REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Technologie de l'Information et de la Communication

Par : Bounif Issam

SUJET

Geolocation web application for vehicule routing problem based on genetic algorithms

Soutenu publiquement le : / /2016 devant le jury composé de :

Mm. Bouzaroura Ahlam	**Université de M'sila**	**Rapporteur**
……	**Université de M'sila**	**Président**
……	**Université de M'sila**	**Examinateur**
……	**Université de M'sila**	**Examinateur**

Promotion : 2015 /2016

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DES MATHEMATIQUES ET DE L'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

MEMOIRE de fin d'étude

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : Technologie de l'Information et de la Communication

Par : Bounif Issam

SUJET

Geolocation web application for vehicule routing problem based on genetic algorithms

Soutenu publiquement le : / /2016 devant le jury composé de :

Mm. Bouzaroura Ahlam	**Université de M'sila**	**Rapporteur**
……	**Université de M'sila**	**Président**
……	**Université de M'sila**	**Examinateur**
……	**Université de M'sila**	**Examinateur**

Promotion : 2015 /2016

Remerciements

Je remercie d'abord et Avant toute chose, ALLAH
Le tout puissant qui m' a donné la volonté et la patience nécessaires pour terminer ce travail.

J'adresse mon profonde gratitude à Madame

Bouzaroura Ahlam.

Je la remercie pour sa gentillesse, et pour tous ses conseils et remarques constructives qui m'ont permis d'améliorer la qualité de ce travail. Je tiens à lui exprimer mon grand respect et du plaisir que j'ai travaillé avec elle.

Comme je remercie le président de jury et l'examinateur pour avoir bien voulu accepter de juger ce travail.

Enfin, je souhaite exprimer toute ma gratitude à tous ceux qui ont contribué par leurs conseils ou leurs encouragements à l'aboutissement de ce modeste travail.

B. Issam

Content

CHAPTER 3 IMPLEMENTATION OF THE SOLUTION TO VRP

Figures list

Tables list

GENERAL INTRODUCTION

In business today companies are undergoing a revolution in terms of implementing new operations strategies and technologies in response to the challenges and demands of the twenty-first century. Businesses have to overcome the challenges of satisfying the demand of customer for products. Customers have become increasingly demanding, expecting ever-higher levels of product and service performance. Gradually, in industry after industry, customers are coming to expect greater customization of products and services to their individual needs.

Increasing competition, complexity, and geographical scope in the business world have led to the supply chain management and continuing improvements in the capabilities of the personal computer have made the optimization of supply chain performance possible. Electronic mail and the Internet have revolutionized communication and data exchange, facilitating the necessary flow of information between the companies in the supply chain.

Supply chain management is a broadened management focus that considers the combined impact of all the companies involved in the production of goods and services, from suppliers to manufacturers to wholesalers to retailers to final consumers and beyond to disposal and recycling.

most companies focused their supply chain initiatives on reengineering supply chain cost structures. These initiatives were driven primarily by corporate restructuring and internal improvement as company strategies responded to the accelerated opening of global markets. Companies that practice supply chain management report significant cost and cycle time reductions. For example, Wal-Mart Stores Inc. announced increases in inventory turns, decreases in out-of-stock occurrences, and a replenishment cycle that has moved from weeks to days to hours

The challenging strategy in the field of supply chain management and logistics industry is to optimize the product delivery from suppliers to customers thus satisfying constraints. the major industry players manage systems to create ever greater value, the decision-making quality is largely related to the ability to solve combined complex problems. Such problems are known

as Vehicle Routing Problems (VRP), in which the vehicles leave the depot, serve customers assigned and upon completion of their routes return to the depot. Each customer is characterized by their own demand. Since the problem is related with only one depot, the VRP is also named Single-depot VRP. In cases with more than one depot, VRPs are known as multi-depot VRPs (MDVRP). VRP problems enhanced with additional features or constraints, dedicated to tactical choices as routing issues over an extended time horizon (. Francis et al 2008); or greater fidelity to reality: legal constraints on driving time (. Goel 2010, Prescott-Gagnon et al 2010), transport axis congestion (. Malandraki and Daskin 1992 Ichoua et al 2003), or finally explicit placement of objects in the load (Iori and Martello 2010).

In VRP exact methods are not suitable to obtain optimal solutions. heuristic algorithms have been adopted to solve the VRPs at a faster rate thus providing computationally efficient solutions. yet, very few methods are now able to solve general way a relatively large set of variants of the VRP. The objective of the problem is aimed at minimizing the total cost of combined routes for a fleet of vehicles. Since cost is associated with distance, in general, the goal is to minimize the distance travelled.

In this thesis, we will try to solve VRP by applying the bio-inspired Genetic Algorithm. After that the solution is implemented through a web application and to increase the challenges we will create an android application for the management of our system .

This thesis constitute a steps to solve vehicle routing problem. The structure of this thesis is the following:

in chapter 1 we present general concepts on web and Android applications. in Chapter 2 we introduce the definition of VRP, its mechanism then the approximate and exact methods. In chapter 03 we propose our solution to VRP algorithm, in addition we present our web and Android application, and the tools used for the implementation. Finally, We conclude by a general conclusion that encapsulate results and difficulties encountered

CHAPTER 1

WEB AND ANDROID APPLICATION

1 Introduction

With the advent of HTML markup language and the unexpected spread of the web, several concepts and terms have invaded the computer field, going static and dynamic sites until the arrival of Web 2.0 with social network sites ... etc. However, despite these developments we need more interactivity and independence when the web is used; Web applications offer us that possibility.

in the following section we will clarify the web application concept, the technology related to it and the difference compared to the traditional application or desktop application. then we will introduce another type of application called *android application* that appeared owing to the spectacular revolution of the telecommunications sector especially from ordinary cell phone to Smartphone that opened the door for the emergence of operating system among them we find the Android.

2 Web Application

Web Applications are a manageable application via a web browser. At the same time and in the same way that web sites, a web applications are typically placed on a server and is handled using widgets, using a web browser, via a computer network (Internet, intranet, local area network, etc.). A Web application is a client-server application, but the reverse is not true. The technologies used to develop web applications are the same as those used in the creation of websites: HTML, XML, Flash, Javascript, Java, PHP, AJAX …etc.

Web messaging, content management systems, are web applications. Search engines, e-commerce software, online games, forum software can be in the form of web application. Devices such as routers network are sometimes equipped with a web application in their firmware .

A web application might be something like Gmail where you have a specific instance of the application that you alone see. Your email and your interaction with the site is completely

separate from that of others. You see the page differently than others do and are able to affect changes to it for yourself and no others. It is interactive in that you can send and receive information in the form of emails, attachments etc.

The main differences of a web application are basically that:

- Each user has a session-based relationship. That means the application is somehow aware of who you are and loads a specific set of variables for your interface.
- Each user can change the interface for their own session. This generally manifests itself in things like themes, colors, organization of elements, etc.
- Users can permanently create, store and change data. This can be as simple as an email message or as complex as a multi-page spreadsheet in a web application like *Google Doc* or even an image in *Pixel* or a video at Animate

2.1 Web server

In the most common technology, the web application is oriented around a web server which is connected application software, sometimes accompanied by a database server. The set is called application server.

The source code for the application software is placed directly in web pages. These pages are stored by the server. When the client requests a page, the web server will search the page, and then executes the statements that it contains. These instructions may appeal to the database server. The web server passes the page with the result of the execution to the client.

The transmission of information between the client and the server is based on the HTTP protocol is also used for web sites. Which allows to use the same client software- a web browser.

Web applications often make use of the cookie mechanism: in response to a query, the server sends tracking information to the client (the cookie). Then the client will return this information during the next request. The mechanism is used to identify the customer and track manipulation.

2.2 Difference entre application desktop and application

A desktop application is an application that runs completely on a single PC [example: Microsoft Word]. Desktop applications have traditionally been limited by the hardware on which they run. They must be developed and installed on a particular operating system, and may have strict hardware requirements that must be met to ensure they are working properly. Application

updates need to be applied by the user directly to their installation, and may require hardware upgrades or other changes to work.

A Web application is an application that uses Web technologies (HTTP, HTML, XML,JSP, ASP, Web services,...) and which is accessed using a browser (FireFox, Google Chrome, IE,...) [Example: Google Maps]. Since web applications are executed via a web browser, developers do not need to develop for multiple platforms. For example, a single application that works in Chrome will work on both Windows and OS X. The developers do not need to distribute software updates to users when the web application is updated. By updating the application on the server, all users have access to the version update.

A client-server application is an application that is installed on one or more PCs but which connects to one (or more) Server (s) that makes its services available.

If the part of the application installed on the PC contains just the GUI as well as a few basic features and most of the features are implemented on the server, referred to as "thin client".

If almost all the features of the application are implemented on the client and the server is that rarely invoked, referred to as "thick-client".

2.3 Technologies related to web application

a web application is that it is programmed in a language that is understandable by a web browser. Since they are applications on the web they must be accessed somehow. The standard interface is through a web browser. Web browsers understand a finite amount of languages: HTML, XML, Flash, Javascript, Java, PHP, AJAX …etc. which means that web applications must be programmed in one of them to be understood. But, you can develop your application with a client that connects to a server without using Web technologies. For example, technology Java "Remote Method Invocation (RMI" = remote method invocation) permits. Other examples: Common Object Request Broker Architecture (CORBA), RPC (Remote Procedure Call).

Web pages may also contain cmdlets. These are pieces of source code that will be executed by the web browser after transmission of the page - unlike the majority of the source code that is executed by the web server before transmission. ActiveX, Java, Adobe Flash and Silverlight are technologies used for applets.

The client software - web browser - is identical to that used to visit a web site. The software needed to run the cmdlets are included in browsers - sometimes in the form of complement

2.4 Rich Internet Application

A rich Internet application or RIA is a Web application that offers similar features to traditional software installed on a computer. Interactive size and execution speed are particularly careful in these Web applications.

An RIA can be:

- run on a Web browser. No installation is required;
- run locally in a secure environment called sandbox.
- The term rich Internet application has been introduced in a publication from Macromedia in March 2002.

– **Sandbox**

In the field of the security of computer systems, a *sandbox* is a mechanism that allows the execution of software(s) with less risk to the operating system. These are often used to run untested code, or dubious provenance.

The term *sandbox* is also used in a broader sense to refer to a test environment for software or Web sites.

In the field of business intelligence, beyond software testing, it can also be question of test data to assess the quality and possible uses, before integrating them into the production warehouse and impose various operating constraints.

2.5 Best known Web application

2.5.1. Google Maps

Google Maps is a free online mapping service. The service was created by Google. A version has been launched in 2004 to the United States and the Canada and in 2005 in Britain (under the name of Google Local). The actually version of Google Maps was launched Thursday 27 April 2006, at the same time in France, Germany, Spain and Italy.

It is a service that allows, from the scale of a country to zoom up the ladder of a street. Fixed shots showing details of some streets are also accessible through a gateway to Google Street View.

Two types of views are available in Google Maps: a plan view classic, with names of streets, neighborhood, city and a satellite image view, which now covers around the world. This service

is no longer in beta since September 12, 2007, and has been added to the links of the Google home page.

2.5.2. Google Earth

Google Earth is a software, owned by Google, allowing visualization of the Earth with an asse mblage of aerial photographs or satellite. This software allows any user to fly over the Earth and zoomin on a location of his choice. According to geographic regions, the information available ar e more or less accurate. Thus an inhabitant of a metropolis can locate his favorite restaurant, get a 3D view of buildings in the metropolis, while the resolution of the
photos of a good portion of the Earth is very low.
Coverage according to Google should improve quickly. Modeling in 3 dimensions of constructio ns , initially usingthe SketchUp software, is now created automatically
using algorithms using in part the shooting Street View and elevation data.

Is the 2007 Google Earth and Maps Director John Hanke.

Since January 20, 2015, Google Earth Pro licenses are free.

In October 2011, Google announced that Google Earth has been downloaded and installed mor e than a billion times worldwide. This makes it the most used GIS (Geographic Information System) in the history .

2.5.3. Gmail:

Gmail is a free email service offered by Google. Messages received on a Gmail account can be read via a mail client or with a web browser. However, many features of the service are available only through the web browser. In January 2012, 425 million Internet users use this service by e-mail electronic.

At April 1, 2004 launch, registration required an invitation. Two years later, the beta was opened to the public. At the time, the initial capacity was 1 GB, and grew steadily until May 13, 2013, date at which Google decides to unify its various services storage spaces. It is possible, by paying to increase its space by inventory . In may 2012, the site was announced as over Hotmail and Yahoo! Mail 3 in number of accounts.

3 Android

Actually, the Android application are the most used applications on Smartphone there are specialized online store to download this application such us: Play Store and Google Apps. An Android Application is a software application running on the Android platform. the Android platform is built for mobile devices, a typical Android app is designed for a Smartphone or a tablet PC running on the Android OS.

3.1 Definition

Android, is a mobile operating system based on the Linux kernel, developed by Google. The system was designed for Smartphone and Tablet PCs, then has diversified in the objects connected and computers as TVs (Android TV), cars (Android Auto), computers (Android - x 86) and smart watch (Android Wear). The system was launched in June 2007 following the acquisition by Google in 2005 by the startup of the same name . By 2015, Android is the most used operating system in the world with more than 80% of market shares in the Smartphone .

3.2 Android Platform

Android is a software platform and operating system for mobile devices, based on the Linux kernel, developed by Google and later the Open Handset Alliance. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries. Applications written in C and other languages can be compiled to ARM native code and run, but this development path is not officially supported by Google.

The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 48 hardware, software, and telecom companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache license, a free-software and open source license.

Google describes Android as:

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation.

