approach to solve problems where an exact method can not be applied. They decided to implement some custom insertion heuristic which helps the system to faster approach a good solution. The choice of the crossover method was pretty intuitive. They implemented one that inserts new elements as a subroute using their heuristic method and another one, which does a simple sequence based crossover. On one hand they get a quite good results but in the most cases they are unable to get to the best results. Thus they tuned the system to have a balance between finding fast a solution and limiting the search space.

3.7.4 *Solving the VRP with time windows with a Mmlti-start Local Search algorithm*

Braysy et al. (2004)[3], presented a heuristic search method that hybridized ideas of evolutionary computation with some other search techniques, such as Tabu Search (TS) or

Simulated Annealing (SA) which have also been used for solving VRPs. Most of the hybrid methods presented used local search mutation instead of the random mutation operators. In the first phase, an initial solution was created by either the cheapest insertion heuristic or the sectoring based genetic algorithm GIDEON. The second phase applied one of the following search procedures that use the interchange mechanism : a local search descent procedure, a SA algorithm or a hybrid SA and TS, where TS is combined with the SA based acceptance criterion to decide which moves to accept from the candidate list. The main feature of the local search procedures was that infeasible solutions with penalties were been allowed if considered attractive .

3.7.5 *Solving the real-world vehicle routing problems using Ant Colony optimization*

In [33], a contribution describes the metaheuristic ant colony optimization and how it can be successfully used to solve a number of variants of the basic VRP.
The main part presents two industrial-scale applications of ACO for the solution of static VRP problems : a VRP with time windows and a VRP with pickup and delivery. Then the contribution focuses its attention on one important dynamic variant of the VRP : the on-line VRP. The problem is receiving increasing attention based on its relevance to real world problems, in particular for distribution in urban environments. The applications of ACO on real-world VRP shows that this metaheuristic inspired by ants has become an important tool in applied operations research.

3.8 CONCLUSION

This chapter represents the literature review of vehicle routing problem by introducing the problem elapsing by the variants of the VRP and the mathematical formulation.
Thereafter we saw the free VRP solvers .Subsequently we saw that the VRP belong's to the Class NP Hard . As well we discussed the similar problems of the VRP and the ralted works.
Solving this problem is only done by approximate way , because till this date there's no algorithm can solve this problem. In the next chapter we are going to present our contribution by modelizing a tool to solve the VRP.

Part III

OUR CONTRIBUTION

In this part we propose a tool in order to solve the vehicle routing problem (VRP).In the Vehicle Routing Problem (VRP), the aim is to design a set of m minimum cost vehicle routes through n customer locations, so that each route starts and ends at a common location and some side constraints are satisfied.

4 OUR CONTRIBUTION

4.1 INTRODUCTION

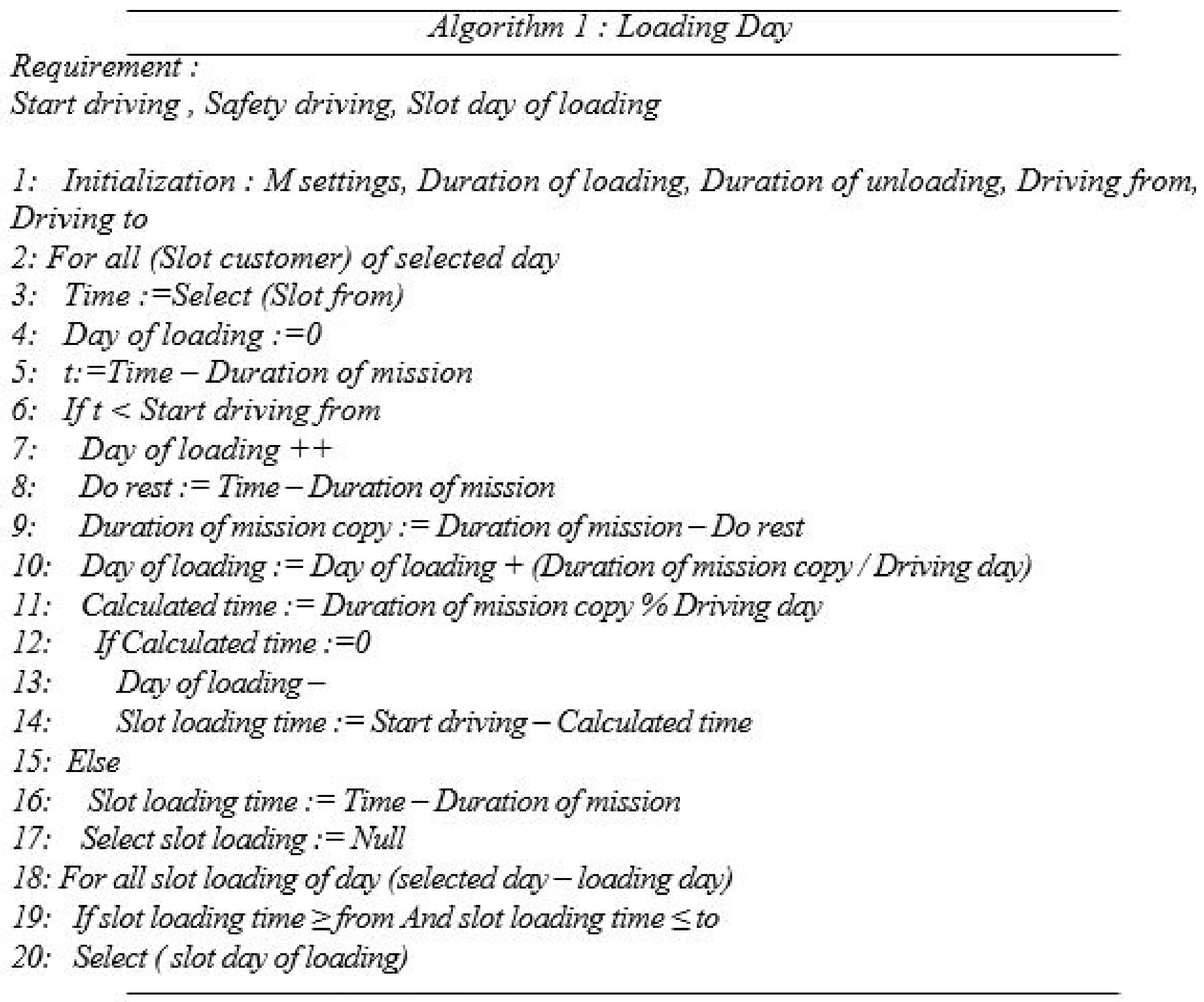
Vehicle Routing Problem (VRP) is one of the most challenging combinatorial optimization tasks, a central problem in the areas of transportation, distribution and logistics. It has been the subject of several studies motivated by the difficulty of its resolution and its many practical applications in logistics.
VRP is to design routes for N vehicles with M depots and P customers in order to meet the given constraints . Looks pretty complicated, doesnt it ?.
Our main objective in our contribution is to develop a tool to optimize the vehicle routing problem (VRP) . This chapter will show us how to modelize this tool.

4.2 PSEUDO ALGORITHM

In this section we'll see our ideas about how we'll work by presenting the pseudo algorithms of our contribution. Here's a look about our algorithms :

- The day of loading :

Figure 26: Algorithm Of The Loading Day



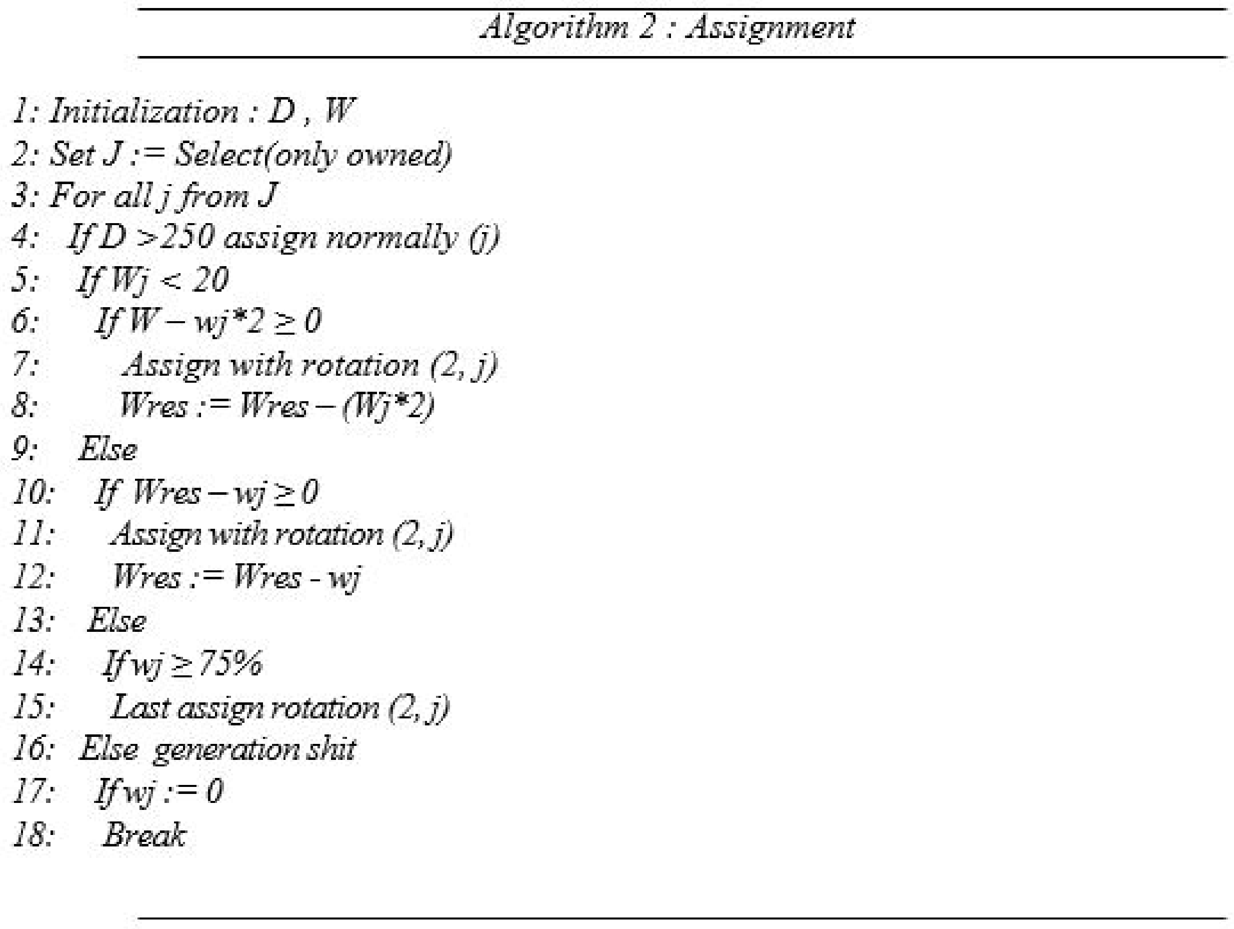
```
Algorithm 1 : Loading Day
Requirement :
Start driving , Safety driving, Slot day of loading

1:  Initialization : M settings, Duration of loading, Duration of unloading, Driving from,
Driving to
2: For all (Slot customer) of selected day
3:  Time :=Select (Slot from)
4:  Day of loading :=0
5:  t:=Time – Duration of mission
6:  If t < Start driving from
7:    Day of loading ++
8:    Do rest := Time – Duration of mission
9:    Duration of mission copy := Duration of mission – Do rest
10:   Day of loading := Day of loading + (Duration of mission copy / Driving day)
11:   Calculated time := Duration of mission copy % Driving day
12:     If Calculated time :=0
13:       Day of loading –
14:       Slot loading time := Start driving – Calculated time
15: Else
16:    Slot loading time := Time – Duration of mission
17:   Select slot loading := Null
18: For all slot loading of day (selected day – loading day)
19:   If slot loading time ≥ from And slot loading time ≤ to
20:   Select ( slot day of loading)
```

This proposed algorithm will help us to calculate the day of loading of the products. In the next sections we'll explain more.

