PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
MOHAMED BOUDIAF UNIVERSITY - M'SILA

FACULTY: Mathematics and Informatics
Department of Computer Science

DOMAIN: Mathematics and Informatics
FIELD: Computer Science

A Dissertation in Fulfillment

For the Requirements of the Degree of Master

Option: Information Systems and software engineering(**SIGL**)

By: Cherif Ibtissam

SUBJECT

Production and Delivery of Radiopharmaceuticals to Medical Imaging Centers

Defended publicly on: 25/06/2018 Board of Examiners:

LOUCIF HAMZA	**M'silaUniversity**	**Chairman**
Dr. LAMRI SAYAD	**M'silaUniversity**	**Supervisor**
THERAFI ABDELLAH	**M'silaUniversity**	**Examiner**

Promotion: 2017/2018

Acknowledgement

The first thanks are to Allah then, Foremost, I would like to express my sincere gratitude to my advisor Prof "LAMRI SAYAD" for the continuous support of my master's study and research, for his patience and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

Finally, I am very grateful to my family and my friends for their unlimited love and support during the whole years of my study. This work is dedicated to all of them without exception.

Table of Contents

List of figures

List of tables

General Introduction

The problem we study falls in the general category of the vehicle routing problem (VRP) which is well studied in the area of Operations Research. There are many papers dealing with several variations/extensions of the VRP, mainly because of its wide applicability in real-world applications.

However, there is very limited research in the distribution of radio-pharmaceuticals and for this reason we focus on the application of VRP in the distribution of short-lived products which must arrive at the customer site between a specified time window. Emphasis is given in applications in the medical field. The original VRP formulation was introduced by Danzig and Ramser back in 1959. A detailed review of the classical VRP has been written by Toth and Vigo and Parragh et al. Laporte provides an overview of major VRP definitions as well as efficient exact and approximate algorithms for solving it. Savelbergh focused on the complexity of the VRP and proved that it is an NP-hard problem.

In the classical VRP we are given a set of trucks with a limited capacity and a set of customers each having a known demand of a product that is manufactured in a specific location. The distance and traveling times from the manufacturing site to each customer location is known. The aim is to determine the minimum cost or distance routes such that (a) every customer is served by only one vehicle, (b) all routes start and end at the depot and (c) the transportation vehicles do not carry weight more than their capacity. A very popular extension of the VRP is the case where the customers request the delivery of their orders to arrive during a pre-specified time interval. This extension is known as the VRP with Time Windows (VRPTW) and it is often used in the transportation of perishable products. Many transportation problems in the radiopharmaceutical industry can be modeled as VRPTW.

In general, there are two types of approaches used to solve VRPTWs: (i) exact (e.g., branch-and-cut, branch-and-price) and (ii) heuristics and metaheuristics (e.g., tabu search, genetic algorithms). Our work concerns the adaptation and implementation of genetic algorithm to deal with the radiopharmaceutical products delivery described as a VRPTW problem.

This manuscript is organized as follows:

- The first chapter defines the concepts of radiopharmaceutical production and delivery and introduces the basic elements regarding VRP problem.
- The second chapter describes mathematically the problem of radiopharmaceutical delivery.
- The third chapter explains some approaches to tackle the described issue. Mainly, it focuses on genetic algorithm
- The last chapter discusses the obtained results.

Chapter 01

Definitions and fundamental concepts of radiopharmaceuticals production and availability

1 Introduction

The use of radioactive materials in medical applications is widespread, especially within the field of medical imaging the use of radiopharmaceuticals/radiotracers is common. A problem that arises is that these drugs have a limited timespan within which they can be used safely and effectively. This makes the logistical problem of delivering these drugs quite difficult. We consider the production, planning and routing problem of supplying several imaging centers with radiopharmaceuticals produced at a single production center with multiple production lines. A limited number of vehicles can be used to transport the drugs at a certain cost. Appointments do not have to be met, but in this case a cost is incurred.