As a disruptive addition to a mature field, it's not hard to see why there has been some confusion about what exactly Android is. Android is not:

- A Java ME implementation Android applications are written using the Java language, but they are not run within a Java ME virtual machine, and Java-compiled classes and executables will not run natively in Android.
- Part of the Linux Phone Standards Forum (LiPS) or the Open Mobile Alliance(OMA). Android runs on an open source Linux kernel, but, while their goals are similar, Android's complete software stack approach goes further than the focus of these standards defining organizations.
- Simply an application layer (like UIQ or S60) While it does include an application layer, Android also describes the entire software stack encompassing the underlying operating system, API libraries, and the applications themselves.
- A mobile phone handset Android includes a reference design for mobile handset manufacturers, but unlike the iPhone, there is no single "Android Phone." Instead, Android has been designed to support many alternative hardware devices.
- Google's answer to the iPhone : The iPhone is a fully proprietary hardware and software platform released by a single company (Apple), while Android is an open source software stack produced and supported by the Open Handset Alliance and designed to operate on any handset that meets the requirements. There's been a lot of speculation regarding a Google-branded Android phone, but even should Google produce one, it will be just one company's hardware implementation of the Android platform. [1]

3.3 Hardware

First and foremost, Android is a software stack for mobile devices. This means that high on the list of priorities is the preservation of battery power and the efficient management of limited memory resources. There are five distinct layers to the Android system stack:

• The Acorn RISC Machine (ARM) Linux core forms the solid base upon which all the other layers stand. Linux is a proven technology that is highly reliable, and the ARM processor family is known for high performance on very low power requirements.

• The libraries provide the reusable and sharable low-level code for basic functions such as codecs — software for coding and decoding digital sound and video — functions for the presentation of rich graphics on a small displays, secure shell support for encrypted TCP/IP

traffic into the cloud, as well as component support for Web browsing (WebKit), SQL database functionality (SQLite), and standard C library functionality you would expect in a Linux system.

• The Dalvik run-time byte-code interpreter, which strongly resembles the Java™ language byte-code interpreter, adds a few distinct features that uniquely define the security and power-preserving model of Android. Every application currently running, for example, has its own user ID and its own copy of the interpreter running to strictly separate processes for security and reliability.

• The Android application framework enables you to use and replace components as you see fit. These high-level Java classes are tightly integrated components that define the Android API.

• The Android core applications include the WebKit browser, Google calendar, Gmail, Maps application, SMS messenger, and a standard e-mail client, among others. Android applications are written in the Java programming language, and you can download many more from the Android market on the fly.

Android is not a single piece of hardware; it's a complete, end-to-end software platform that can be adapted to work on any number of hardware configurations. Everything is there, from the boot loader all the way up to the applications. And with an Android device already on the market, it has proven that it has what it takes to truly compete in the mobile arena. [1]

3.4 Android the Operating System(s)

Android uses Linux for its device drivers, memory management, process management, and networking. However you will never be programming to this layer directly.

The next level up contains the Android native libraries. They are all written in C/C++ internally, but you'll be calling them through Java interfaces. In this layer you can find the Surface Manager (for compositing windows), 2D and 3D graphics, Media codecs (MPEG-4, H.264, MP3, etc.), the SQL database (SQLite), and a native web browser engine (WebKit).

Next is the Android runtime, including the Dalvik Virtual Machine. Dalvik runs dex files, which are converted at compile time from standard class and jar files. Dex files are more compact and efficient than class files, an important consideration for the limited memory and battery powered devices that Android targets.

Figure 1.1 Android Operating System

The core Java libraries are also part of the Android runtime. They are written in Java, as is everything above this layer. Here, Android provides a substantial subset of the Java 5 Standard Edition packages, including Collections, I/O, and so forth.

The next level up is the Application Framework layer. Parts of this toolkit are provided by Google, and parts are extensions or services that you write. The most important component of the framework is the Activity Manager, which manages the life cycle of applications and a common "back-stack" for user navigation.

Finally, the top layer is the Applications layer. Most of your code will live here, along side built-in applications such as the Phone and Web Browser. [1]

3.5 Network Connectivity

It supports wireless communications using:

• GSM mobile-phone technology • 3G • Edge • 802.11 Wi-Fi networks

3.6 Security

Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications. Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform, and per- URI permissions for granting ad-hoc access to specific pieces of data. [1]

3.7 Security Architecture

A central design point of the Android security architecture is that no application, by default, has permission to perform any operations that would adversely impact other applications, the operating system, or the user. This includes reading or writing the user's private data (such as contacts or e-mails), reading or writing another application's files, performing network access, keeping the device awake, etc. An application's process is a secure sandbox. It can't disrupt other applications, except by explicitly declaring the permissions it needs for additional capabilities not provided by the basic sandbox. These permissions it requests can be handled by the operating in various ways, typically by automatically allowing or disallowing based on certificates or by prompting the user. The permissions required by an application are declared statically in that application, so they can be known up-front at install time and will not change after
that. [1]

3.8 Advantages of Android

There are already many mobile platforms on the market today, including Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiMo), and more. When I tell people about Android, their first question is often, Why do we need another mobile standard? Where's the "wow"?

Although some of its features have appeared before, Android is the first environment that combines the following:

• A truly open, free development platform based on Linux and open source: Handset makers like it because they can use and customize the platform without paying a royalty. Developers like

it because they know that the platform “has legs” and is not locked into any one vendor that may go under or be acquired.

• A component-based architecture inspired by Internet mashups: Parts of one application can be used in another in ways not originally envisioned by the developer. You can even replace built-in components with your own improved versions. This will unleash a new round of creativity in the mobile space.

• Tons of built-in services out of the box: Location-based services use GPS or cell tower triangulation to let you customize the user experience depending on where you are. A full-powered SQL database

let’s you harness the power of local storage for occasionally connected computing and synchronization. Browser and map views can be embedded directly in your applications. All these built-in capabilities help raise the bar on functionality while lowering your development costs.

• Automatic management of the application life cycle: Programs are isolated from each other by multiple layers of security, which will provide a level of system stability not seen before in smart phones. The end user will no longer have to worry about what applications are active or close some programs so that others can run. Android is optimized for low-power, low-memory devices in a fundamental way that no previous platform has attempted.

• High-quality graphics and sound: Smooth, antialiased 2D vector graphics and animation inspired by Flash are melded with 3D accelerated OpenGL graphics to enable new kinds of games and business applications. Codecs for the most common industry- standard audio and video formats are built right in, including H.264 (AVC), MP3, and AAC.

• Portability across a wide range of current and future hardware: All your programs are written in Java and executed by Android’s Dalvik virtual machine, so your code will be portable across ARM, x86, and other architectures. Support for a variety of input methods is included such as keyboard, touch, and trackball. User interfaces can be customized for any screen resolution and orientation.

Android offers a fresh take on the way mobile applications interact with users, along with the technical underpinnings to make it possible. But the best part of Android is the software that you are going to write for it. This book will help you get off to a great start.[2]

3.9 Latest android version

Android *Marshmallow* is the version 6.X of the mobile operating system Android developed by Google. It is available since October 2015.

Marshmallow focuses primarily on the improvement of the overall user experience introduced by Android *Lollipop* and brings some additional features.

' The features highlighted in this new version are the introduction of a system of authorization for applications (access to contacts, location,...), profound connections between applications (including the "Chrome Custom Tabs"), contactless payment via NFC (not available in France at the moment), the native support of the recognition of fingerprints as well as options to improve autonomy as feeding Doze (sleepiness) that puts the device in deep sleep when it is not manipulated by the user and ' Standby app"which made Hibernate rarely used applications.

On the side of the user interface, one can note an improvement in the volume control, a new sharing menu, an improvement of text selection and the introduction of more successful than total "priority/silence" mode "do not disturb" mode of Lollipop.

4 Conclusion

In this chapter we explain all of the Web application and the difference between him and desktop application, as we talked about the latest technology related to web application , as we mentioned the most important Web applications in the world .We talk about the famous and the most important operating system in the world from all sides Android .

CHAPITRE 2

VEHICLE ROUTING PROBLEM AND ITS SOLUTION

1 Introduction

The challenging strategy in the field of supply chain management and logistics industry is to optimize the product delivery from suppliers to customers thus satisfying constraints. Such problems are known as Vehicle Routing Problems (VRP), in which the vehicles leave the depot, serve customers assigned and upon completion of their routes return to the depot.

Distribution undertakings must deliver and collect packages in order to satisfy customers. This activity can be modeled as routing problem vehicles, where certain goods are to be delivered from a deposit to clients, while others must be collected on customer to be deposited to the deposit. In this chapter, the studied problem is the routing of the vehicle which we will call (vrp).

We give a detailed description of the components and settings as well as the Mathematical formulations of the problems of conventional vehicles (VRP) tours And list some of its variants.

We conclude the chapter by the presentation of the algorithmic complexity of the problems of vehicle tours.

2 Definition of vehicle routing problem

The VRP (Vehicle Routing Problem) was proposed for the first time by Lord Hamilton in 1859 [3], and then reintroduced by Dantzig and Ramser in 1959 [4]. Since then, the VRP has been the subject of intensive study. The vehicle routing problem is that a classic extension M - PVC PVC the basic version of the VRP reads as follows: a fleet of vehicles, based in one or more deposit (s), must ensure tours between several clients (or cities) having requested a certain commodity or service. All clients visited by a vehicle means the tour of it and each tour begins and ends at the depot. Each client must be serviced once and only once and by one vehicle. The

objective of the VRP is to minimize the sum of distances or the total journey time vehicles tours while satisfying the demand of the customers.

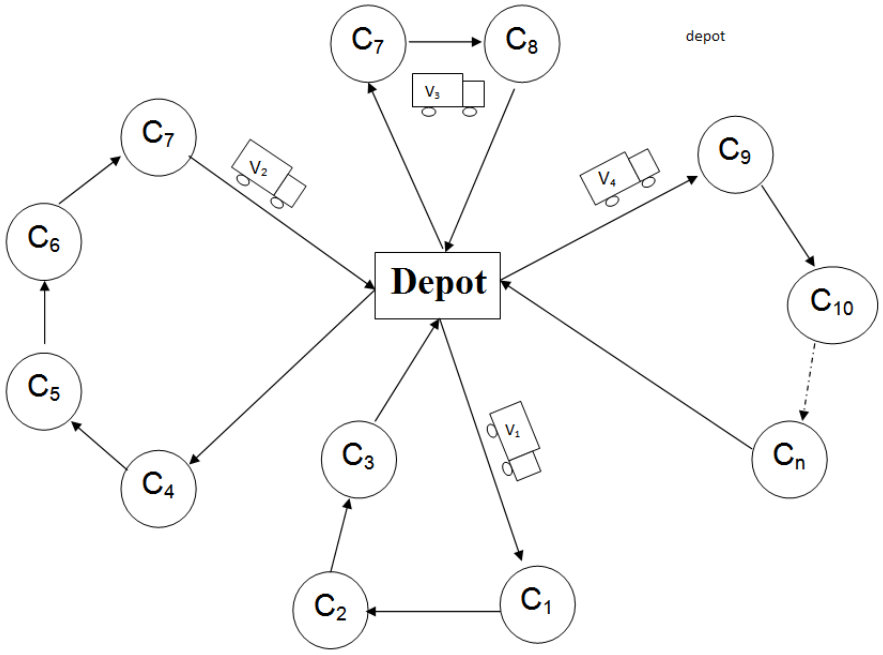

Figure 2.1 VRP

3 The VRP settings

The VRP is characterized by the network that constitutes, the customer to serve, the fleet of vehicles and the objective function.

3.1 Network

The transport network allows the movement of individuals, freight, or information flows. It is somehow the skeleton of a system to establish a form of communication. It can be summarized in the form of a graph full, symmetrical or asymmetrical. The vertices represent customers, characterized by its geographical position *(coordinates (x,y))* and the edges represent paths connecting different clients. Under certain constraints, this graph can be oriented.

3.2 clients:

The customer is characterized by its application that may be a request for a service or products (goods), these products can be of one or several types. The total demand of customers for a same tour must not exceed capacity *Q* of the vehicle. Also, it is characterized by its position in space.

Finally the application may be deterministic (amount requested by the customer is fixed and known by the Distributor) or uncertain (stochastic).

3.3 Fleet of vehicles

The first criterion of the fleet is its size (the number of vehicles the component), the second is its homogeneity (the vehicles are characterized by the same capacity and the same cost of transport) or heterogeneity (vehicles have payload capacity and/or different transport costs).

3.4 Objective function

The most common objectives are either the minimization of the number of used vehicles or the minimization of the total distance travelled by vehicles (**Figure 2.2**). Other objectives may be considered:

- Minimization of the total time of the course of the tour, the waiting time, delay time, service time
- Minimization of the number of vehicles.
- Minimization of the total cost of the tour, fixed cost to the depreciation of the equipment (vehicle or otherwise), salary of drivers, vehicles and in variable cost expenses cited penalties related to the delays for the VRPTW
- Maximization of the gain generated by the tour in the case of collection of products with customers.
- Maximization of quality of service.
- Maximization of loading of vehicles used for tours.