- Assignment:

Figure 27: Algorithm Of Assignment



```
Algorithm 2 : Assignment

1: Initialization : D , W
2: Set J := Select(only owned)
3: For all j from J
4:   If D >250 assign normally (j)
5:    If Wj < 20
6:      If W – wj*2 ≥ 0
7:        Assign with rotation (2, j)
8:        Wres := Wres – (Wj*2)
9:    Else
10:     If Wres – wj ≥ 0
11:       Assign with rotation (2, j)
12:       Wres := Wres - wj
13:    Else
14:      If wj ≥ 75%
15:        Last assign rotation (2, j)
16:    Else  generation shit
17:     If wj := 0
18:      Break
```

This Proposed algorithm of assign trucks will help us know the way of how to assign trucks and help us know the available trucks in the depot. More explanation in the next sections.

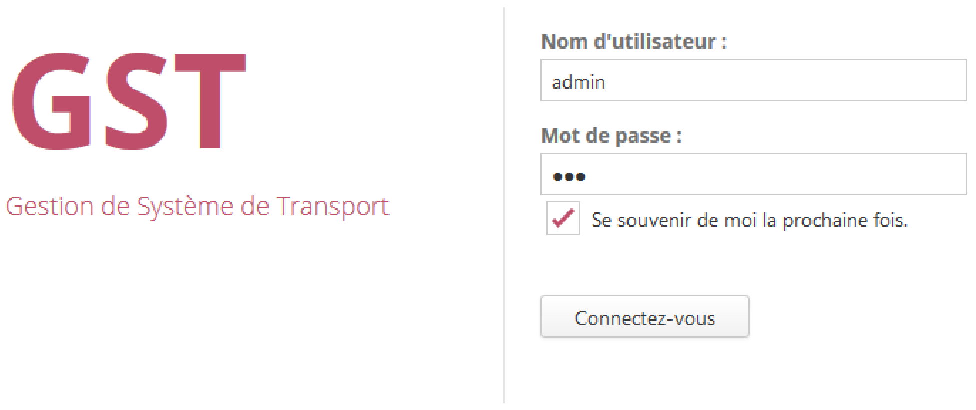
4.3 GLOBAL ARCHITECTURE

In this section, we will briefly describe the structure of our tool of VRP. We will be focusing on the technical details, for which the interested user can refer to the userâs manual.
In order to control access to our database in terms of security, we have opted for the

principle of authentication which consists in the creation of a username and a password. The following window is displayed when the connection is established.

From this window, we can perform several operations such as displaying the list of products, the list of customers or to add a customer, a type of truck, an order ... etc.
A look to our authentication window of our tool in the following Figure 28 :

Figure 28: Authentication Window



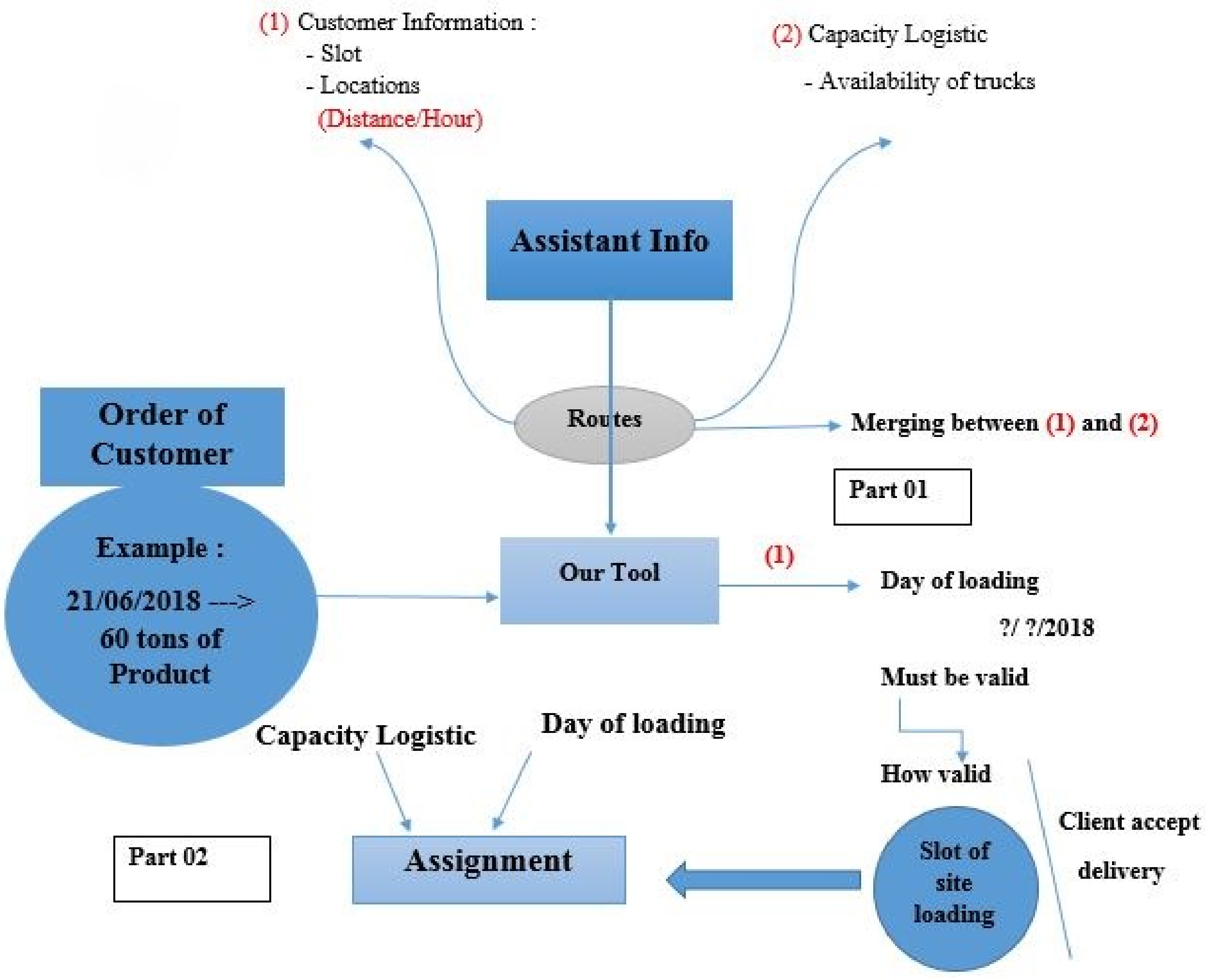
Nom d'utilisateur :
admin
Mot de passe :
•••
✓ Se souvenir de moi la prochaine fois.

Our tool keeps the data about the elements of a VRP in separate worksheets, and adopts an incremental flow of information.
The architecture of our tool is based of multiple characteristics which are :

- The main depot of the company .
- Customers
- Logistics (trucks,..etc.).
- Planning of roads.
- Assignment.

Here's a look about the global architecture of our tool in the Figure 29 below :

Figure 29: The Global Architecture



Starting by explaining the global architecture of the tool which is divided into two parts in the following :

- Part 01 :

  The company have their depot. Every company receives commands of purchase goods from their customers, every customer have a different order.

  An order is where we can find in it the information about the needs of the customers. In every order we find the following informations which is the location of the customer and itâs time windows (slot) of serving the customer and the amount of the product the customer needs, and the delivery time.

Then the company have their logistics (fleet of trucks). Every truck have a trailer and every trailer have a different capacity. Here we have three different types of trucks which we will see in the next section.

After receiving the orders from the customers our tool generates for us the day of loading which is an extract from the orders of customers. The day of loading must be valid , to be valid we must check the slot (time windows) of the customer because in this interval of time the customers accepts the delivery.

Now moving to the main idea which is the planning of roads which is the merging between the order of purchasing and the priority of the customer :

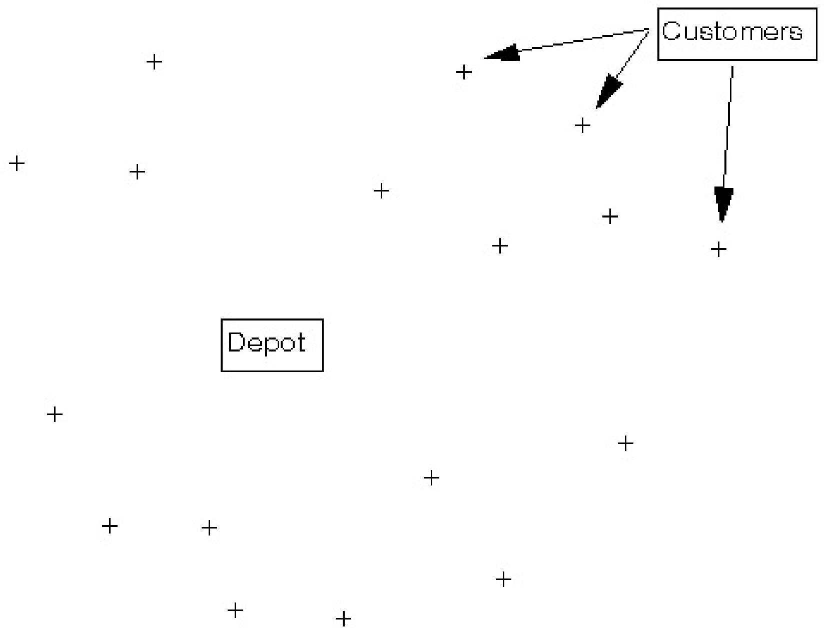
After receiving the order of purchase from the customers, this point planning of roads is done by creation and building of a calendar where we can find :

– The customers with their orders and locations and their time windows (slot).
– The time window of loading for each truck.
– Each truck has a time period within which it must leave the depot.
– Each truck has a total working time from departure to arrival back at the depot.