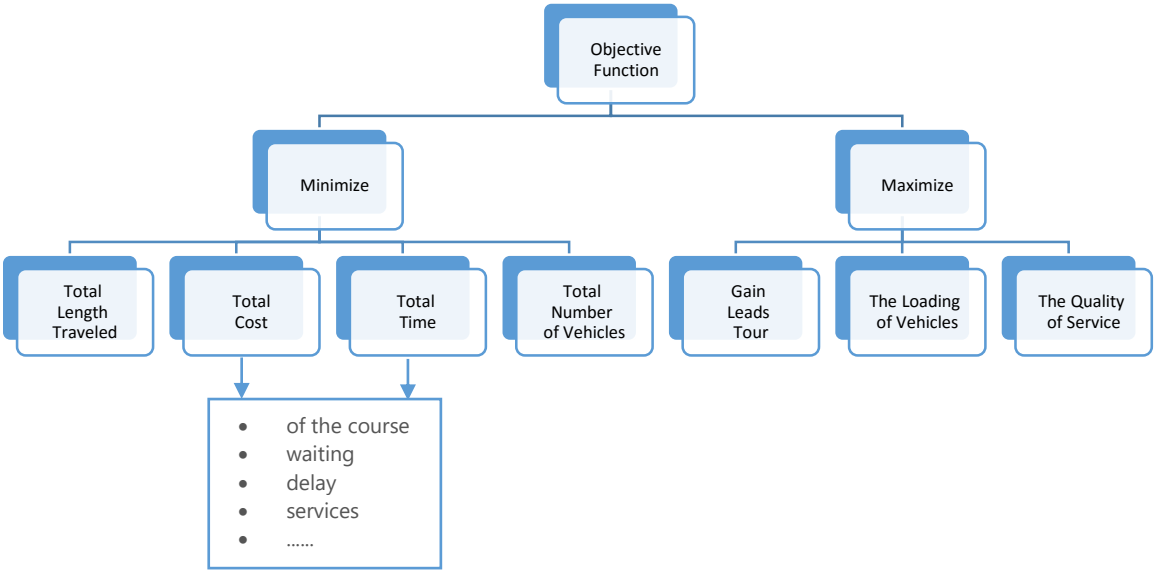

Figure 2.2 Different Objective Functions

4 mathematical Formulation of the VRP

For many formulations of the VRP exist in literature. The common point of all of these formulations is the representation of the problem of vehicle in the form of a directed graph.

The graphic representation of the classic VRP is described as follows: $\mathcal{G} = (\mathcal{X}, \mathcal{U})$graph-oriented where:

$\mathcal{X}$: all the vertices of the graph $\mathcal{G}$, represents clients $\mathcal{N}$and (s) deposit (s) O of VRP

$\mathcal{U}$: all the arcs of the graph G, represents the paths connecting clients between them and the filing of the VRP

n: node

$$\mathcal{X} = \mathcal{N} \cup \{O\} \qquad et \qquad (|\mathcal{X}| = n + 1)$$

$$(|\mathcal{U}| = ((n + 1)n)/2)$$

- Weighting can be performed on vertices (resp. edges) to set the amount requested by the client (dir. distance between two clients or time to travel...)

Mathematical formulation

For many mathematical formulations of the classical VRP exist in the literature. Both formulations presented below are among the busiest.

Before the formulations, it is necessary to set the following parameters:

n: number of clients.

m: number of vehicles ($m \leq n$).

Q_k: capacity of the k^{th} vehicle.

T_k: maximum time for the tour of the k^{th} vehicle.

as_j: number of outgoing arcs of the j Summit.

aej: number of incoming arcs of the j Summit.

qj: the j^{th} client request $d_0 = 0$.

skj: the service life of the j^{th} client by the k^{th} vehicle ($s_o^k = o$).

tij: time for the i^{th} client to client j^{th} ($t_{jj}^k = +\infty$).

dij: distance between i^{th} client of the j^{th} client ($d_{jj}^k = +\infty$).

C_{ij}: the cost of the relocation of the i^{th} client to the client j^{th}.

First Formulation

The mathematical formulation of the classical VRP most commonly used in the literature is the one adopted by Laporte 1992, Rego and Roucairol 1994, Toth and Vigo 2001a, Crainic and San Diego 2006. Indeed, it requires the definition of n × n variable binary, to three indices, following decision:

$$x_{ij}^k \begin{cases} 1 \text{ if the arc (i, j) is traversed by } k^{th} \text{ vehicule} \\ 0 \text{ else} \end{cases} \qquad (1.1)$$

In other words:

$$x_{ij}^{k} \in \{0,1\} \; ; i = 0, \dots, n \; ; j = 0, \dots n; k = 0, \dots, m \qquad (1.2)$$

The problem models as follows:

$$\text{Minimiser} \sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{\mathrm{j}=1}^{\mathrm{n}} C_{ij} \sum_{k=1}^{m} k_{ij} (1.3)$$

within

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} \sum_{k=1}^{m} x_{ij}^{k} = 1; \; j = 2, \cdots, n (1.4)$$

$$\sum_{\mathrm{j}=1}^{\mathrm{n}} \sum_{k=1}^{m} x_{ij}^{k} = 1; \; i = 1, \cdots, n (1.5)$$

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} x_{ip}^{k} - \sum_{j=1}^{n} x_{pj}^{k} = 0; \; k = 1, \cdots, m; \; p = 1, \cdots, n (1.6)$$

$$\sum_{\mathrm{j}=1}^{\mathrm{n}} q_i (\sum_{k=1}^{m} x_{ij}^{k}) \leq Q_k; \; k = 1, \cdots, m (1.7)$$

$$\sum_{\mathrm{i}=1}^{\mathrm{n}} s_i^k x_{ij}^{k} + \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij}^{k} x_{ij}^{k} \leq T_k; \; k = 1, \cdots, m (1.8)$$

$$\sum_{j=1}^{n} x_{0j}^{k} \leq 1; \; k = 1, \cdots, m (1.9)$$

$$\sum_{i=1}^{n} x_{i0}^{k} \leq 1; \; k = 1, \cdots, m (1.10)$$

$$\sum_{\mathrm{i}\in s}\sum_{j\in s} x_{ij}^{k} \leq |S|-1; pourtoutS \in N; et\ 2 \leq |S| \leq (n-1). \quad (1.11)$$

The formula (1.3) represents the function to optimize or the objective function of the classical VRP.

Generally, the goal is to find the minimum of the overall cost of the tours.

Formulas of (1.4) (1.18) represent the problem forced:

The constraints (1.4) and (1.5) ensure that each customer is served once and only once

and by one vehicle. The constraint (1.6) ensures the continuity of the tour:

not only a vehicle must pass once and only once at a customer (1.4) and (1.5)

But it must leave it once the service is completed. The constraint (1.7) ensures that the capacity of the vehicle is not exceeded. The constraint (1.8) assures that the total duration of a tour will never exceed the maximum total duration. (1.10) and constraints (1.9) ensure no excedance of the availability of each vehicle, a vehicle out of the depot and returned only once. The last constraint (1.11) ensures the elimination of sub tour.

Second formulation

It is proposed by Fisher and Jaikumar 1978 and language 1981, is based primarily on two categories of binary variables, one to three indices and one to two indices:

$$x_{ij}^{k} \begin{cases} 1 \text{ If the arc (i j)is traversed by} k^{th} \text{vehicule} \\ 0 \text{ else} \end{cases} \quad (1.12)$$

$$y_{ij}^{k} \begin{cases} 1 \text{ If i}^{th} \text{ client is served by } k^{th} \text{ vehicule} \\ 0 \text{ else} \end{cases} \quad (1.13)$$

The previous formula is equivalent to:

$$y_{i}^{k} \in \{0,1\};\ i = 1,\dots,n;\ k = 1,\dots,m \quad (1.14)$$

The problem models as follows:

$$\text{Minimiser} \sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{\mathrm{j}=1}^{\mathrm{n}} C_{ij} \sum_{k=1}^{m} k_{ij} (1.15)$$

Within :

$$\sum_{k=1}^{n} y_{ik} = 1;\ i = 1,\cdots,n (1.16)$$

$$\sum_{k=1}^{m} y_{0k} = m (1.17)$$

$$\sum_{i=1}^{n} q_i y_{ik} \leq Q_k;\ k = 1,\cdots,m (1.18)$$

$$\sum_{j=1}^{n} x_{ij}^{k} = \sum_{k=1}^{m} x_{ij}^{k} = y_{ik};\ k = 1,\cdots,m\ ;\ i = 1,\cdots,n\ (1.19)$$

$$\sum_{i\in s}\sum_{i\in s} x_{ij}^{k} \leq |S| - 1;\ 8S \in \{2,\cdots,n\}\, etk = 1,\cdots m (1.20)$$

The formula in (1.15) is equivalent to the formula (1.10), it represents the function to optimize. Formulas of (1.16) (1.20) represent the constraints of the problem. The constraint (1.20) is equivalent to the constraints (1.11) and (1.12), it ensures that each customer is served once and only once per one and a single vehicle. The constraint (1.17) ensures that all vehicles lie at the depot. The constraint (1.18) ensures that the capacity of each vehicles is not exceeded. The constraint (1.19) allows to link the two types of constraints on the one hand, and on the other hand, it ensures that the vehicle in leaves the customer immediately that it has been served. The last constraint (1.12) ensures the elimination of sous-tour.

Note (1.3)

The previous formulation represents the classical VRP to which it has added two constraints: one of the vehicles carrying capacity and a total time of each vehicle.

5 Resolution of VRP and Meta heuristics methods

If exact resolution methods used to obtain a solution with optimality is guaranteed, in some situations, it can however seek solutions of good quality, without guarantee of optimality, but for the benefit of a reduced calculation time we use meta heuristics. Heuristics or the metaheuristics generally exploit random processes in the exploration of the search space to cope with the

combinatorial explosion caused by the use of exact methods. In addition to this stochastic basis, the metaheuristics are often iterative, so the same process is repeated during resolution. Their main interest is precisely their ability to avoid local minima in admitting a degradation of the objective function during their progression.

Combinatorial Optimization (OC) occupies a very important place in operations research and computer science. Many applications can be modelled in the form of a problem of combinatorial optimization (POC) such as the problem of salesman, scheduling of tasks, etc. (POC) consists of a finite set of solutions, where each solution must satisfy a set of constraints relating to the nature of the problem, and an objective function to evaluate each solution found. The optimal solution is the one the value of the lens is the most short (resp. large) in the case of minimization (resp. maximization) among the set of solutions.

A combinatorial optimization problem can be defined by:

-Vector of variables $x = (x_1, x_2,...,x_n)$.

-Domain of variables $D = (D_1, D_2,...,D_n)$ where the (D_i) $i = 1$,..., n are finite, sets

-Set of constraints,

-An objective function f to minimize or maximize,

-Set of all solutions achievable potential is $S = \{x = (x_1, x_2,...x_n) \in D / x$ satisfied all constraints$\}$, all S is also called a research area.

(POC) resolution is to find the best solution, defined as the overall best solution or a global optimum.

Solving Combinatorial problems is quite tricky because the finite number of feasible solutions generally increases with the size of the problem, as well as its complexity. This led researchers to develop many methods of resolution in operational research (or) and artificial intelligence (AI). These approaches to resolution can be classified into two categories: approximations and exact methods.

The exact methods were allowed to find optimal solutions for problems of reasonable size and generally have difficulties facing large size applications. On the other hand the approached methods do not guarantee to find an exact solution, but only an approximation.

5.1 Exact methods

The exact methods are based on the use of algorithms that lead safelyTo the optimal solution. The essential principle of these methods is to enumerate the set of solutions of the search space in an implicit manner. Despite the important time of calculation that require, generally, these approaches, several methods have been developed.

For vehicle routing problem, exact methods effectively resolve problems ranging up to 50 clients. In 2001

A method solving a problem containing 100 customers was proposed in Ralphs 2002 [5].

These methods include:

- Branch and Bound method
- Branch and cut method
- Dynamic programming
- other exact methods

5.2 Approached methods

Approached or approximate resolution methods are usually used where the exact methods fail. Indeed, an exact solution requires to browse all of the search space, which becomes impossible when one wants to solve big problems. In this case, a partial execution of the exact algorithm rarely to obtain a good quality solution. Approximate resolution methods have been developed to quickly provide solutions of good quality but not optimal.

5.2.1. Heuristics

A heuristic is an algorithm that provides a feasible, not necessarily optimal, solution to an NP-hard optimization problem quickly (in polynomial time). Saw that exact methods restrict the number of possible clients in the problems and involve, in most cases, a calculation time important, the development and the use of heuristics have proved very useful. These methods allow to manage problems of large sizes with resolution times and acceptable results.

The heuristics that deal with the VRP, we include:

- the group first, second; the road
- The road first, second; the Group
- The algorithm in petal.

5.2.2. Metaheuristics

Faced with the difficulties encountered by the heuristics to have a feasible solution of good quality for hard optimization problems, metaheuristics have emerged. These algorithms are more complete and complex than a simple heuristic, and generally provide a very good quality solution for problems from the areas of operations research and engineering which it does not know of methods effective to treat or when the resolution of the problem requires a high time or a large memory storage.

The relationship between the execution time and the quality of the solution found a Metaheuristic remains then in the majority of interesting cases compared with the different types of approaches to resolution.

Most metaheuristics use random and iterative processes as a means of gathering information, to explore the search space and to cope with problems like combinatorial explosion. A Metaheuristic can be adapted to different types of problems, whereas a heuristic is used to a given problem. Many of them are often inspired by natural systems in many areas such as: Biology (evolutionary algorithms and genetic) Physics (simulated annealing), and also ethology (Ant Colony algorithms).

One of the issues in the design of metaheuristics therefore facilitate the choice of a method and setting the parameters to fit a given problem.

Metaheuristics can be classified in many ways. One can distinguish those who work with a population of solutions from those who handle only one solution at a time. Methods that attempt to iteratively improve a solution are called local search methods or methods of trajectory. These methods construct a path in the space of solutions by attempting to move towards optimal solutions. Examples of these methods are: searching taboo and simulated annealing. Genetic algorithms, Particle Swarm Optimization and Ant Colony algorithms are the best-known examples of the methods that work with a population.

6 Genetic algorithms

Genetic algorithms (GA) are stochastic optimization algorithms based on the mechanisms of natural selection and genetics. They have been adapted to optimize by John Holland (Holland1975) [28], also the work of David Goldberg contributed to enrich them Goldberg 1989[6], (Goldberg 1989) [7].

The vocabulary used is the same that of the theory of evolution and genetics, we use the term individual (potential solution), population (ensemble), genotype (a representation of the solution), gene (a part of the genotype), parent, child, reproduction, crossover, mutation and generation, etc.

Their operation is extremely simple, you start with a population of potential solutions (chromosomes) initial, arbitrarily chosen. Evaluating their relative performance (Fitness). On the basis of these performances is created a new population of potential solutions using simple evolutionary operators: selection, crossing and mutation. A few individuals reproduce, others disappear and only those individuals most suited are supposed to survive. It starts this cycle until it finds a satisfactory solution. Indeed, genetic inheritance through generations allows the population to be adapted and therefore meet the criterion for optimization, **Figure 2.3** shows the main steps of a genetic algorithm. A genetic algorithm searches the extrema of a function defined on space data. Its implementation requires:

- **Data encoding**

The first step is to define and properly encode the problem. This step associated with each point in the search space a specific database, called genotype or set of chromosomes, structure that will characterize each individual in the population. The coding of each individual in sequence is essential in the development of a genetic algorithm which depends on including the implementation of operators of transformations. Thus, this phase determines the data structure that will be used to encode the genotype of individuals in the population. The encoding must therefore be adapted to the problem treated.

Several types of encodings are used in the literature, the first theoretical results on genetic algorithms have opted for a coding by a binary sequence of fixed length through the notion of schema Goldberg 1989 [6]. The efficiency of genetic algorithm depends on the proper choice of the type of encoding.

- **Generation of the initial population**

The generation of the initial population, i.e. the choice of starting devices which we are going to evolve, this choice of the initial population of individuals strongly determines the speed of the algorithm. However, random initialization is easier to achieve: the values of the genes are drawn randomly according to a uniform distribution. However, it may be useful to guide the initial generation to sub interesting fields of the search space. For example when searching of optima in

a problem of optimization under constraints, it is preferable to produce satisfying the constraints. The initial population must be sufficiently diverse and large enough so that the research can browse state space in a limited time.

The genetic algorithm steps are:

1. input: Instance of the problem
2. output: A solution
3. initialization of the Population;
4. assessment of the population;
5. until (Condition of judgment not satisfied)
6 selection;
7 crossing;
8 mutation;
9 replacement;
10. end tan that

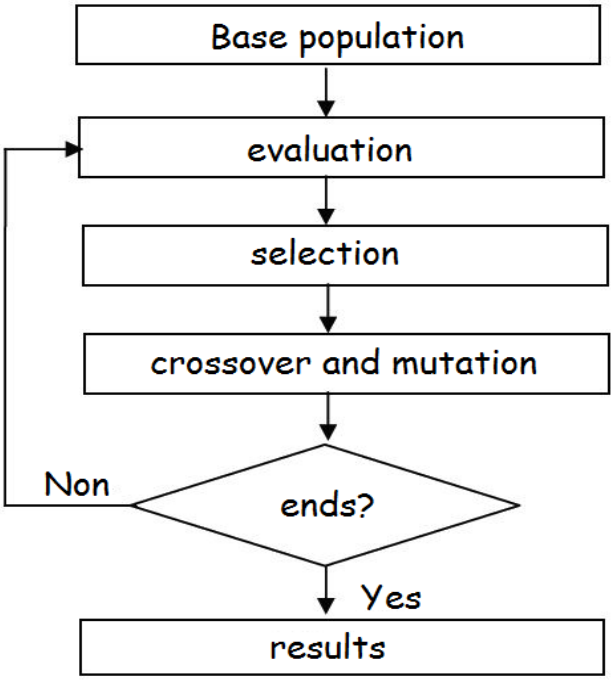

Figure 2.3 Operation of a genetic algorithm

- **Adaptation function (Fitness)**

The Fitness assessment is usually the step in which measured the performance of each individual. To be able to judge the quality of an individual and thus compare to others, must establish a common measurement evaluation. There is no rule to set this function, its calculation can be, whether it's a simple equation or a function affine. The easiest way is to ask the function of adaptation as the formalization of the optimization criterion.

- **Selection**

The selection allows to statistically identify the best individuals of a population and eliminate the bad, during the passage from one generation to another, this process is based on the performance of the individual. The selection operator must be designed to also give a chance to the bad elements, because these elements can, by crossover or mutation, generate relevant offspring compared to the optimization criterion. There are different selection techniques, we propose some.

- uniform Selection: there is no interest in fitness and be selected in a random and uniform manner such that each individual i has the same probability $Prob\ (i) = 1 / T_{pop}$ as all other individuals (T_{pop} is the size of the population).
- Selection by tournament: two individuals are selected at random, comparing their features of adaptation and the best suited is selected.
- Elitism: this method of selection designed to encourage the best individuals of the population. It is therefore the most promising individuals who will contribute to the improvement of our population. It can be seen that this method induces a premature convergence of the algorithm.
- Roulette selection: the selection of individuals by the method of roulette draws the lottery wheel on which each individual is represented by a proportional to its fitness area. It spins the wheel and select an individual. Best evaluated individuals have statistically more likely to be born-selection, but also gives a possibility to people ill-suited to be selected. Individual i a probability is associated with being selected proportional to its fi adaptation:

$Prob\ (i) = f_i / \sum f_j$ *where* $\sum f_j$ means the sum of the adjustments to the population.

- Remains stochastic without replacement: this selection combines the selection by roulette and deterministic selection. A number mini-evil of representatives of every individual among prospective parents is determined in advance by his adaptation, then the population will be supplemented by a random prints. For an individual i, the number of representation in the future generation is given by E (fi), where E is the integer part and fi means the adaptation of i reported the average of adaptations of all individuals. The next generation is then supplemented by the method of selection by wheel such that the evaluation of an individual is given by the stochastic rest

  RI = fi − E (fi).

- **Crossing**

Crossing operator promotes exploration of the search space and enhances the diversity of the population by manipulating the structure of chromosomes, crossing made with both parents and generates two children, hoping that one of the two children at least will inherit good genes from both parents and will be better suited them. There are several methods of crossing for example the crossing at a point (Pc), or multiple-point (Pm) see figures 2.4 and 2.5.

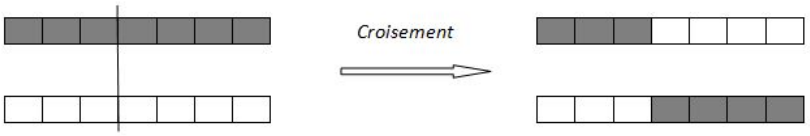

Figure 2.4: Crossing at a point.

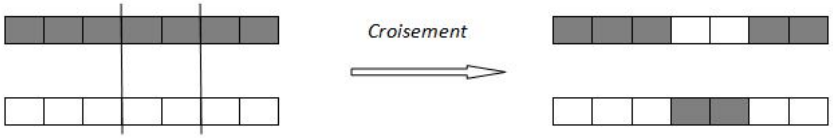

Figure 2.5: crossover two points

- uniform crossover: it is the generalization of the Exchange principle introduced by the crossing at a point. It indeed makes the exchange of each element with a fixed probability. The set of parameters is therefore reduced to the data of this probability.

- Crossing MPX: the MPX (Maximal Preservative X) crossing, has been proposed by Mulhenbein 1993 [8] to solve the traveling salesman problem. The idea of this operator is to insert a part of the chromosome of a parent in the chromosome from the other parent in such a way that the resulting chromosome either as close as possible to his parents while avoiding duplicates.

- **Mutation**

The mutation operator is a process where a minor change in the genetic code applied to an individual to introduce diversity and thus avoid falling into local Optima. This operator is applied with a probability Pm generally smaller than the cross Pc, see figure 2.6

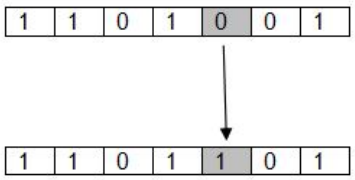

Figure 2.6 Schematic Representation of a mutation in the case of a binary encoding.

The efficiency of genetic algorithms depends on the setting of the different parameters characterizing these algorithms, and which are sometimes difficult to determine. Parameters such as the size of the population, the maximum number of generations, the probability of mutation pm, and the probability of crossing pc.

The first two parameters is directly dependent on the nature of the problem and its complexity, and their choice must represent a compromise between the quality of the solutions and execution time.

The probability of crossing pc is linked to the form of the function devaluation. In general, his choice is heuristic. The higher its value, as the population undergoes important changes.

The probability of mutation pm is generally low because a high rate may lead to a local optimum. On the other hand, a low probability to ensure a good exploration of the search space without disrupting the convergence.

The success of genetic algorithms also depends on the way of coding of individuals. In Combinatorial problems, this encoding is suggested by the nature of the problem, which can lead to better results.

7 Conclusion

In this chapter we have presented the problem of the VRP. From the fields of application and formulation mathematics, and we have listed variants. But the problems of tour vehicles belong to the difficult NP class. Their resolution can only be approximate way, to this day, there is no algorithm for their resolution in polynomial time. We conclude that metaheuristics are a class of methods approached adaptable to a variety of combinatorial optimization problems and lead to relevant results. But there are quite few contributions to understand the reason for this performance, and no evidence can show that a Metaheuristic will be more effective than another on a given problem. Some metaheuristics have the advantage of being easy to implement, such as the case of simulated annealing, others are well adapted to the resolution of certain classes of problem, very forced, such as Ant Colony system. The quality of the solutions found by metaheuristics depends on their settings, and the balance between a scan of the entire space of solutions and a local exploration (the intensification). So, Meta heuristics are methods adapted to each problem addressed, with however the downside to not have in return for any information about the quality of solutions obtained are applied. In the next chapter we will applied a solution for the VRP based on genetic algorithms.

CHAPITRE 3

IMPLEMENTATION OF THE SOLUTION TO VRP

1 Introduction

In this chapter we will explain our approach to solve Vehicle Routing Problem .we opted to use genetic algorithms as a meta heuristic. VRP can be seen like a variant of Traveling Salesman Problem, given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? In our problem we have the depot, the vehicles, clients and other constraints. So, our solution is based on the work of researchers on genetic algorithms of the last decades. After that, we will implement a location web and an Android application that hold up the solution of VRP.

In the following section we will describe and enlarge the solution based on genetic algorithms, the geolocation web application and the Android application.

2 Genetic algorithms for the VRP

Genetic algorithms have been applied successfully to the Traveling Salesman Problem(TSP), and were therefore naturally adapted also to the resolution of the problem of routing of vehicles in its common variants.

These algorithms are particularly efficient on the VRP and have produced some of the best known solutions on the tests of problems [9].

Adaptations of genetic algorithms to the VRP are so numerous that the state of the art drawn by (Haj-Rachid et al. ,2010) [10] are 5 types of coding, 2 methods of creation of the initial population, 3 methods of assessment of individuals, 5 types of selection parents, 22 types of crosses, 6 types of mutations, 4 strategies for the construction of the next generation, and 3 types of criteria of judgment.

Our goal is not to exhaustively detailing all adaptations, we simply two among the best of them (Crainicand San Diego, 2006) [11]: it's adaptations of (Baker and Ayechew ,2003) [12] and

(Prins ,2004) [13] applied to the VRP (CVRP) capacity constrained. To illustrate, we will use the example of figure 3.1 (VRP having n = 20 customers).

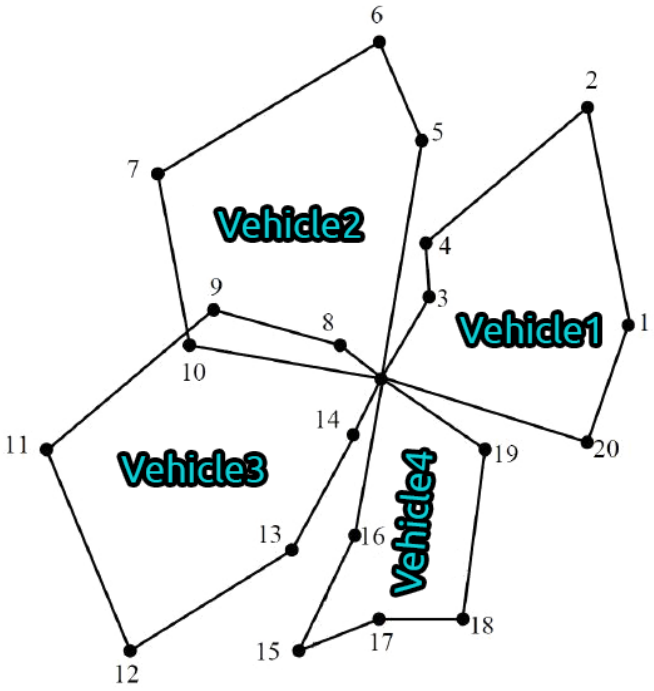

Figure 3.1:VRP having n = 20 customers

2.1 The initial population

The initial population is generated by random insertion methods, seeking to promote the diversity of the population to better explore the space of solutions.

- **Encoding**

An example of direct encoding of a solution of VRP is to represent it in the form of several sequences of clients each of which corresponds to a tour.

Delimiters, called separators tours, can be inserted between the sequences in question for separate tours.

For example, figure 3.1 solution can be represented by the following chromosome:

0	20	1	2	4	3	**0**	5	6	7	10	**0**	8	9	11	12	13	14	**0**	16	15	17	18	19	**0**

Table 3.1: Table of initial chromosome

The tour separator used is the 0 (depot) symbol and allows to distinguish between the beginning and the end of each tour.

The order of customers is important because it determines the direction in which tours are conducted.

By removing the filing of this chromosome, one obtains a second chromosome corresponding to an indirect encoding of the same solution:

20	1	2	4	3	5	6	7	10	8	9	11	12	13	14	16	15	17	18	19

Table 3.2: Second chromosome table

Indeed, this representation can be seen as a single meta-tour of a trade traveler. It certainly indicates the order of customers on the tours, but does not know the beginnings and endings of these. Therefore, an additional procedure of cutting is used. It is generally based on the constraints of the problem such as the capacity of the vehicles or the time windows clients.

(Prins, 2004) [13] uses this second type encoding because encodings with tours separators have not proved to be very effective because of repair procedures which they must use.

The cutting procedure (called Split) applied to a chromosome consists in solving a shortest path problem (Haj-Rachid and al.,2010) [10].