Also it generates the distance between the depot and the location of the customer and the journey time . Which gave us the starting point from the site of loading to the arrival point (the customer).

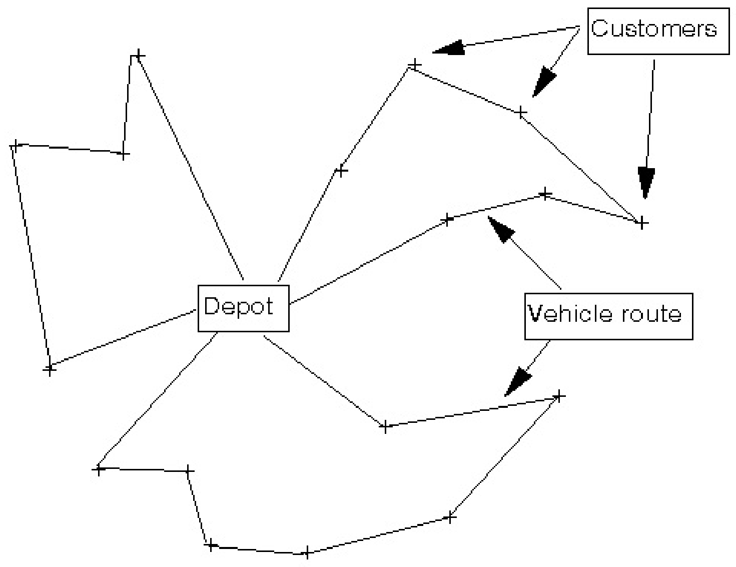
Consider the situation shown below where we have a depot surrounded by a number of customers who are to be supplied from the depot.

Figure 30: A Depot Surrounded By A Number Of Customers



The depot manager faces the task of planning routes (such as those shown below) for his delivery trucks and this problem of route planning is known as the vehicle routing or vehicle scheduling problem.

Figure 31: Planning Of Routes



Also the planning generates the safety settings for the driver of the trucks like :

- No driving in the night.

  - Driving for two (02) hours needs to stop and rest for 30 min.
  - Max driving time. Ex : 14 h.

  The main idea of the planning is done by advance that 's mean our tool plan for a whole week. The company receives the orders from customers a week before the delivering of products.

  Likewise in the section of planning there's a specific day which is the day before the starting of the delivery. In this specific day the company calls customers to confirm their orders in order to know if the customer is in need for the product or no. If the customer says he doesn't need any product he'll be deleted from the planning of this week.

- Part 02 :

  Ending with the assignment. After the company confirms the orders of their customers in the day before delivery, come the last part of assignment.

  In this part, the company sees their logistic capacity (availability of trucks) and the day of loading to assign every available truck to load goods and start the delivery to fulfil the commands of the customers.

  Each vehicle has a total working time from departure to arrival back at the depot.

  Each vehicle has a time period within which it must leave the depot, typically to ensure that space is available for incoming vehicles to resupply the depot.

  In the next section we'll see the logistics and the types of trucks that the company have.

4.4 WHAT WE NEED IN OUR TOOL

To build a tool to solve a problem, you have to know an essential part which is computing (informatics), by understanding how to design and build efficient algorithms taking into account all what you know about the vehicle routing problem.

Our model uses the following data that is available as inputs :

4.4.1 *Product*

An article or substance that is manufactured for sale. The unity of measuring of any product it's by kilograms (KG).

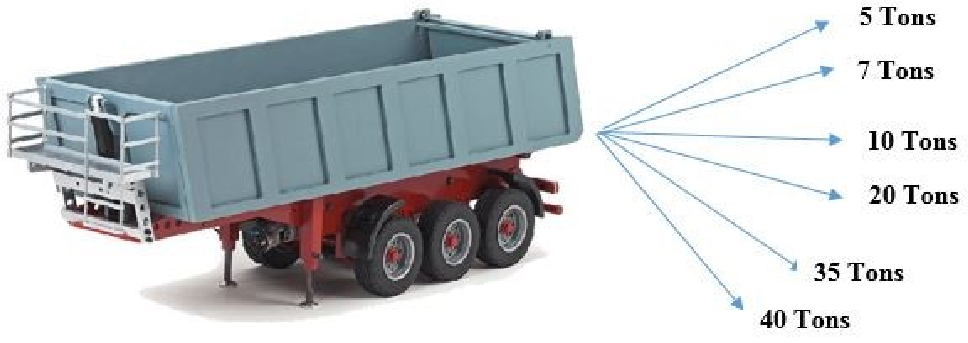
4.4.2 *Logistics*

Here we are talking about the availability of the fleet of trucks that serve the customers from one depot .

The trucks leave and return to the depot . Each truck has a number of time periods during which it does nothing (driver rest periods). There is a set of trucks with different capacities. Every trailer have a different capacity.

An example about the capacity of the trailer :

Figure 32: Capacity Of A Trailer



The most important thing here is that we have a three (03) different types of trucks that are :

4.4.2.1 *The owned trucks by the company*

It is a type of trucks used for transporting goods and products. This are the trucks of the company itself. Each truck from this trucks have a defined tonnage and a trailer.

This trucks have a constraint to start serving and transporting products to the customers depot which are :

- If the distance from the company to the location of the customer surpass 250 km then the company will order them to start serving.
- If the capacity of the order of the customer surpass 20 tons , than the company uses these trucks .

4.4.2.2 *Available trucks*

The type of this trucks means that this trucks have been rented by the company throughout the year.

The company aims to use these trucks all the time, and it is important for the company to use these trucks in any order delivery.

But these type of trucks have conditions to move which are :

- If the distance is less than or equal to 250 km, then this trucks start delivering .
- If the capacity of the order is equal or less than 20 tons , then this strucks start delivering.

4.4.2.3 *Spot trucks*

This type of trucks is meant to use when the company runs out of trucks for delivering the goods to the customers.

The company have a list of drivers with this type of trucks, they call them and agrees with them about the delivery, in one condition which is that when the company calls them, the drivers must came immediately.

There is no conditions of starting the delivery , this type of trucks goes anywhere with any capacity.

4.4.3 *Customers*

Every client have characteristics which are the following :

4.4.3.1 *Locations*

Every customer have a location and could have a multiple locations , it could be in the same place or a diiferent place from the first location .

Every location of a custumer is far from the depot by a distance and time.

The unity of measuring the distance is by Kilometres (km) and the unity of measuring the time is by hours .

4.4.3.2 *Slot (Time Windows)*

A fixed interval of time, where the delivery locations of the customers have time windows within which the deliveries (or visits) must be made.

Every customer have a specific day to serve him, in this specific day the customer have an interval of time , the earliest and the latest times.

For example a customer might be prepared only to accept delivery between 10.30 and 11.30 or between 14.00 and 16.15. These two periods of time are the time windows for the customer. Time windows are convenient to customers as they know when delivery is likely to occur and they can schedule deliveries to suit the work pattern of their staff (who may be needed to check incoming goods, sign for them, help unload them, etc). Time windows are inconvenient to delivery companies as they limit their flexibility (e.g. two customers right next door to each other might have very different time windows).

If the truck of delivery arrives early, he cannot deliver to the customer till the specific time windows in that day.

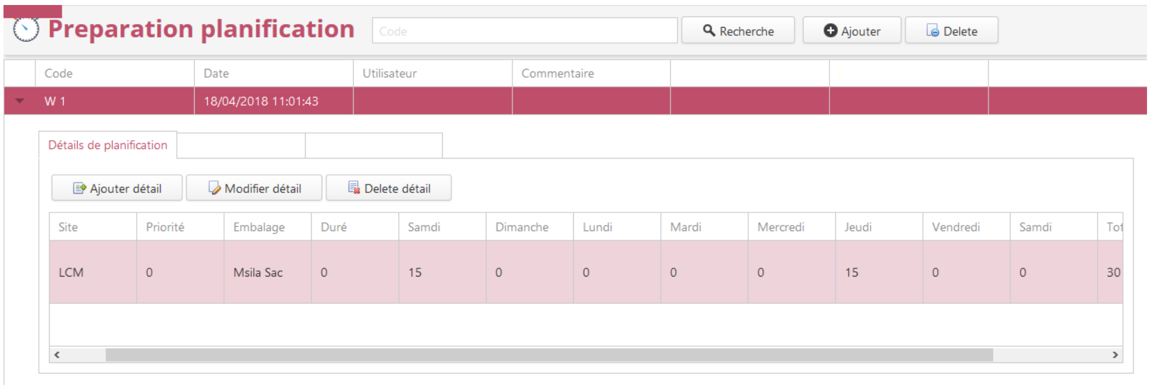
If the truck arrives late, a penalty of late must paid from the company. Hereâs an example :

4.5 PRE-PROCESS OPERATIONS

4.5.1 *Order of product*

Here we can see the orders of the customers during the whole week . An example about it in the next Figure 33 :

Figure 33: Order Of Product



Preparation planification

Code	Date	Utilisateur	Commentaire
W 1	18/04/2018 11:01:43		

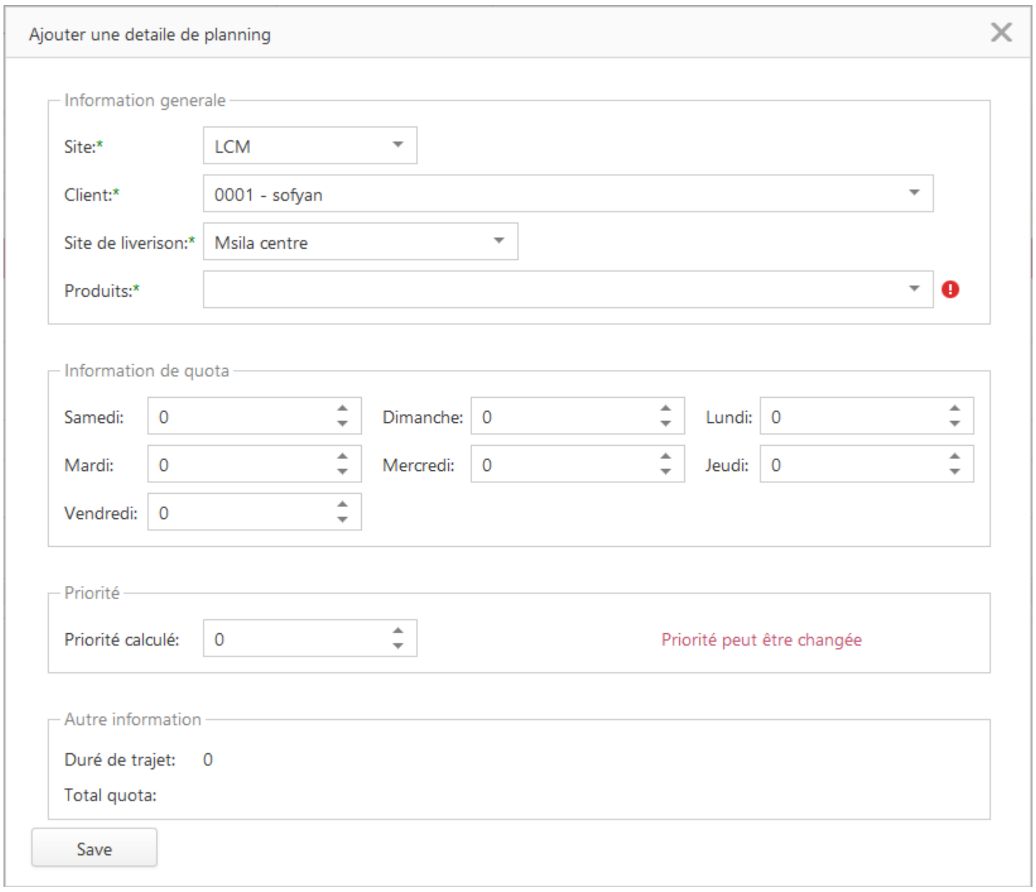
Détails de planification

Site	Priorité	Embalage	Duré	Samdi	Dimanche	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samdi	Tot
LCM	0	Msila Sac	0	15	0	0	0	0	15	0	0	30

As we see in the Figure 33 that's an example about a simple order of product. Like we see there's the site and the product and the tonnage in a specific day.

in the following Figure 34 we'll see how to add an order of a customer :

Figure 34: Add Of An Order Of Product



Ajouter une detaile de planning

Information generale

Site:*	LCM
Client:*	0001 - sofyan
Site de liverison:*	Msila centre
Produits:*	

Information de quota

Samedi:	0	Dimanche:	0	Lundi:	0
Mardi:	0	Mercredi:	0	Jeudi:	0
Vendredi:	0				

Priorité

Priorité calculé: 0 Priorité peut être changée

Autre information

Duré de trajet: 0

Total quota:

To add an order of a product we must complete the general informations. Like we see the site of loading and name of the client also the product and the delivery site.

Also we see any customer can have a delivery in any day he wants. For example the customer order a 15 tons in saturday and a 30 tons in thursday. Of course every customer have a prioriy.

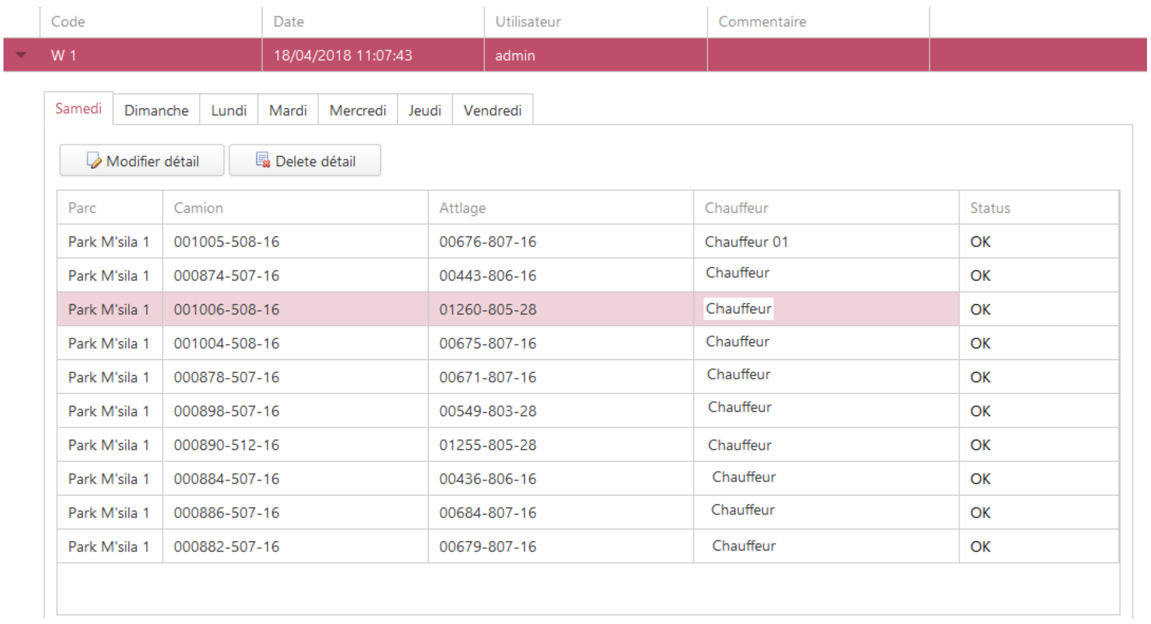
4.5.2 *Capacity logistic*

In this section we'll see the capacity logistic of the company of the whole week of the fleet of trucks that they have.

1. The owned trucks by the company.
2. Available trucks.
3. Spot trucks.

An example about it in the next Figure 35 :

Figure 35: Logistic Capacity



Code	Date	Utilisateur	Commentaire
W 1	18/04/2018 11:07:43	admin	

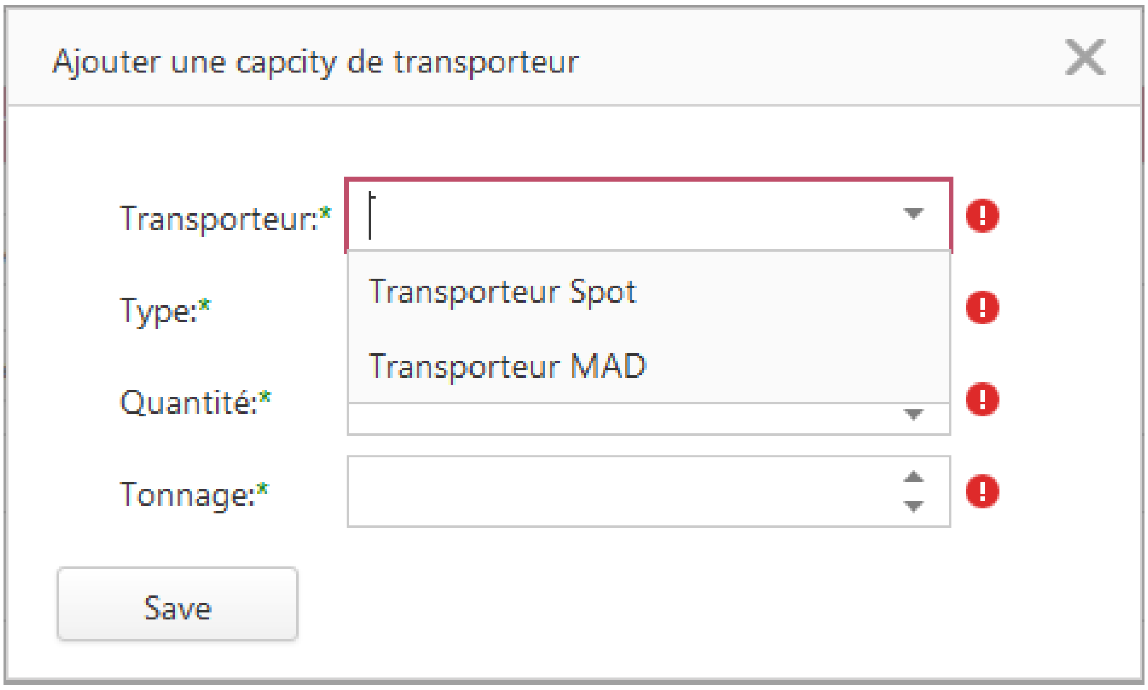
Parc	Camion	Attlage	Chauffeur	Status
Park M'sila 1	001005-508-16	00676-807-16	Chauffeur 01	OK
Park M'sila 1	000874-507-16	00443-806-16	Chauffeur	OK
Park M'sila 1	001006-508-16	01260-805-28	Chauffeur	OK
Park M'sila 1	001004-508-16	00675-807-16	Chauffeur	OK
Park M'sila 1	000878-507-16	00671-807-16	Chauffeur	OK
Park M'sila 1	000898-507-16	00549-803-28	Chauffeur	OK
Park M'sila 1	000890-512-16	01255-805-28	Chauffeur	OK
Park M'sila 1	000884-507-16	00436-806-16	Chauffeur	OK
Park M'sila 1	000886-507-16	00684-807-16	Chauffeur	OK
Park M'sila 1	000882-507-16	00679-807-16	Chauffeur	OK

In the Figure 35 we see the the owned trucks of the company which are ready to the delivery.

Every truck have a regestration number with a trailer also with a different regsistration number. We see also the driver of every truck, with their status. The status is Ok means they are ready for the delivery also means that the trucks aren't in the maintenance park.

In other hand when the owned trucks are all out , we can add the other types of trucks like we discussed in previous section. Here's an example about it in the follownig Figure 36 :

Figure 36: Add Of Logistic Capacity



Ajouter une capcity de transporteur

Transporteur:*	
Type:*	
Quantité:*	
Tonnage:*	

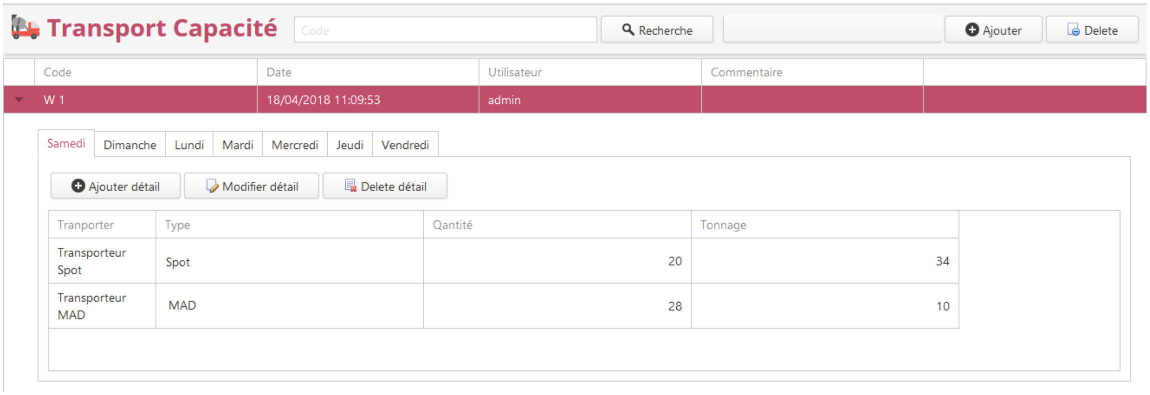
Transporteur Spot

Transporteur MAD

Save

We can see an example about the other types of trucks with an order to deliver in the following Figure 37 :

Figure 37: Spot And Available Trucks



Transport Capacité

Code	Date	Utilisateur	Commentaire
W 1	18/04/2018 11:09:53	admin	

Tranporter	Type	Qantité	Tonnage
Transporteur Spot	Spot	20	34
Transporteur MAD	MAD	28	10

We see the spot trucks and the available(Rented) trucks with the quantity of trucks and the exact tonnage to deliver.