These first two encodings as cycles correspond to a second road first-cluster approach. A second first-route cluster approach would result in a type assignment (Crainic and San Diego, 2006) [11]

It is thus an indirect encoding since it gives no information on the order of tours within the scores of customers.

(Baker and Ayechew 2003) [12] use this encoding, and represent the solution of **Figure 3.1** with the following chromosome:

1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1

Table 3.3: third chromosome table

2.2 Genetic operators

A genetic algorithm has several genetic operators that can be modified to improve the performance of particular implementations. These operators include parent selection, crossover and mutation. Selection is one of the important operations in the process.

- **Selection procedure:**

Selection is one of the important operations in the genetic algorithm process. There are several methods of selection of chromosomes, we have used:

✓ **The selection by rank**

Rank selection first sorts the population by fitness. Each chromosome is associated a ranking based on its position. The worst chromosome will be rank 1, the next 2, and so on until the best chromosome which has rank N (for a population of N chromosomes). The selection by rank of a chromosome is the same as roulette, but the proportions are in relation with the rank rather than the value of the assessment. With this method of selection, all chromosomes have a chance of being selected. However, it leads to a slower convergence towards the right solution.

This is because the best chromosomes differ not much of the worst.

2.3 Intersection operator

To illustrate the crossover operators, we will use the solution in figure 3.1 as the first parent (P1), and another solution as a second parent P2 (n = 20 clients and resolved with m = 4 vehicles problem).

Operators to perform at two points, between positions6-7 and 15-16 of the chromosome.

(Baker and Ayechew, 2003) [12] use an operator classic crossover at two points, identical to that of figure 3.2. The crossover between the two solutions of figure 3.3 is done this way:

The chromosomes of the two solutions children inherit each of part of those of the parents. This legacy is carried out as follows: child E1 (respectively E2), the elements between the points of intersection are inherited from the parent P1 (respectively P2) and the remaining elements P2 (respectively P1)

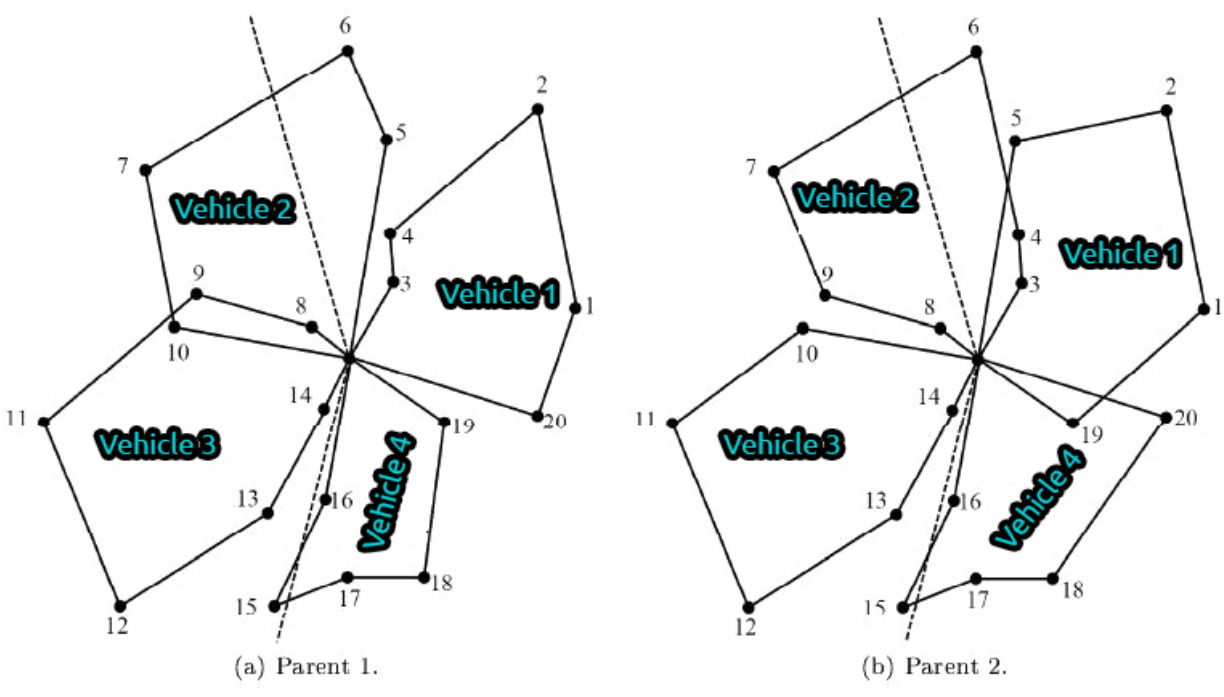

Figure 3.2: Two solutions used as parents to illustrate the intersection of Baker and Ayechew operator (2003).

P2 (respectively P1) :

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1 :	1	1	1	1	2	2	2	3	3	2	3	3	3	3	4	4	4	4	4	1
P2 :	1	1	2	2	1	2	2	2	2	3	3	3	3	3	4	4	4	4	1	4
							⇓													
E1 :							2	3	3	2	3	3	3	3	4					
E2 :							2	2	2	3	3	3	3	3	4					
							⇓													
E1 :	1	1	2	2	1	2	2	3	3	2	3	3	3	3	4	4	4	4	1	4
E2 :	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	1

Figure 3.3: Crossover of Baker and Ayechew (2003).

The chromosome of the E2 solution, we know that the vehicle 1 (respectively 2.3, 4) ensures the tour between customers {1, 2, 3, 4, 20} (respectively {5, 6, 7, 8, 9}, {10, 11, 12, 13, 14}, {15, 16, 17, 18, 19}). The missing information is the order in which to perform these tours: a VRP is

then solved for each tour. The resulting tours are drawn in figure 3.4.

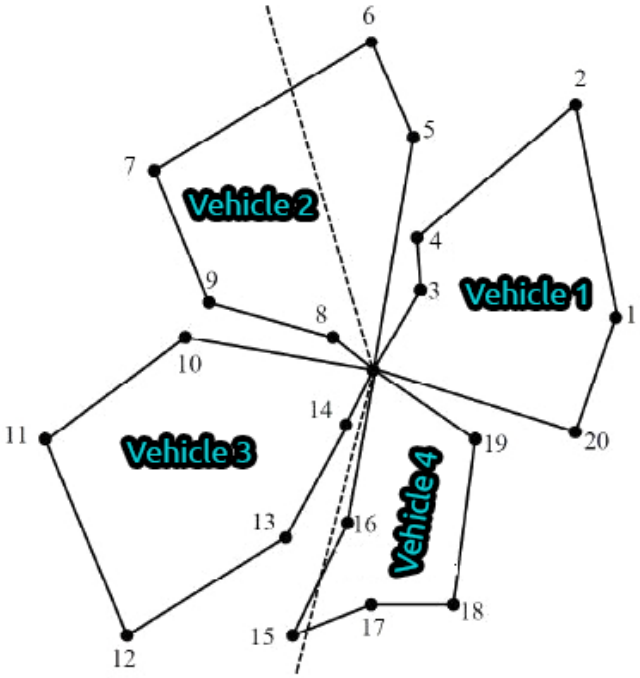

Figure 3.4: Child E2 generated by the intersection of the parents of **Figure 3.2** [Baker and Ayechew, 2003].

We will use an alternative parent P2 to illustrate the operator of intersection (Prins, 2004); its chromosome is as follows:

17	18	19	20	1	3	4	2	6	5	8	7	9	10	11	14	13	12	15	16

Table 3.4 : Fourth chromosome table

Principles of intersection are the following:

1. for the **E1** child (respectively **E2**), the elements between the points of intersection are inherited from the parent **P1** (respectively **P2**),.

2. the remaining elements of the **E1** child (respectively **E2**) are also inherited from the parent **P1** (respectively **P2**) but are not copied as what (otherwise**E1** (resp**E2**)) would be identical to **P1** (resp. **P2**)). They are copied according to their order of appearance in the other parent **P2** (resp. **P1**) from the position just after the second crossover point.

In the example, the elements between the points of intersection in **E2** are in highlighted ingrey. The remaining elements are {20, 1, 2, 4, 3, 5, 16, 15,17, 18, 19} and are highlighted in dark gray. These elements are outlined in the second parent P2. From the 16th position (located just after the second crossover point), these items appear in **P2** in order

{15, 16, 17, 18, 19, 20, 1, 3, 4, 2, 5} and are added to **E1** in this same order of appearance:

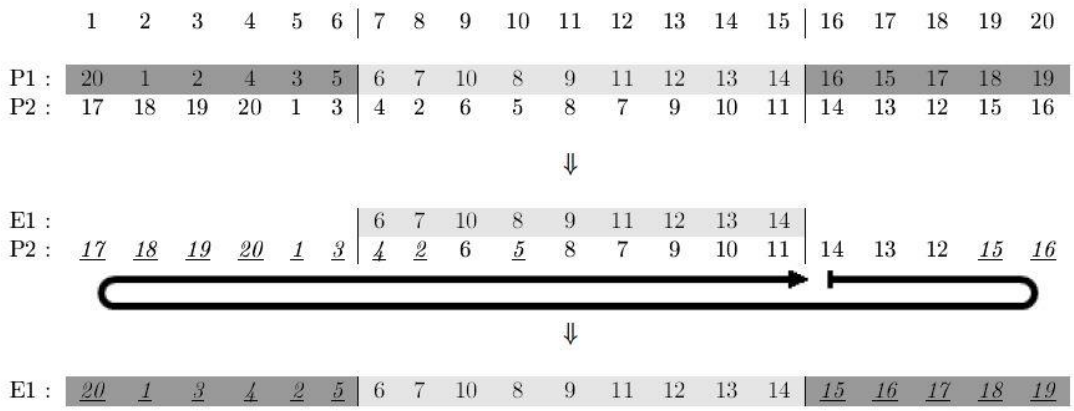

Figure 3.5: Crossover of chromosomes 1

Similarly, the elements added to the **E2** child are{15, 16, 17, 18,19, 20, 1, 3, 4, 2, 5}, obtained chromosome is then:

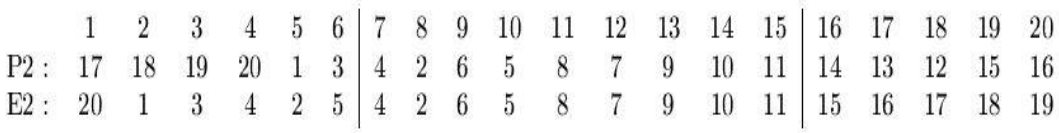

Figure 3.6: Crossover des chromosomes 2

Note that Linear Order Crossover (LOX), another classic, said operator operates similarly to OX: after copying elements between the two points of intersection, the fill remaining elements starts at 16thposition (just after the second crossover point) is done according to their order of appearance in the other parent. However, in LOX, this order is calculated from the first position, and not the 16th.

The application of LOX to both treated parents gives previously that the child **E1**, the remaining elements are copied in the order {17, 18, 19, 20, 1, 3, 4.2, 5, 15, 16}.

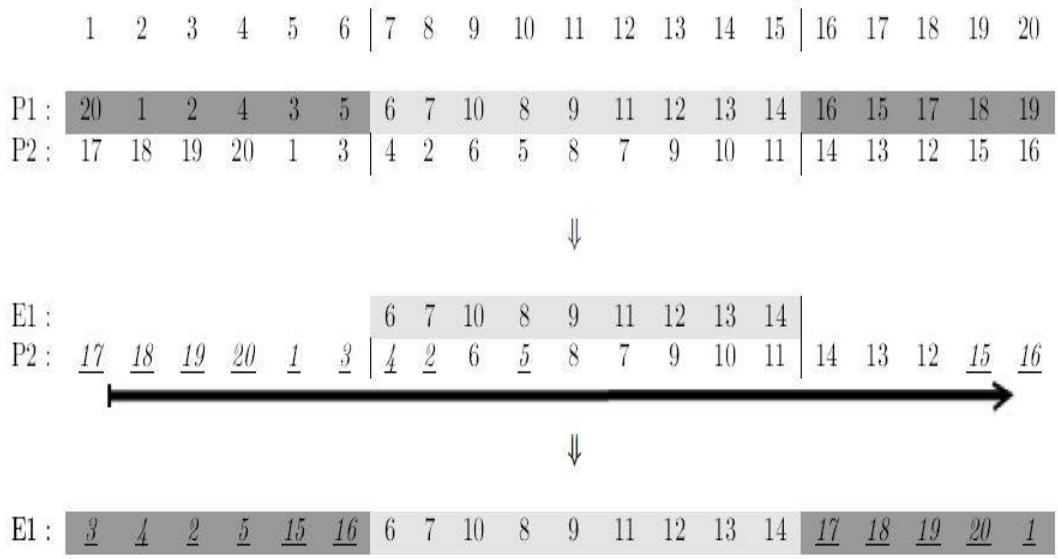

Figure 3.7: Crossover of chromosomes 3

Despite their simple principle, the two operators from cross OX and LOX are the most used in the literature (Haj-Rashid, 2010) [7]. (Prins ,2004) [36] chose OX because it has proved more effective than LOX in experiments.

2.4 Mutation operator

Baker and Ayechew (2010) uses a classical mutation operator based on the random Exchange of two clients between two tours. And we

2.5 initial population

Both approaches use simple heuristics to generate the initial population; such as the method of the economies and the scanning method. Haj-Rashid et al. 2010 [14] identify two other ways to initialize genetic algorithms: generate totally random population, or combine the random generation and the use of heuristics.

2.6 Fitness

Most often, the evaluation function (fitness) is calculated from a weighted sum of multi-objective (method aggregation).

Berger and al. 1998 [15], for example, proposed the formula

$$\text{Fitness} = \mathbf{R} - \boldsymbol{R_{min}} + \frac{\boldsymbol{min(D, 2D_{min})}}{\boldsymbol{Dmin}}$$

Where:

R is the number of innings in the solution to be evaluated,

$R_{min}$is the number of innings in the best solution in the current population

D is the total distance travelled in the solution to evaluate

$R_{min}$is the total distance travelled in the best solution of the current population.

This formulation of the objective function involves the best sought solutions have little tours and small total distances.

The formulation of the fitness proposed by (Sushil and Rilun ,1999) [16] allows to take into account the time windows

$$Fitness = w_1 * D + w_2 * T + w_3 * R_1 + w_4 * R_2 * w_5 * O\backslash$$

Where:

Dis the total distance travelled by vehicles,

T is the total time of the different tours,

$R_1$is the total delay of vehicles among customers,

$R_2$is the total delay of the deposit vehicles,

O is the total vehicle capacity overload and the $w_i$are weight 10 (coefficients) weighting of these objectives.

3 UML Diagrams of our web application

In the following section we will introduce UML essential Diagrams of our web application, this application is created to administrate our solution, where the manager can notices the client request and lunch our solution to satisfied them,

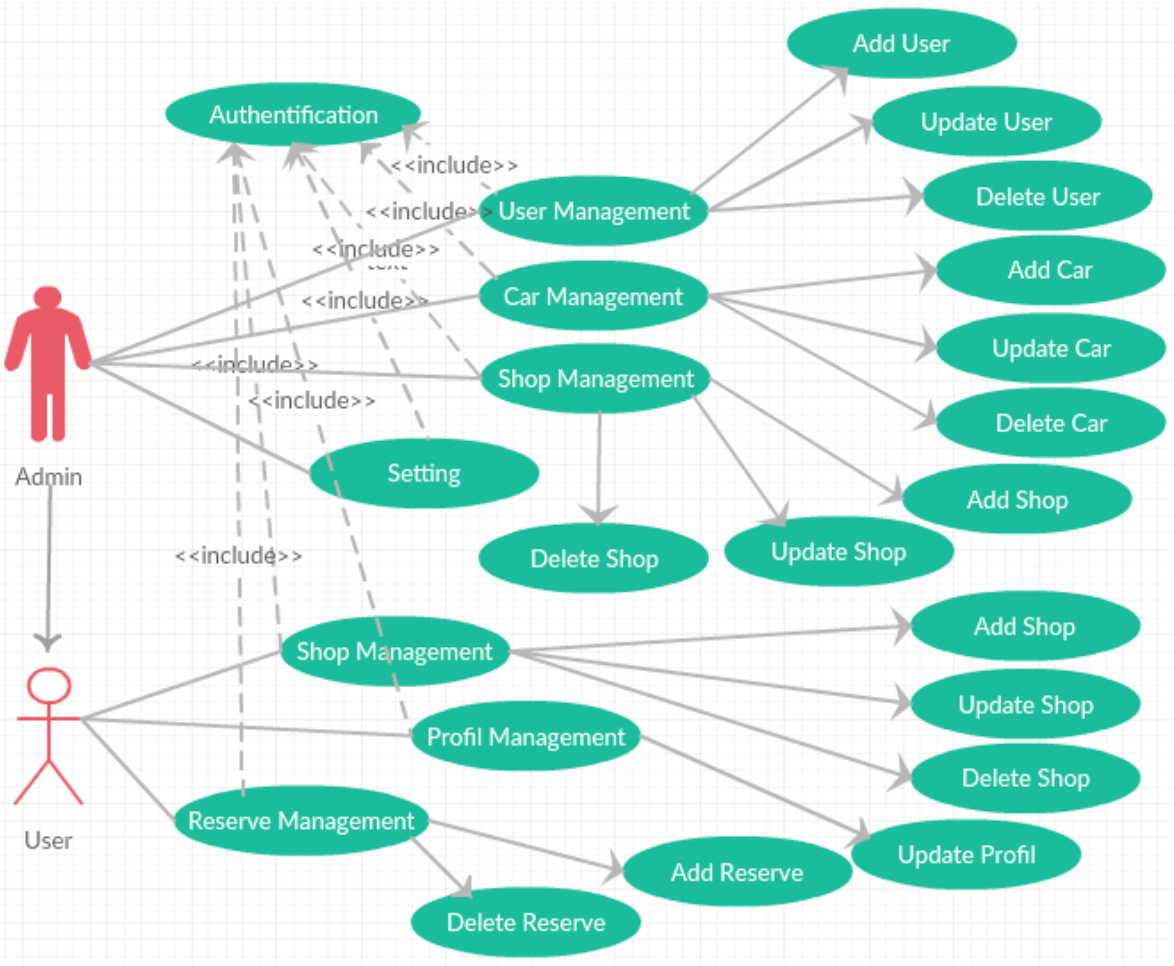

Figure 3.8: Use case diagram

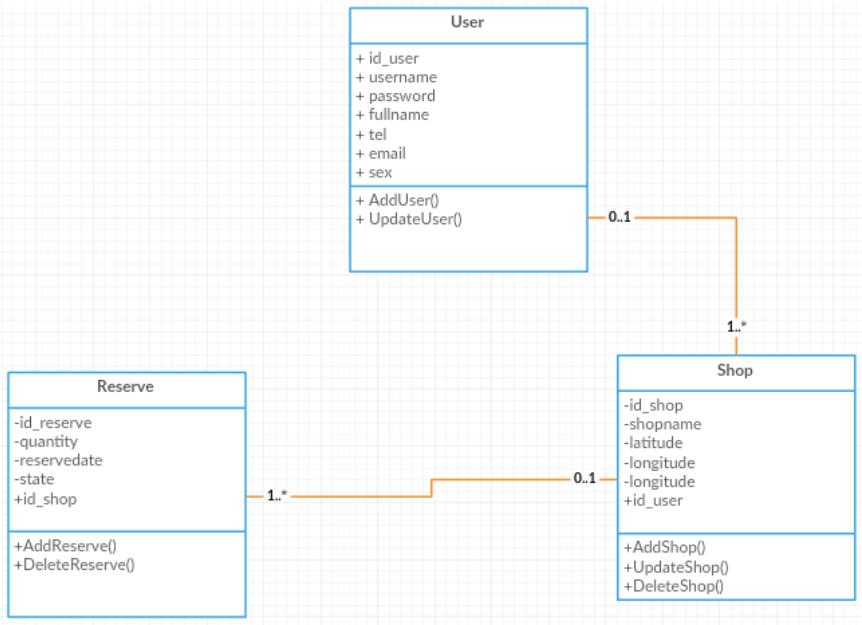

Figure 3.9: Class Diagram

4 Tools and Technologies used for our implementation

Our solution of VRP is implemented as a geolocation web application based on Google API and web technologies like: HTML, PHP, CSS3 in addition to MYSQL are used for the web application design and development. Since our web application is based on client/server approach, Java Servelet is the best choice to implement our application . Other technologies are used to such as: JQuery and Json are also employed. We use JQuery to benefit of the update ofinformation on a webpage in real time, without reloading the page, and the Json is used for linking between Application android and the server web.

4.1 Google Maps API

The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface. It is designed to work on both mobile devices as well as traditional desktop browser applications. The API includes language localization for over 50 languages, region localization and geocoding, and has mechanisms for enterprise developers who want to utilize the Google Maps API within an intranet. The API HTTP services can be accessed over a secure (HTTPS) connection by Google Maps API Premier customers.

4.2 jQuery

Figure 3.10: jQuery Logo

jQuery is a JavaScript library that allows web developers to add extra functionality to their websites. It is open source and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in web development.

To implement jQuery, a web developer simply needs to reference the jQuery JavaScript file within the HTML of a webpage. Some websites host their own local copy of jQuery, while others simply reference the library hosted by Google or the jQuery server. For example, a webpage may load the jQuery library using the following line within the <head> section of the HTML:

```
<scripttype="text/javascript"src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1
/jquery.min.js">
</script>
```

Once the jQuery library is loaded, a webpage can call any jQuery function supported by the library. Common examples include modifying text, processing form data, moving elements on a page, and performing animations. Since jQuery runs on the client side (rather than the web server), it can update information on a webpage in realtime, without reloading the page. A common example is "autocomplete," in which a search form automatically displays common searches as you type your query. In fact, this is how TechTerms.com provides search suggestions when you type in the search box.

Besides its free license, the other main reason jQuery has gained such popularity is its cross-browser compatibility. Since each browser renders HTML, CSS, and JavaScript differently, it can be difficult for a web developer to make a website appear the same across all browsers. Instead of having to write custom functions for each browser, a web developer can use a single jQuery function that will work in Chrome, Safari, Firefox, and Internet Explorer. This multi-browser support has led many developers to switch from standard JavaScript to jQuery, since it greatly simplifies the coding process.

4.3 Json

Figure 3.11: JSON Logo

Short for JavaScript Object Notation, JSON is a lightweight data-interchange format that is easy for humans to read and write, and for machines to parse and generate. JSON is based on the object notation of the JavaScript language. However, it does not require JavaScript to read or write because it is a text format that is language independent.

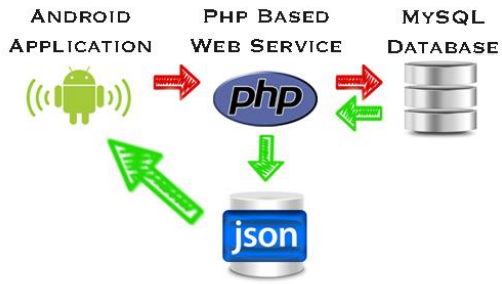

Figure 3.12: Explain JSON

4.4 Servlet

A servlet is a Java class that allows to dynamically create data within an HTTP server. These data are most commonly presented in HTML format, but they can also be in the XML format or any other format intended for web browsers. Servlets use the Java Servlet API (package javax.servlet).

A servlet runs dynamically on the web server and allows the extension of the functions of the latter, typically: access to databases, e-commerce transactions, etc. A servlet can be loaded automatically when you start the web server or on the first requestof the client. Once loaded, servlets are still active waiting for other client requests.

The use of Servlet is done through a servlet container (framework) server-side. This is the servlet execution environment and allows it to persist between requests fromclients. The API defines the relationships between the container and the servlet. Thecontainer receives the request from the client, and selects the servlet that will have to deal with it. The container provides also a range of standard services to simplify the management of requests and sessions.

A (e) or multiple servlets are Java web applications, their principle of operation (architecture, features, configuration, deployment) is described following an official specification, led by Sun Microsystems and to which any person may contribute throughthe Java Specification Requests (Java Community Process). The current version of the servlet specification is 3.1

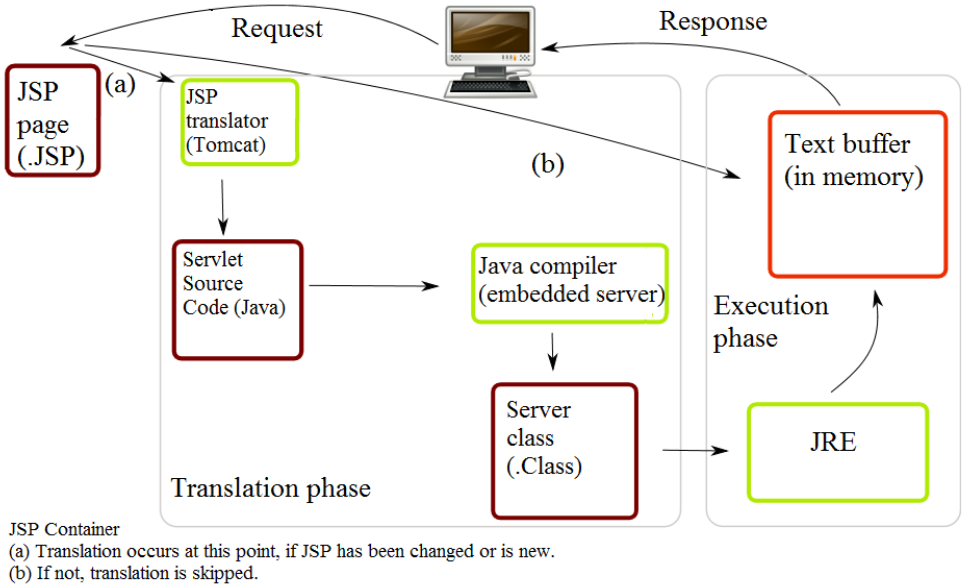

Figure 3.13: Explain Servlet

4.5 MySQL

MySQL is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a Web scripting language like PHP to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

4.6 WAMP

Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often used for web development and internal testing, but may also be used to serve live websites.

The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet.

4.7 Eclipse

Figure 3.14 Android Developer Tools

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

4.8 NetBeans IDE

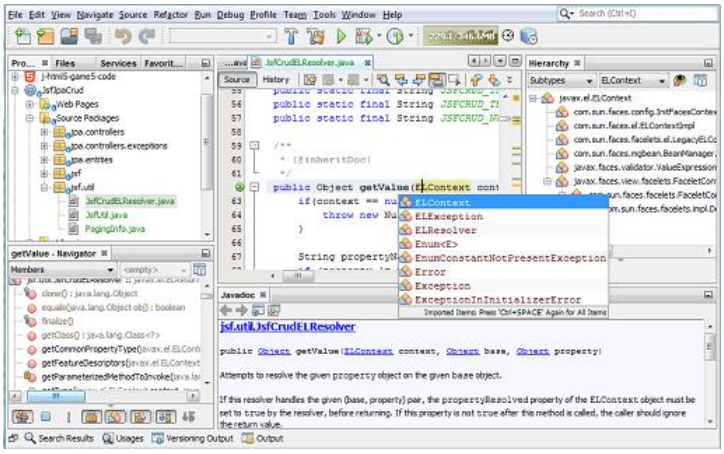

Figure 3.15: NetBeans interface

NetBeans IDE is a free and open source integrated development environment for application development on Windows, Mac, Linux, and Solaris operating systems.