4.5.3 *Generation of Routes*

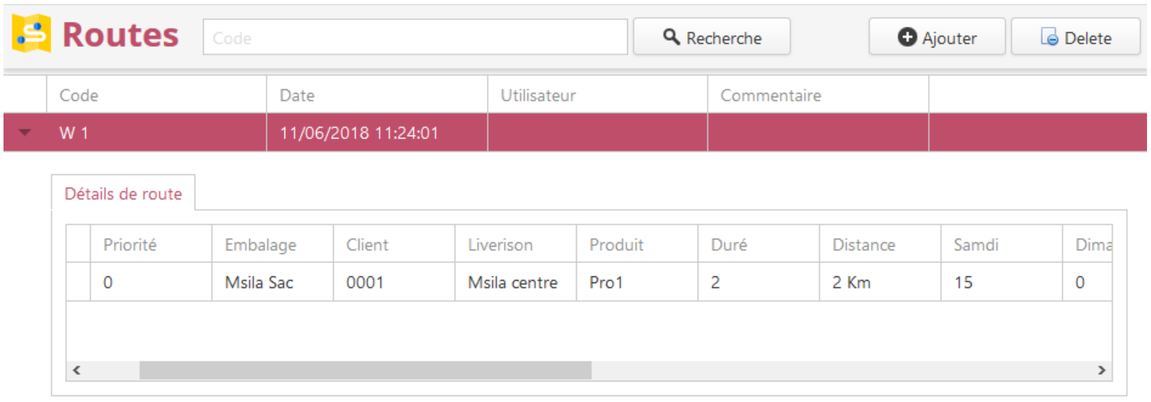
The main idea of the generation of roads it is the merging between the order of purchasing and the priority of the customer.

The Priority means which customer will be served first, to know the priority we must check the database to classified the customers in the right priority.

Also in the generation of the roads we must verify if the order of the purchase is validated or no , in order to serve the customers in the right priority.

Here is an example about the idea in the next Figure 38 :

Figure 38: Generation Of Routes



Code	Date	Utilisateur	Commentaire
W 1	11/06/2018 11:24:01		

Priorité	Embalage	Client	Liverison	Produit	Duré	Distance	Samdi	Dima
0	Msila Sac	0001	Msila centre	Pro1	2	2 Km	15	0

4.6 THE CORE OF THE TOOL (VRP ALGOITHM)

4.6.1 *Description*

In this section we'll describe the two main parts of our algorithm which are :

- The day of loading.
- Assignment.

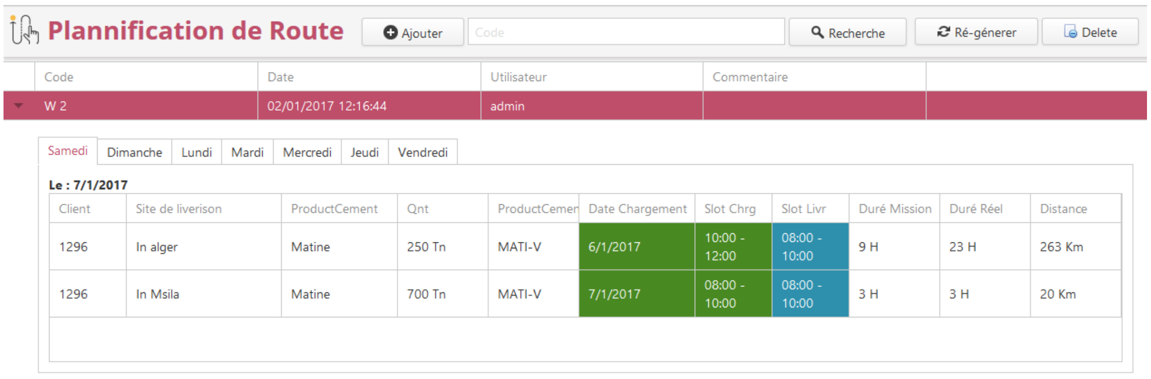
we'll begin by the description of the part (01) which is :

- The day of loading :

  This part consist of knowing the date of loading , and to know the date of loading we need an essential information which is the customer and his informations : slot (time windows),the distance of the location from the depot and the mission duration from the depot to the location.

In the next Figure 39 we'll see an example then we'll continue the explanation.

Figure 39: Planning Of Routes



Code	Date	Utilisateur	Commentaire
W 2	02/01/2017 12:16:44	admin	

Le : 7/1/2017

Client	Site de liverison	ProductCement	Qnt	ProductCemen	Date Chargement	Slot Chrg	Slot Livr	Duré Mission	Duré Réel	Distance
1296	In alger	Matine	250 Tn	MATI-V	6/1/2017	10:00 - 12:00	08:00 - 10:00	9 H	23 H	263 Km
1296	In Msila	Matine	700 Tn	MATI-V	7/1/2017	08:00 - 10:00	08:00 - 10:00	3 H	3 H	20 Km

In this example we have customer number one (01) with his location in **'Alger'** with a distance from the depot by 263 KM, duration mission of 9 hours, and a slot (time windows) from [8.00to10.00].

This Customer will receive his order in the date of 07/01/2017 from [8.00to10.0]. Now in order to calculate the date of loading, our tool will divide the interval of time to two sides.

The first is : From 8.00 in 07/01/2017 , we put the date of when the customer receives the product which is 07/01/2017, and we substract it from the duration of the mission which is 9 hours, which lead us to decrease in time.

The new found time will also be subject to the following constraints :

1. No driving in night.
2. Driver rest periods
3. Max hours of driving

We'll find a new date which is : From 8.00 in 06/01/2017, but this is not the final date of loading until we verify the day of loading is valid or not.

we verify the slot (time windows) of loading site, in this example it's : [10.00to12.00] and we verify if the required quantity by the customer is $\geqslant 0$ in the site of the loading.

If all of this constraints are valid then the date of the loading is validated.

But in our example the date it's not validated because of the site of loading have an interval from 10.00 to 12.00 and our date it' from 8.00.

So we see the seconde side is : To 10.00 in 07/01/2017, we repeat the same steps as the side number one.

Which lead us to a result which is : To 10.00 in 06/01/2017. So we repeat the phase of the verification by the same steps.

we have the slot in the site of chargement is from 10.00 to 12.00 in 06/01/2017, and our result is To 10.00 in 06/01/2017, so we see that the result and the slot of site of loading are matching.

So the date of 06/01/2017 is the date of loading.

- Assignment :

  This part it's a merging between the day of loading and the logistic capacity.

  The main idea of how it work is : by verify in our tool the constraints that we discussed in the section of logistics.

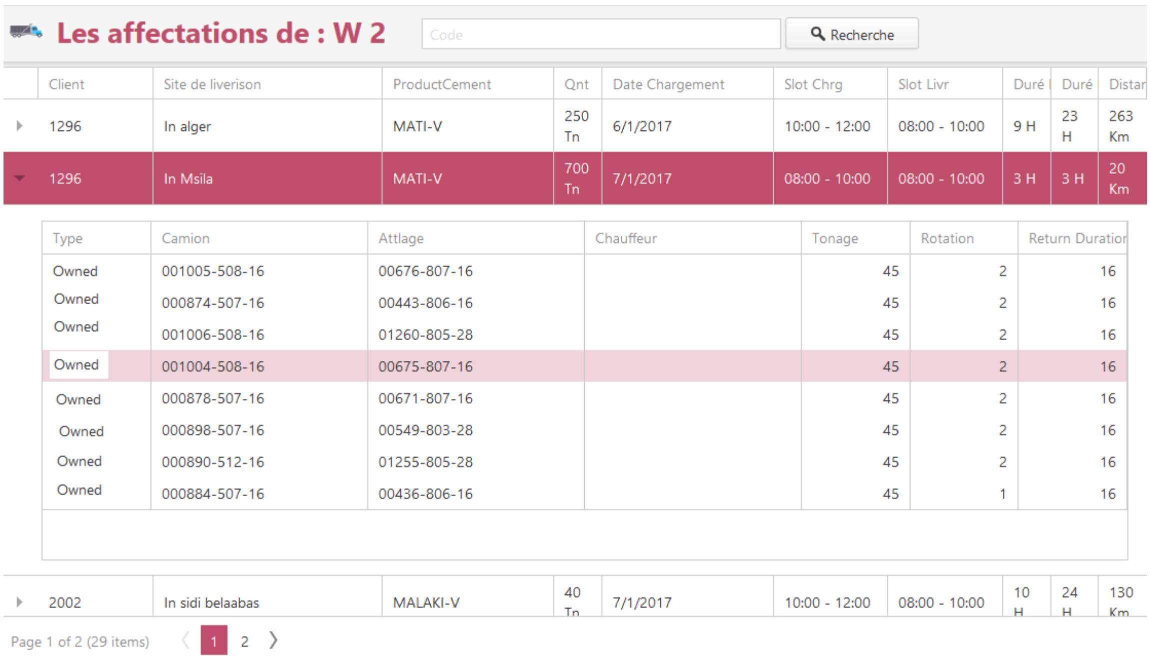
  At this moment we know the day of loading but we must verify the logistic capacity in order to assign every available truck to deliver the products.