The IDE simplifies the development of web, enterprise, desktop, and mobile applications that use the Java and HTML5 platforms. The IDE also offers support for the development of PHP and C/C++ applications.

5 Presentation of the system

In the following section we will present snapshot of the most important pages of our managing web application and the android application in relation

5.1 Web application

5.1.1 For the client(Customer)

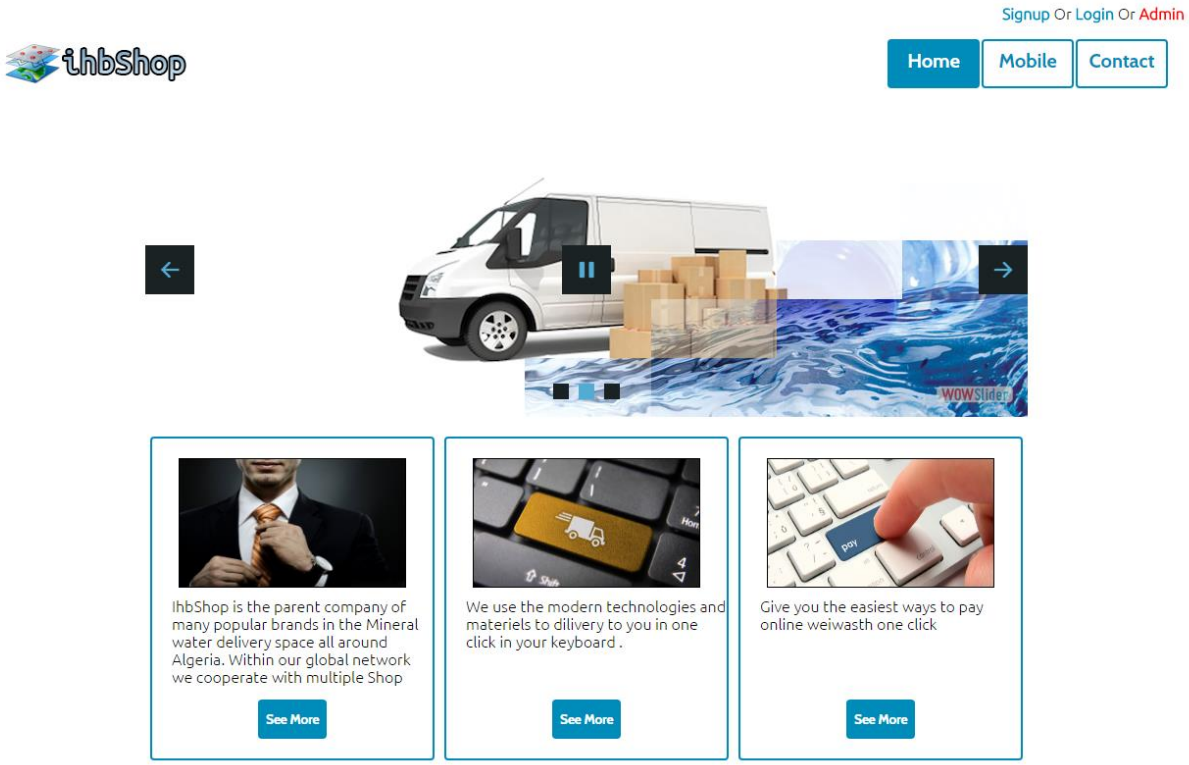

Figure 3.16: Website interface

Figure 3.17: Signup and login

If you are a new user you need to register by clicking to Signup.

If you are already registered , To access Click the Login .

If you are an Admin To access To Control Panel of the website Click Admin.

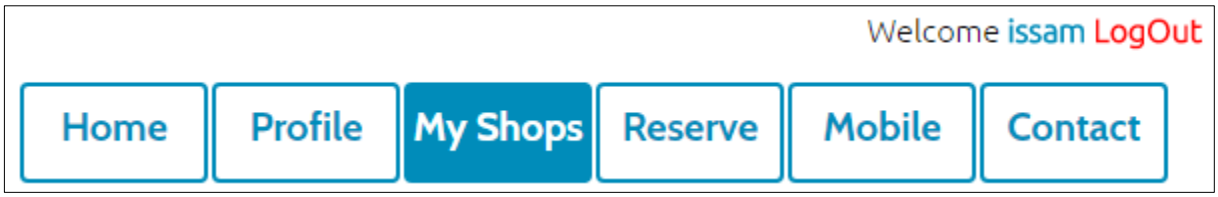

Figure 3.18: Menu Bar

Profile :To see and change personal information when needed
My Shops : To see and change Shops information when needed
Reserve : To see and change Reserves when needed
When we chose My shops we obtain :

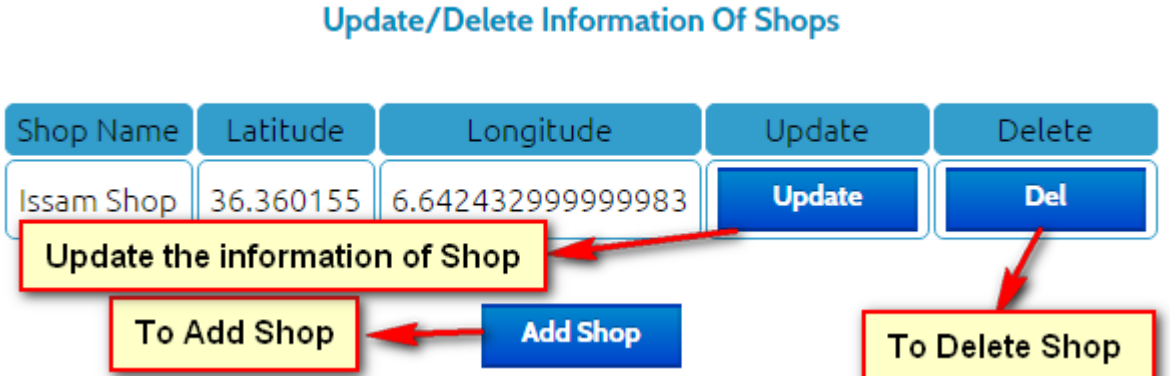

Figure 3.19: My shops setting

When we click in Add Shop Button we obtain

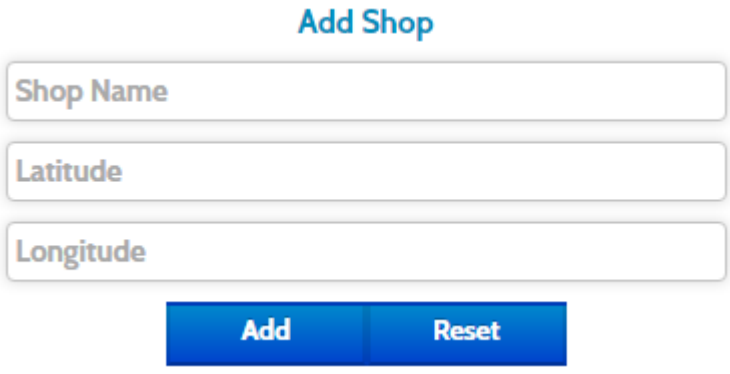

Figure 3.20: Add shop Form

To Add new Shop You must enter your shop name and store information for localization .

When we click in Reserve in bar menu

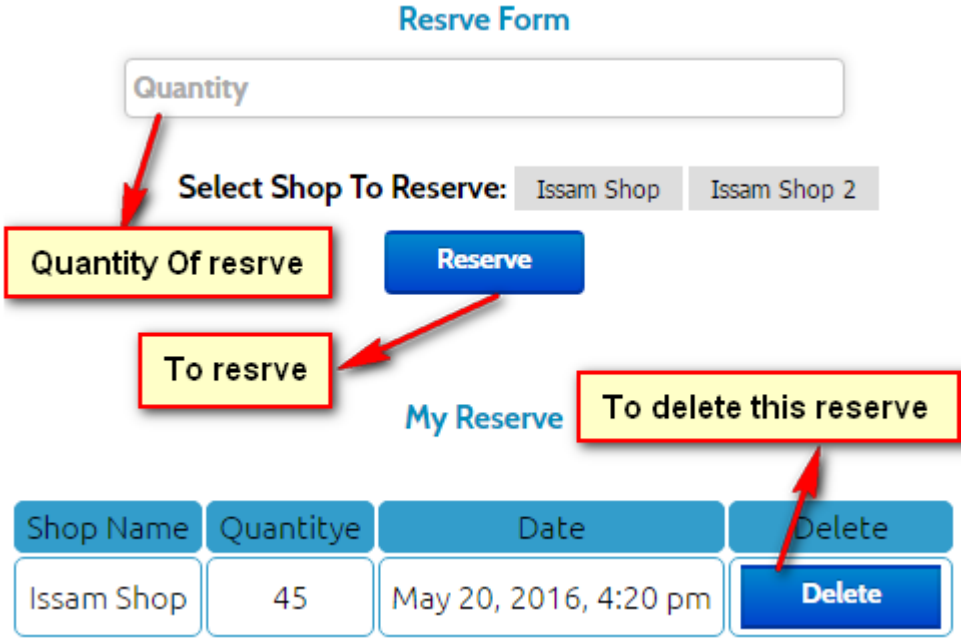

Figure 3.21: Reserve setting

5.1.2 For the Admin

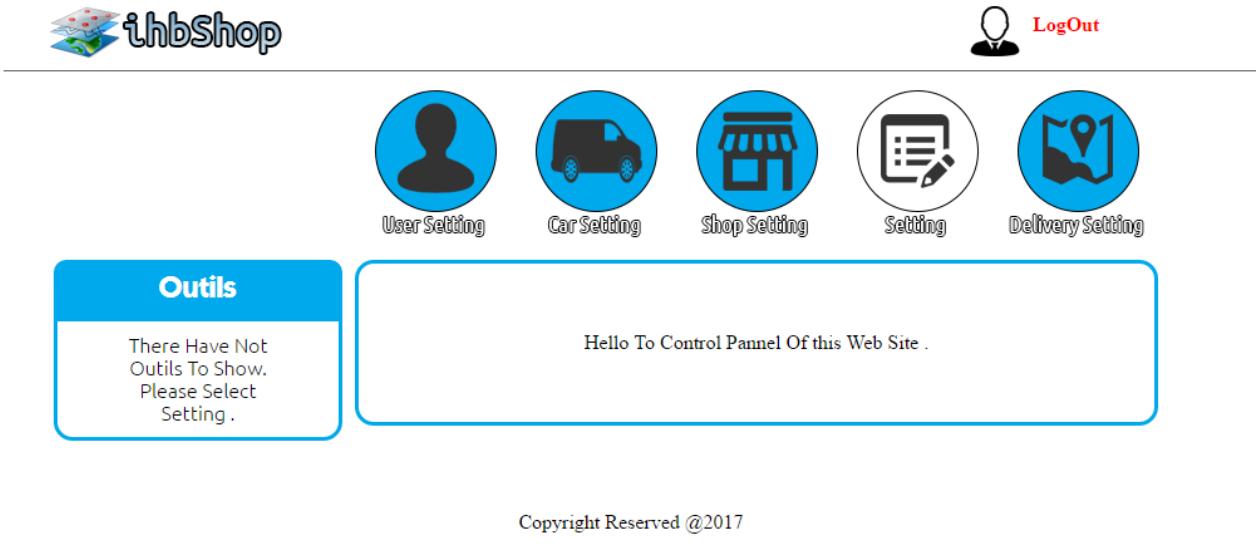

Figure 3.22: Admin Control panel interface

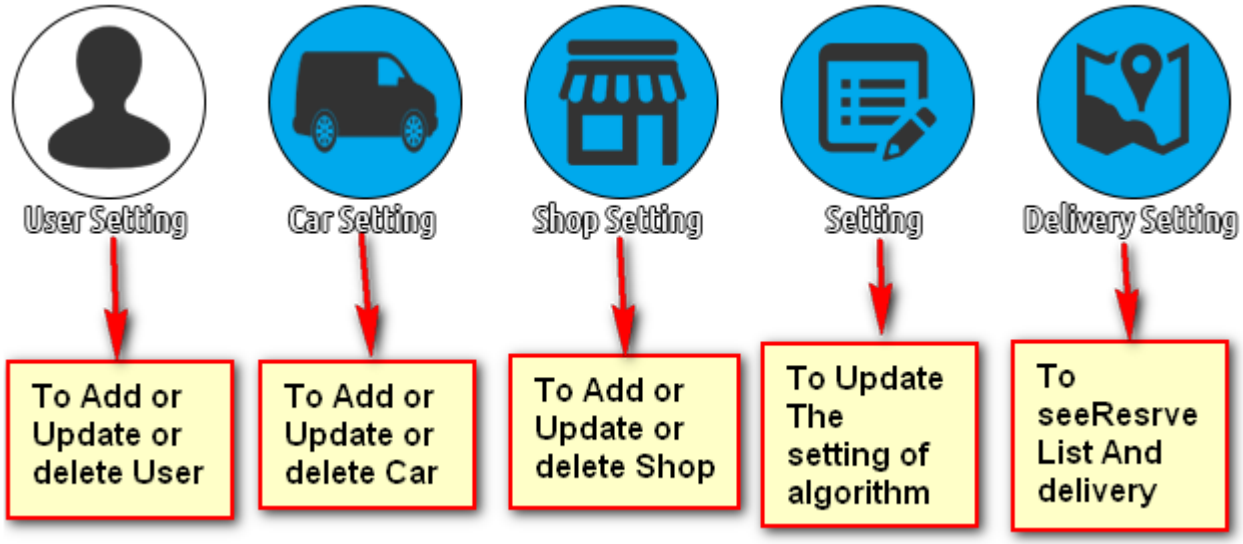

Figure 3.23: Menu bar of control panel

After the reservation of customers and the application of our based on GA we will obtain This the result (here I used a simple example):

Figure 3.24: Example of a delivery

5.2 The Android application

since the Smartphone invaded our life, w will decide to create an Android Application to facilitate the demands of custemer .

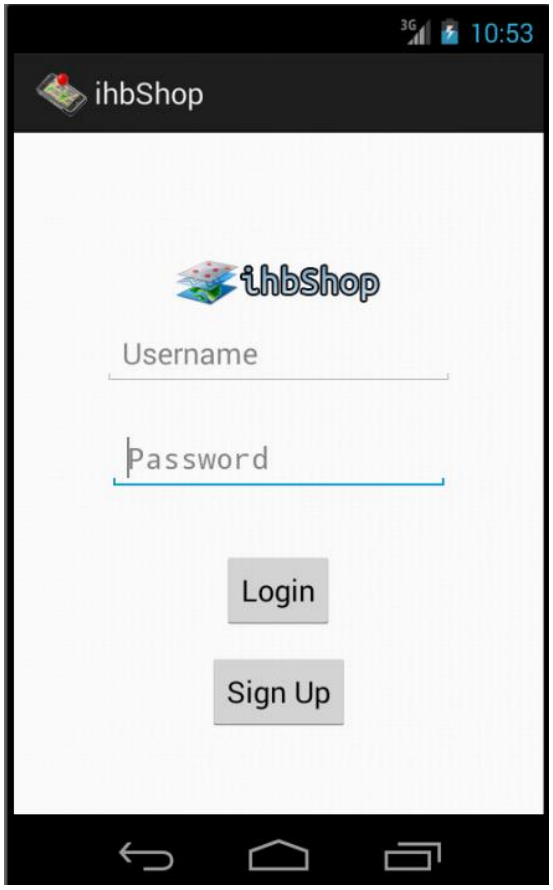

Figure 3.25: Login Form

If you are a new user you need to register By pressing the Sign Up button.

If you are already registered You must enter Username , Password And Clicking the Login Button .

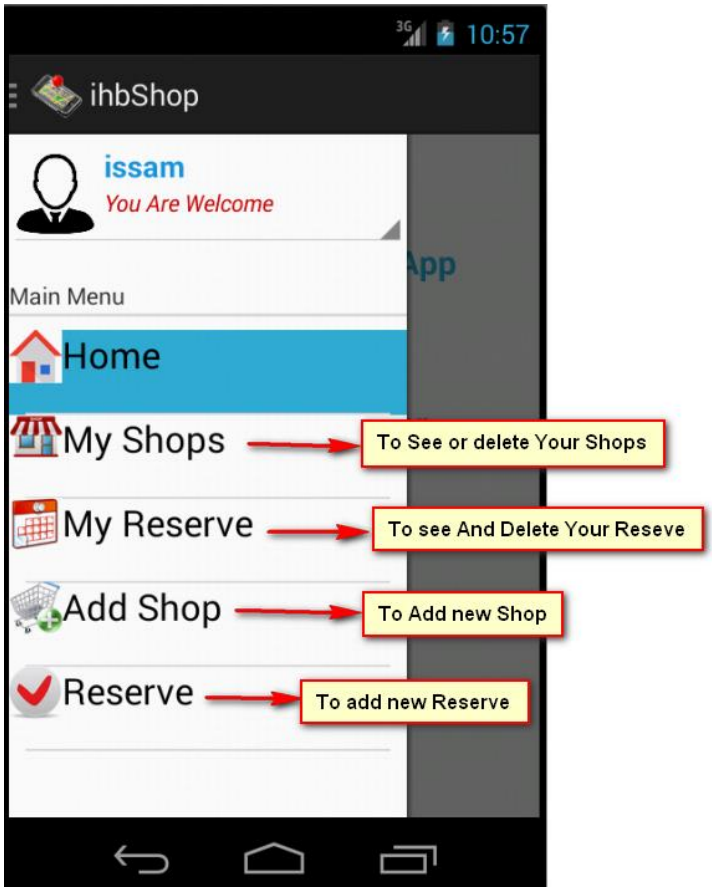

Figure 3.26: Navigation Drawer

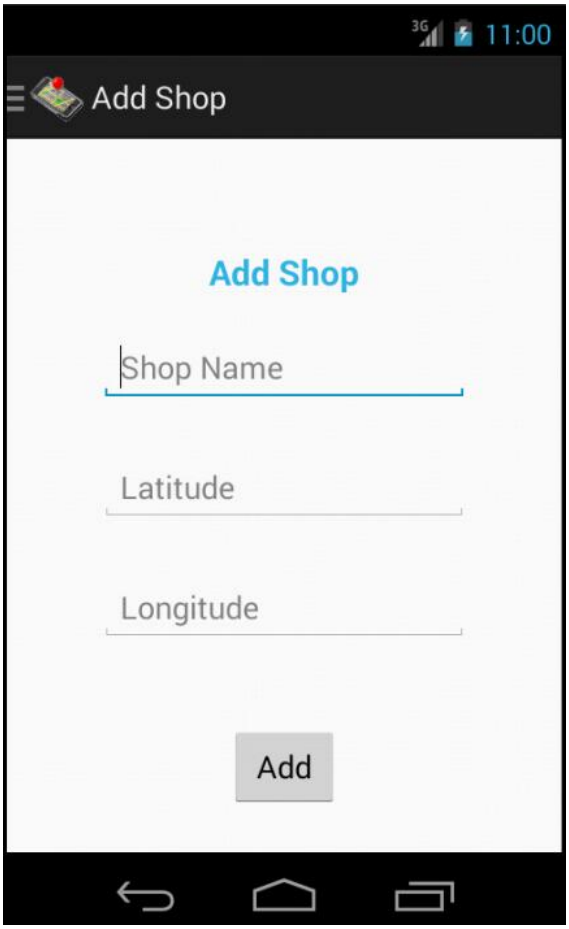

Figure 3.27: Add Shop Form

To Add new Shop You must enter his name and store information for localization .

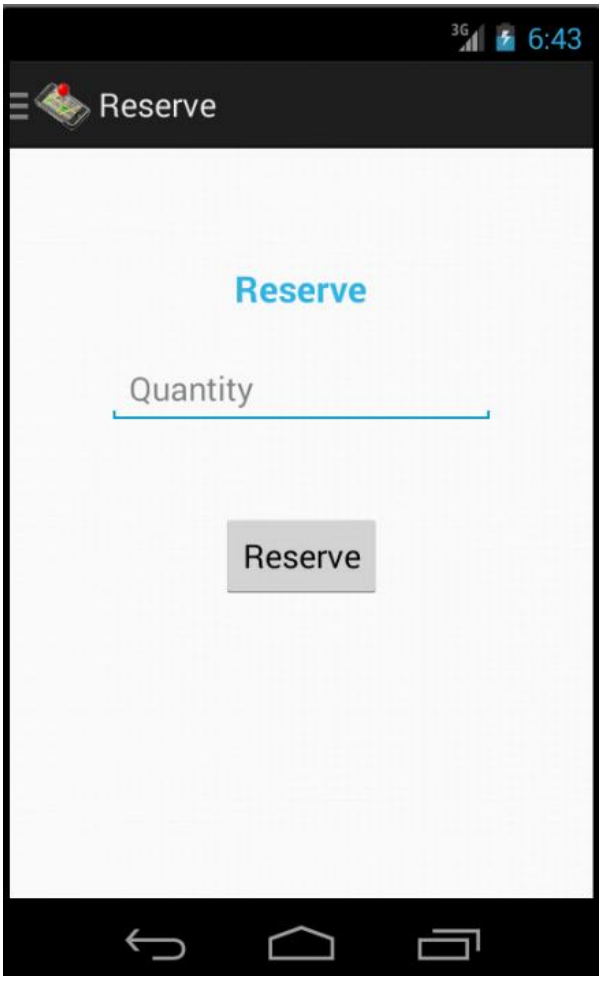

Figure 3.28: Reserve Form

6 Conclusion

In this chapter we presented and explained our vision to solve Vehicle Routing Problem, our approach is based on heuristics, its aspired by Genetic Algorithms. Our goal is to optimize the time taken by vehicles along the shortest path and satisfy more consumers. The solution is supported by a web application that helps the manager satisfy client request by lower costs. In addition, we have implement our solution via an Android application for the client side.

GENERAL CONCLUSION

The vehicle routing problems are an integral part of deciders and planners daily life. It's about determine the vehicles fleet tours to deliver a clients list, or make interventions tours (maintenance, repair, controls) or visits (medical visits, commercial, etc.).

As part of this thesis, we are interested in a variant important to these problems which is the problem of transport on demand in several vehicles with a limited capacity, and time window in the static case; it means that all the clients requests are known beforehand. this consists in determining the tours and schedules for vehicles that carry out the users transport on their demand, in order to determine the shortest path.

We proposed our approach to solving the VRP, it is mainly based on the genetic algorithm with a new technique which consists in doing crossing and mutation not on the individual as a whole, but only on the order succession of query execution.

The testing has shown that our approach is a good combination between the transport total cost and the transport time for each client.

For the real data, the developed genetic algorithm has proven its effectiveness by:

_ the minimization of transport time.

_ the minimization of transportation cost.

_ the minimization of the vehicles path.

This work is concerned for one seller seen as an administrator, in a prospect of improvement; we propose that an infinite number of sellers can register on this site or application and exploit it, also clients can pay for their purchases with electronic payment. We also propose to increase the challenge and try to solve Multi-Depot Vehicle Routing Problem.

Bibliography

[1] Martin Jantscher,ANDROID, Android Team 2009

[2] Susannah David on palzer,hello, android. PragmaticBookshelf, 2009-7-21

[3] Raksmey PHAN, Une stratégie hybride pour le problème de tournées de véhicules avecLivraisons et collectes. Laboratoire d'Informatique, de Modélisation et d'optimisation des systèmes, septembre 2009.

[4] Cesar Rego et Catrine Roucairol, Le Problème de Tournées de Véhicules : étude et résolution approchée. INRIA, février 1994.

[5]T.K. Ralphs, L. Kopman, W.R. Pulleyblank and L.E. Trotter, "On the capacitated vehicle routing problem", Mathematical Programming 94, 343-359, 2001.

[6] B. Goldberg, A. Assad, A. Levy and F. Gheysens "The fleet size and mix vehicle routing problem", Management Science and Statistics. Working paper, n° 82-020, 1982.

[7] D. Goldberg. Genetic algorithms in search, optimization and machine learning. Addison Wesley, 1989b. (Cité page 16.)

[8] H. Mülhenbein. Evolutionary Algorithms : Theory and Applications. Wiley, 1993. (Cité page 20.)

[9] M. Gendreau, G. Laporte, and J.-Y. Potvin. Métaheuristiques pour le problème des tournées de véhicules. In Résolution de problèmes de RO par les métaheuristiques (Traité IC2, série Informatique et systèmes d'information). Hermès, 2003.

[10] A. Colorni, M. Dorigo, et V. Maniezzo. An investigation of some proprieties of an "ant algorithm". In Manner and Manderick, pages 509–520, 1992b.

[11] T.G. Crainic and F. Semet. Recherche opérationnelle et transport de marchandises. In Optimisation combinatoire. 3, Applications. Hermès Science : Lavoisier, 2006.

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

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

[14] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. présenté au Congress on Numerical Methods in Combinatorial Optimization, Capri, Italie, 1986. (Cité page 14.)

[15] J. Berger, M. Salois, and R. Begin. A hybrid genetic algorithm for the vehicle routing problem with time windows. In Robert Mercer and Eric Neufeld, editors, Advances in Articial Intelligence, volume 1418 of Lecture Notes in Computer Science, pages 114127. Springer Berlin / Heidelberg, 1998. 30

[16] J. Louis Sushil and Tang Rilun. Interactive genetic algorithms for the traveling salesman problem. In Proceedings of the 1999 Genetic and Evolutionary Computation Conference (GECCO 1999), volume 1, pages 385 392, Orlando, Florida, USA, 1999

Abstract

In the industry field The challenging strategy of supply chain management and logistics is to optimize the product delivery from suppliers to customers thus satisfying constraints. Such problems are known as Vehicle Routing Problems (VRP), in which the vehicles leave the depot, serve customers assigned and upon completion of their routes return to the depot. In this thesis, our goal is to minimize the distance travelled by applying the bio-inspired Genetic Algorithms. After that our solution is implemented through a web application and to increase the challenges we will create an android application for the management of our system .

Key words: Vehicle Routing Problems, optimization, Genetic Algorithms

Résumé

Dans le domaine de l'industrie, le défi dans la gestion de la chaîne logistique d'approvisionnement est d'optimiser la livraison des produits des fournisseurs aux clients ainsi satisfaire les contraintes. De tels problèmes sont connus comme des problèmes d'acheminement des véhicules, dans lequel les véhicules quittent le dépôt pour satisfaire les clients puis retournent au dépôt. Dans cette thèse, notre objectif est de réduire au minimum la distance parcourue par l'application des algorithmes génétiques. Après notre solution est mise en œuvre par le biais d'une application Web et pour augmenter les défis nous allons créer une application Android pour la gestion de notre système.

Mots clés: problèmes d'acheminement des véhicules**,** optimisation**,** algorithmes génétiques.

ملخص

في مجال الصناعة أكبر تحدي في استراتيجية إدارة سلسلة التوريد والخدمات اللوجستية هي تحسين تسليم المنتجات من الموردين للعملاء بوجود قيود و شروط ومن المعروف أن مثل هذه المشاكل تعرف بإسم مشاكل توجيه المركبات، حيث تغادر المركبات المستودع، متوجهة بالسلع للعملاء ، وعند الانتهاء تعود إلى المستودع .هدفنا في هذه الأطروحة، هو تقليل المسافة التي تقطعها مركبات بتطبيق الخوارزميات الجينية .بعد ذلك توفير حلنا من خلال تطبيق ويب ولزيادة التحديات قمنا بإنشاء تطبيق اندرويد لإدارة نظامنا.

الكلمات المفاتيح: مشاكل توجيه المركبات، التحسين، الخوارزميات الجينية