  So we verify the status of truck and we see the following constraints :

  1. If the distance from the company to the location of the customer surpass 250 km and the tonnage surpass 20 tons then the owned trucks will start loading and the deliver.
  2. If the distance from the company to the location of the customer it's 250 km or less and the tonnage is 20 tons or less then the available (rented) trucks will start loading and the deliver.
  3. If there's no owned trucks and rented trucks to deliver then the spot trucks will come to load and deliver

  In the next Figure 40 we'll see an example about the assignment :

Figure 40: Assignment Of Trucks



Les affectations de : W 2

Code | Recherche

	Client	Site de liverison	ProductCement	Qnt	Date Chargement	Slot Chrg	Slot Livr	Duré	Duré	Distar
▸	1296	In alger	MATI-V	250 Tn	6/1/2017	10:00 - 12:00	08:00 - 10:00	9 H	23 H	263 Km
▾	1296	In Msila	MATI-V	700 Tn	7/1/2017	08:00 - 10:00	08:00 - 10:00	3 H	3 H	20 Km
▸	2002	In sidi belaabas	MALAKI-V	40 Tn	7/1/2017	10:00 - 12:00	08:00 - 10:00	10 H	24 H	130 Km

Type	Camion	Attlage	Chauffeur	Tonage	Rotation	Return Duratior
Owned	001005-508-16	00676-807-16		45	2	16
Owned	000874-507-16	00443-806-16		45	2	16
Owned	001006-508-16	01260-805-28		45	2	16
Owned	001004-508-16	00675-807-16		45	2	16
Owned	000878-507-16	00671-807-16		45	2	16
Owned	000898-507-16	00549-803-28		45	2	16
Owned	000890-512-16	01255-805-28		45	2	16
Owned	000884-507-16	00436-806-16		45	1	16

Page 1 of 2 (29 items) ‹ 1 2 ›

As we see in this example the customer in the red zone order 700 tons of product, with his location if far from the depot by 20 km and a duration of 3 hours and a slot in [8.00to10.00] and a slot of site of loading in [08.00to10.00].

we see here that the distance is less than 250 KM so suppose that the spot trucks will start loading and deliver but the constraint of the tonnage it's not valid. So our tool will assign the owned trucks to load and start delivering.

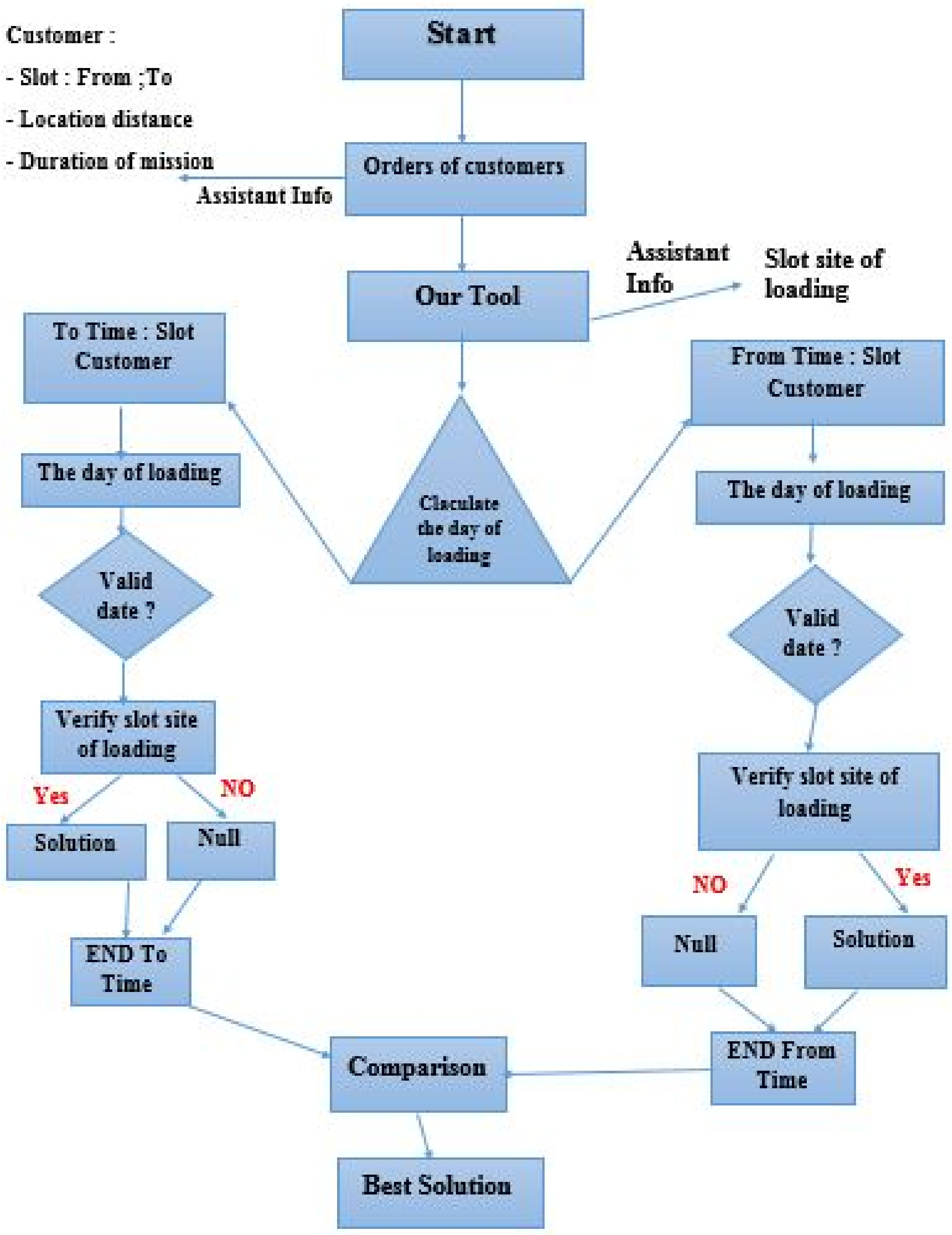
Also we see the rotation section it mean's that the truck will leave to load and deliver for two times ith a return duration to the depot of 16 min.

4.6.2 *Flowchart*

We'll see a flowchart about the two parts to explain more :

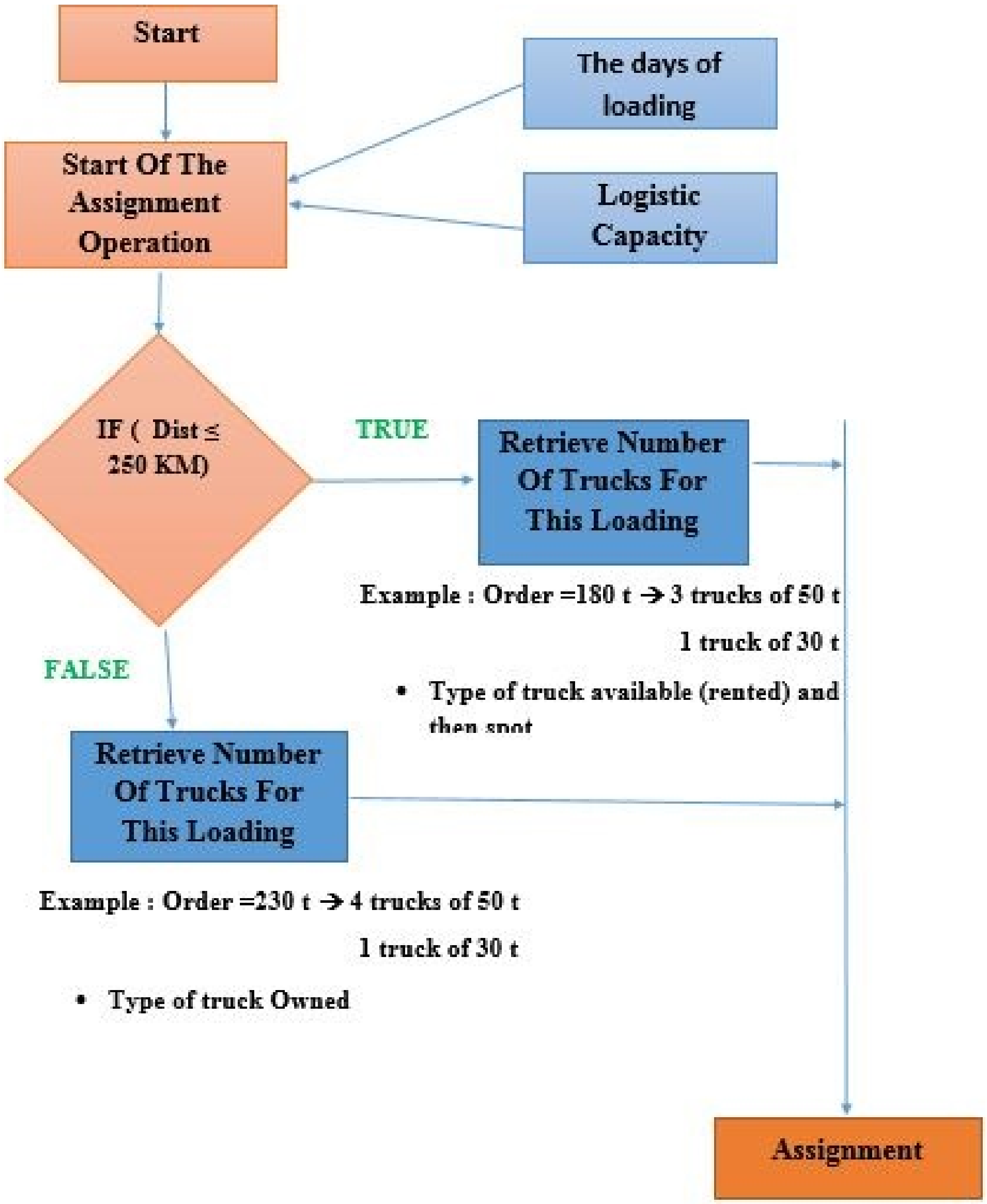
This is how our tool calculate the day of loading, like we'll see in the next Figure 42 :

Figure 41: How To Calculate The Day Of Loading



In the next flowchart we'll see the way of how to assign the trucks :

Figure 42: How To Assign The Trucks


Start
The days of loading
Start Of The Assignment Operation
Logistic Capacity
IF (Dist ≤ 250 KM)
TRUE
Retrieve Number Of Trucks For This Loading
Example : Order =180 t → 3 trucks of 50 t
1 truck of 30 t
• Type of truck available (rented) and then spot
FALSE
Retrieve Number Of Trucks For This Loading
Example : Order =230 t → 4 trucks of 50 t
1 truck of 30 t
• Type of truck Owned
Assignment

4.7 EXPERIMENTS AND RESULTS

In this section a collection of experiments and results are provided. Our experiments was tested on a 1.70 GHz i3 laptop Pc with 4 GB RAM. This machine are running Microsoft Windows 7 operating system. The results obtained from 8 experiments are summarized in the following charts. Our experiments are divided into two parts the planning of routes and assignment.

In this experiments a total of 290 orders were generated with parameter settings (including the number of orders for each customer and time factor and the accuray percentage).

Furthermore, the time factor in each experiment has changed not a lot but a noticeable change.

In the following chart who belongs to the first part the planning of routes ,we'll see that there's an order for each client and the expeiment it's bound with the time. We are measuring the time factor in each experiment in order to know our tool how long it takes to plan the routes when the client orders a product.

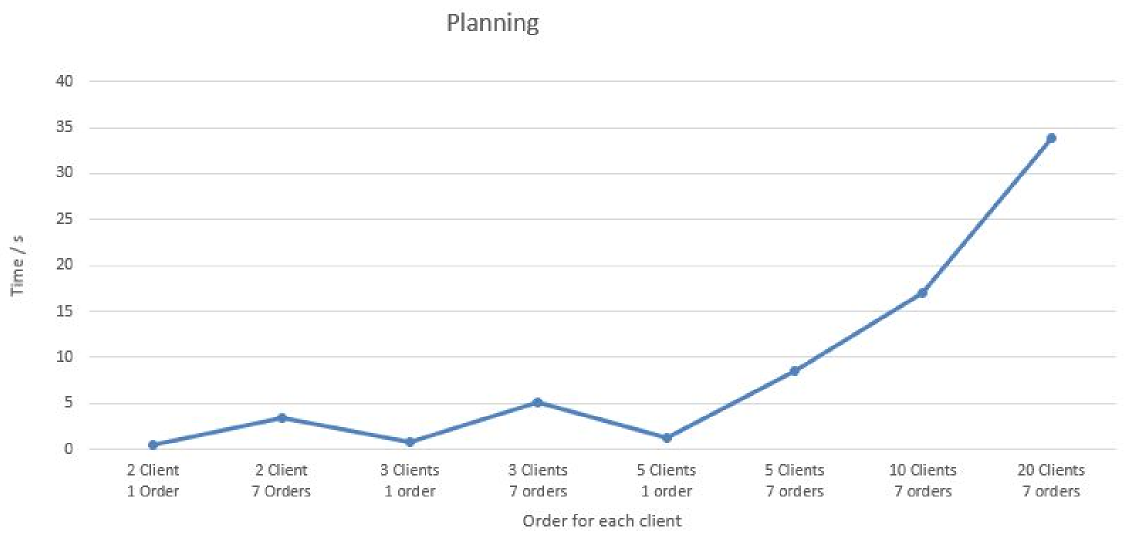
For every customer there's one order per day, and the maximum is 7 orders in the week. There's no limits for the number of the customers.

As show we see that the time is unsteady this is because the time is linked with number of orders not the customers. Which means that if the number of orders is 1 and the number of customers is high, we see that the time factor is small no matter what.

On the other hand, when the number of orders fo each client increased, we see that the time is increasing as well.

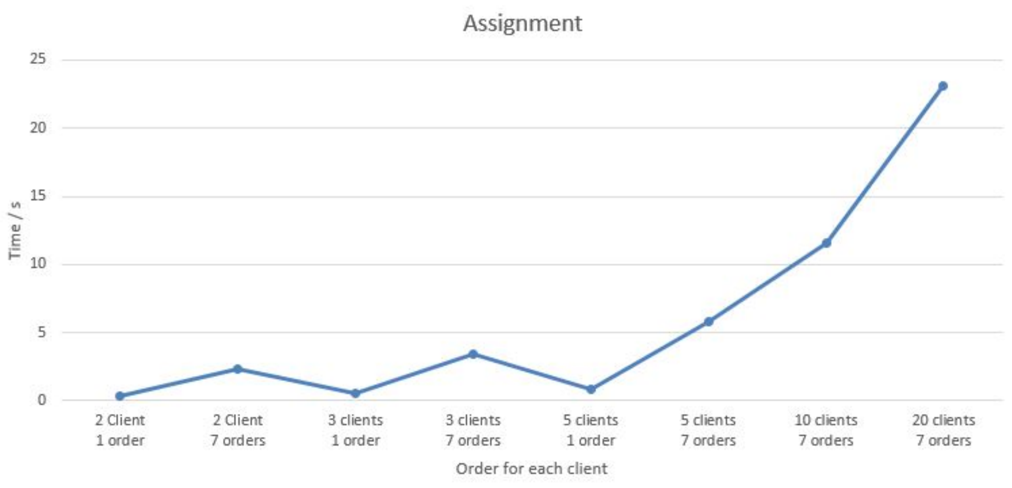
For Example : in the chart we see 2 client with 1 order fo each, have less time than 2 clients with 7 orders for each, the same is for the other experiments. This enhances our analysis which is that the time is linked with the number of orders of each customer not the number of the customers. The next Figure 43 will show us the chart of planning routes that enhances our talk :

Figure 43: Chart Of Planning Of Routes



Moving to the next part which is the assignment, as well as possible we can say that the operation of assignment will take a less time than the part of the planning of routes. That's because the part of assignment it's easier, for every order of a customer our tool will manage to assign every available truck in less time . The next Figure 44 will show us the chart of assignment.

Figure 44: Chart Of Assignment

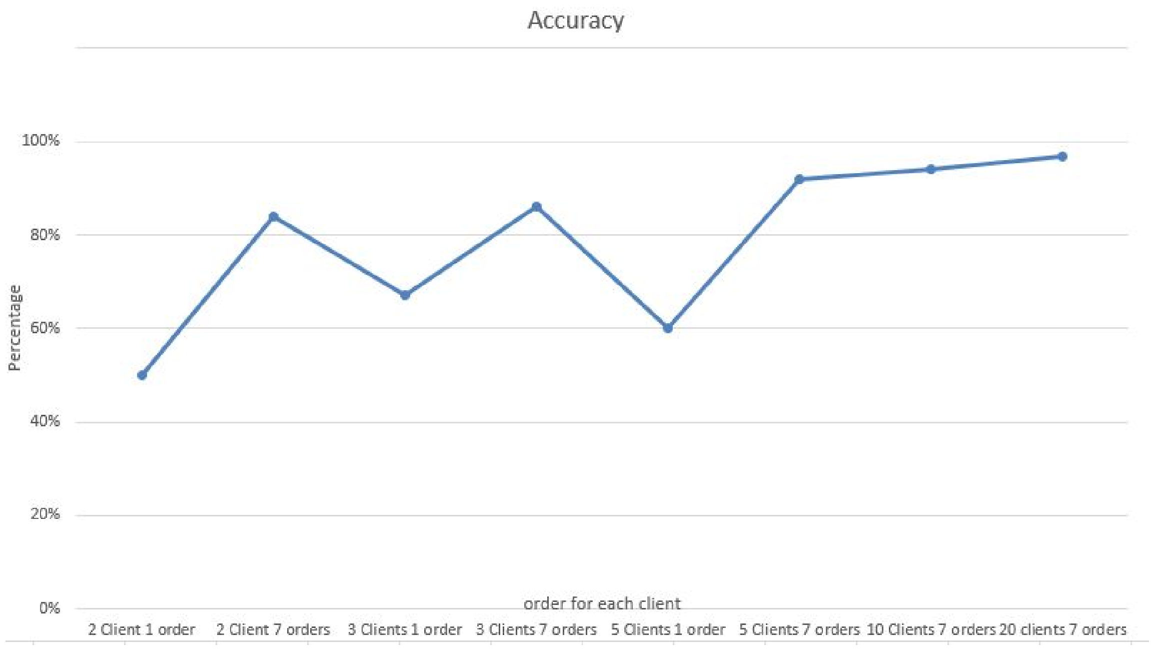


Almost when we look to the two charts, we can say that they are the same, but after a good analysis we see that the maximum time of planning of routes is between 30 and 35 seconds, and the maximum for the assignmet is between 20 and 25 secondes.

By this analysis we can say that the part of planning of routes is the most difficult part.

Now moving to the accuracy of our tool with this experiments,we'll see the chart first than we'll explain :

Figure 45: Chart Of Accuracy



We are in front of the accuracy of our tool within the experiments which we have done , we see that when the number of the orders is one for each client the accuracy is small, but when the number of the orders for each client is high than the accuracy it's rising.

For example : we see that the experiment of 2 client with one order for each the accuracy of the tool is 50 percent which means that we cannot find a solution for he day of loading for one client.

Our tool didn't touch the barrier of the 100 percent accuracy, but she did approached it.

The main cause of our tool didnt touch the 100 percent is that our tool used to work with high number of orders and customers and data. Whenever the number of data and orders is high our accuracy of the tool touch the 100 percent barrier.

4.8 weakness points

Our tool are not perfect, the tool have weakness points but this does not allow us to say that it is not effective. Its effectiveness has been proven by dealing with large data by giving us results with good accuracy.

However, we'll talk about the weakness point. The first weakness it's that our tool when it comes to the part of planning of routes to generate the day of loading, it only calculate on the edges of the slot (time windows) of the client. So the weakness here it's when the calculating of the day of the loading starts From time to time, we can fall into a trap of not finding a solution on the edges of the slot.

Our tool doesnt look in the middle of the interval which is a weakness, because we can found a solution in the middle of the interval.

It will keep us working on find the solution on the edges by dividing it into to parts, this will take so much time and time is money for the companies. Also it will keep us dangling in vicious circles. We cannot even make a move and start looking for a solution into the middle of the interval because our tool cannot do it.

Another weakness which is in a specific slot of day of loading we have a specific amount of product in the site of loading. If we can find a solution of the day of the loading we need to check to and verify the the amount of product in the site of loadind in that specifi slot. If the tonnage in the order of the client surpass the tonnage of product in the site of loading we cannot fulfil the demand of the customer, so here it's a diffuclty for us.

That's the two main weakness points of our tool. In the next section future works we'll see the proposition to make this tool overtake this difficulties.

4.9 future work

we have presented our tool of the vehicle routing problem, we have a proposition which is the application of genetic algorithms.

By the application of the genetic algorithms we assume that we can change our tool to the best, by finding the best possible solution.

The genetic algorithms help us raise and improve the level of our tool by apply the settings of the genetic algorithm.

The GA will help us surpass the weakness points of the searching of a solution in the slot. By apply the GA we can search for the solution in the middle of the interval. This is our main objective in the near future.

4.10 conclusion

In this chapter we have presented our tool, by modelize it and understanding the architecture of it. We present in this chapter the main idea of how the day of loading is calculated and how to assign the trucks. We saw that you cant know the day of loading without knowing the informations of the customer. Also we have presented the main core of the algorithm of the VRP which wil help us optimize this problem. As well we saw the weakness points of our algorithm. Furthermore in future works we'll see the apllication of the genetic algorithm to give us a better solution.

Part IV

CONCLUSION

5 GENERAL CONCLUSION

Vehicle Routing Problem (VRP) was defined as determining the appropriate delivery routes for a series of given customers , so that the vehicles can depart from the distribution center and return to the original center after servicing all customers, with certain constraints (such as vehicle capacity, customer demand, time window, etc.), in order to achieve certain goals (shortest distance, least cost, etc.).

As we have seen previously Vehicle Routing Problems have very important applications in the area of distribution management. VRP is both of theoretical and practical interest (due to its real world applications), which explains the amount of attention given to the VRP by researchers in the past years, and since VRP is an NP-Hard problems.

In chapter one, we carried out an experimental study of heuristic and metaheuristic resolution methods. We discussed the methods of resolution which lead us to the exact and the approximated methods.

The exact methods generally consist in enumerating, implicitly, all the solutions of the search space and guaranteeing an optimal solution. And the approximated methods generally deal with the problems of large size, they do not ensure to find the optimal solution but are effective.

As well we in chapter two we introduced our main problem which is the Vehcile Routin Problem. We saw the variants of VRP, passing by the history,the mathematic fomulation and the origins of it and the simailar problems. Also we saw some related works about the VRP.

Furthermore, in the third chapter we introduced our tool, we saw the modelization of it, and the global architecture. Then we talked about the working of our tool by showing how the tool are working.

The main objective of this contribution is to develop a tool to optimize the vehicle routing problem (VRP).

A collection of experiments and results are provided. Some experiments are done by showing the relation between time and orders of customers. In this experiments a total of orders were generated.

We terminate our dissertation by future works and a proposition. The proposition is to apply a Genetic Algorithm to improve our tool and make it effevctive.

BIBLIOGRAPHY

[1] Md. Faisal Faruque Md. Iqbal Hasan Sarker Abdul Kadar Muhammad Masum, Mohammad Shahjalal. Solving the vehicle routing problem using genetic algorithm. *International Journal of Advanced Computer Science and Applications*, 2, no 7:126–131, 2011. URL https://thesai.org/Downloads/Volume2No7/Paper%2019-Solving%20the%20Vehicle%20Routing%20Problem%20using%20Genetic%20Algorithm.pdf.

[2] Claudia Archetti and Maria Grazia Speranza. The split delivery vehicle routing problem: A survey . *Springer*, pages 103–122, 2008.

[3] Olli Braysy, Geir Hasle, and Wout. Dullaert. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, 159:586–605, December 2004. doi: 10.1016/S0377-2217(03)00435-1.

[4] Tonci Caric and Hrvoje Gold. *Vehicle Routing Problem*. In-Teh, September 2008. ISBN 978-953-7619-09-1. Printed In Croatia.

[5] Jenna Carr. *An Introduction to Genetic Algorithms*, chapter 1, pages 1–40. Jenna Carr, 2014. URL https://www.whitman.edu/Documents/Academics/Mathematics/2014/carrjk.pdf.

[6] V. CERNY. A thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory*, 45, January I985. URL http://www.webpages.uidaho.edu/~stevel/565/literature/tsp.pdf.

[7] Symphony development. URL https://projects.coin-or.org/SYMPHONY. SYMPHONY is an open-source solver.

[8] Surekha P* Dr.S.Sumathi. Solution to multi-depot vehicle routing problem using genetic algorithms. *World Applied Programming*, 1 no 3: 118–131, August 2011. URL https://pdfs.semanticscholar.org/fcff/394938381091eccc1c346373a7b5adccc06b.pdf.

[9] Eliana M. Toro O. Antonio H. Escobar Z. Mauricio Granada E. Literature review on the vehicle routing problem in the green transportation context. *Luna Azul*, no.42, June 2016. doi: 10.17151/luaz.2016.42.21.

[10] Birhanu Beshahâ Eshetie Berhan and Daniel Kitaw. Stochastic vehicle routing problem: A literature survey. *Journal of Information & Knowledge Management,*, 13.no3:12, September 2014.

[11] Abel Garcia-Najera. The vehicle routing problem with backhauls: A multi-objective evolutionary approach. In *Evolutionary Computation in Combinatorial Optimization*, April 2012.

[12] S.M. Sheykhalishahi A. Bavandpouri E. Ashoor G.H. Dastghaibifard, E. Ansari. A parallel branch and bound algorithm for vehicle routing problem. In S.M. Sheykhalishahi A. Bavandpouri E. Ashoor G.H. Dastghaibifard, E. Ansari, editor, *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008*, volume 2, 2008.

[13] Hacene Halim. Etude comparative des methodes heuristiques d optimisation combinatoire. Master's thesis, Universite Mohamed Khider Biskra, June 2013.

[14] Bo Dong Hongying Ya Hongmei Jia, Yang Li. An improved tabu search approach to vehicle routing problem. In Bo Dong Hongying Ya Hongmei Jia, Yang Li, editor, *CICTP 2013*. Elsevier Ltd, 2013.

[15] Manar I. Hosny. Bridging the gap between theory and practice in the vehicle routing research. *Applied Mathematics, Electronics and Computers*, October 2014.

[16] S. M. Johnson. Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, March 1954.

[17] jsprit. URL `https://www.graphhopper.com/open-source`.

[18] Omprakash Kaiwartya, Pawan Tiwari, Sushil Kumar, and Mukesh. Prasad. *Designing and Implementing Global Supply Chain Management*, chapter Chapter 3- Dynamic Vehicle Routing Solution in the Framework of Nature-Inspired Algorithms, pages 36–50. Business Science Reference, IGI Global, 1 st edition edition, December 2015. doi: 10.4018/978-1-4666-9720-1.ch003.

[19] Fred Glover Gary A. Kochenberger. *Handbook Of Metaheuristics*. Springer, 2003.

[20] Wan-Yu Liu, Chun-Cheng Lin, Ching-Ren Chiu, You-Song Tsao, and Qunwei Wang. Minimizing the carbon footprint for the time-dependent heterogeneous-fleet vehicle routing problem with alternative paths. *Sustainability*, 6:4658–4684, July 2014.

[21] J.-Y. Potvin M. Gendreau. *Handbook of Metaheuristics*, chapter 2, pages 41–56. Springer, 2003. URL `https://www.springer.com/us/book/9781441916631`.

[22] Vijini Mallawaarachchi. An introduction to genetic algorithms, July 2017. URL https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3.

[23] R. Marti and G. Reinelt. *The Linear Ordering Problem, Exact and Heuristic Methods in Combinatorial Optimization*, volume 175. Springer, 2011. doi: 10.1007/978-3-642-16729-42. URL https://epdf.tips/the-linear-ordering-problem-exact-and-heuristic-methods-in-combinatorial-optimiz.html.

[24] Professor Scott Moura. Ce 191: Civil and environmental engineering systems analysis. University of California, Berkeley-United States Of America, 2014.

[25] et H. A. Teller N. Metropolis, M. N. Rosenbluth. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21,number6, June 1953. URL http://bayes.wustl.edu/Manual/EquationOfState.pdf.

[26] neo. URL http://neo.lcc.uma.es/vrp/vrp-flavors/capacitated-vrp/.

[27] neopick. URL //http://neo.lcc.uma.es/vrp/vrp-flavors/vrp-with-pick-up-and-delivering/.

[28] Networking and Emerging Optimization. Vehicle routing problem. URL http://neo.lcc.uma.es/vrp/vehicle-routing-problem/.

[29] Sylvain Perifel. *Complexite Algorithmique*. Creative Commons, 2014. Version electronique sous licence Creative Commons.

[30] J. Christopher Beck Patrick Prosser and Evgeny Selensky. Vehicle routing and job shop scheduling: What's the difference? In *ICAPS-03 Proceedings*, 2003. The International Conference on Automated Planning and Scheduling (ICAPS).

[31] Surya Prakash Singh Rajesh Matai and Murari Lal Mittal. *Traveling Salesman Problem, Theory and Applications*, chapter 1, pages 1–24. InTech, 2010. doi: 10.5772/12909.

[32] Mustafa Mohammed Rashid. Tabu search, December 2013. URL https://fr.slideshare.net/mustafarashid1/tabusearch-final.

[33] Roberto & Lucibello Enzo & Maria Gambardella Luca. Rizzoli, Andrea-Emilio & Montemanni. Ant colony optimization for real-world vehicle routing problems. *Swarm Intelligence*, 1:135–151, November 2007. doi: 10.1007/s11721-007-0005-x.

[34] A. Arnout S. Coene and F. Spieksma. The periodic vehicle routing problem: a case study. DEPARTMENT OF DECISION SCIENCES AND INFORMATION MANAGEMENT (KBI), October 2008. URL https://lirias.kuleuven.be/bitstream/123456789/205411/1/KBI_0828.pdf.

[35] SANGEETA(Student) SONIA SHARMA(A.P). Simultaneously pickup and delivery mdvrp with multi objective g.a. *Computer Application*, 5-no4:6, June 2015. URL https://rspublication.com/ijca/2015/june15/23.pdf.

[36] Ze-jun Zhang Shi-hua Zhan, Juan Lin and Yi wen Zhong. List based simulated annealing algorithm for traveling salesman problem. *Computational Intelligence and Neuroscience*, 2016:12, March 2016. doi: 10.1155/2016/1712630. Article ID 1712630.

[37] Brian KallehaugeJesper LarsenOli B.G. MadsenMarius M. Solomon. *Column Generation*, chapter 3, pages 67–98. Springer, Boston, MA, 2005. ISBN 978-0-387-25485-2.

[38] Clifford Stein and R. L. Rivest. T. H. Cormen, C. E. Leiserson. *Introduction to algorithms*. McGraw-Hill Book Company, United States of America, second edition edition, 2001. ISBN ISBN 0-262-03293-7 (hc. : alk. paper, MIT Press).âISBN 0-07-013151-1 (McGraw-Hill).

[39] The OptaPlanner team. *Opta Planner User Guide*, version 6.2.0.cr3 edition. URL https://docs.jboss.org/drools/release/6.2.0.CR3/optaplanner-docs/html/index.html.

[40] Google Optimization Tools. The job shop problem. URL https://developers.google.com/optimization/scheduling/job_shop.

[41] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi. Optimization by simulated annealing. *Science*, 220, May 1983. URL https://pdfs.semanticscholar.org/beb2/1ee4a3721484b5d2c7ad04e6babd8d67af1d.pdf.

[42] Gilbert Laporte Paolo Toth Daniele Vigo. Vehicle routing: historical perspective and recentcontributions. *EURO Journal on Transportation and Logistics*, 2, May 2013.

[43] vrp. *The open-vrp Reference Manual*. URL http://quickref.common-lisp.net/open-vrp.html.

[44] Wikepedia. URL https://en.wikipedia.org/wiki/Vehicle_routing_problem.

[45] Wikipedia. Tsp. URL https://en.wikipedia.org/wiki/Travelling_salesman_problem.

[46] The Microsoft Excel workbook. vrp-spreadsheet-solver. URL http://verolog.deis.unibo.it/vrp-spreadsheet-solver.