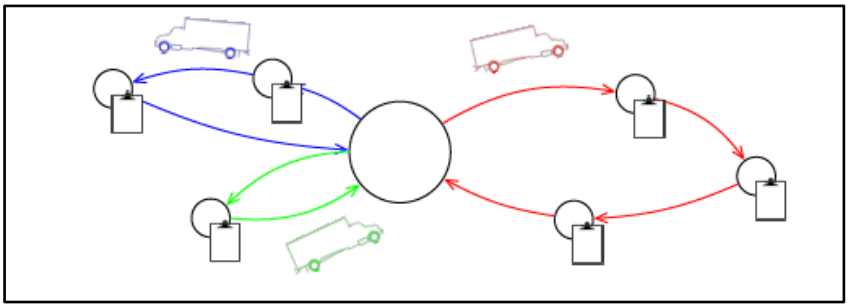


Figure 1.1: Vehicle routing from one production center to several imaging centers according to appointments. [16]

2 Radioisotopes for radiopharmaceuticals: history and growth

1. Using a specific radiotracers called radiopharmaceuticals for imaging organ function and disease states are a unique capability of nuclear medicine. imaging modalities such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI) and Ultrasonography (US), nuclear medicine procedures are capable of mapping physiological function and metabolic activity and thereby giving more specific information about the organ function and dysfunction. The use of mapping for the radiopharmaceutical distribution in vivo provides images of functional morphology of organs in a non-invasive manner and plays an important role in the diagnosis of many common diseases associated with the malfunctioning of organs in the body as well as in the detection of certain type of cancers. The outspread utilization and growing demands for these techniques are directly attributable to the development and availability of a vast range of specific radiopharmaceuticals. [17]

2. The radiopharmaceuticals are medicinal formulations containing radioisotopes which are safe for administration in humans for diagnosis or for therapy. in spite of radiotracers were tried as a therapeutic medicine immediately after the discovery of radioactivity, the first significant applications came much later with the availability of cyclotrons for acceleration of particles to produce radioisotopes. later, nuclear reactors realized the ability to prepare larger quantities of radioisotopes. Radioiodine (iodine-131), for example, was first introduced in 1946 for the treatment of thyroid cancer, and still the most efficacious method for the treatment of hyperthyroidism and thyroid cancer. [18]

3. from the main objects for setting up nuclear research reactors was for the preparation of radioisotopes. Betwixt the several applications of radioisotopes, medical applications were considered to be of the highest priority. the most of medium flux and high flux research reactors now are routinely used to produce radioisotopes for medical, and also industrial, applications. Most of the commonly used reactor produced isotopes in medical applications are molybdenum-99 (for production of technetium-99m), iodine-131, phosphorus-32, chromium-51, strontium-89, samarium-153, rhenium-186 and lutetium-177. [19]

4. The using early of cyclotron in radiopharmaceuticals field was for the production of long lived radioisotopes that can be used to prepare tracers for diagnostic imaging. And for this, medium to high energy (20-70 MeV) cyclotrons with high beam currents were needed. Isotopes such as thallium-201, iodine-123 and indium-111 were prepared for use with single photon emission computed tomography (SPECT). With the advent of positron emission tomography (PET), there has been a surge in the production of low energy cyclotrons (9-19 MeV) exclusively for the production of short lived PET radionuclides such as fluorine-18, carbon-11, nitrogen-13 and oxygen-15. Figure 1.2 shows such a machine. The majority of the cyclotrons (~350) worldwide are now used for the preparation of fluorine-18 for making radiolabeled glucose for medical imaging. [20]

Figure 1.2: A 13 MeV cyclotron (indigenous product) in operation in Chosun University, Republic of Korea. [13]

2.1 Radiopharmaceuticals Production Aspects and Challenges

5. Presently there are over 100 radiopharmaceuticals developed using either reactor or cyclotron produced radioisotopes and which are used for the diagnosis of several common diseases and the therapy of a few selected diseases, including cancer. Radiopharmaceuticals production involves handling of large quantities of radioactive substances and chemical processing. The sides which need to be addressed in radiopharmaceuticals production, including the management of radioisotope production, include import, operation and

maintenance of processing facilities, complying with the codes of current good manufacturing practices (cGMP), ensuring effective quality assurance and quality control (QA & QC) systems, registration of the products with national/regional health authorities and radioactive material transport etc. [21]

6. the production of Radiopharmaceuticals, contrast conventional pharmaceuticals production, is remains on a relatively small scale and implementing the cGMP guidelines which are applicable for the drugs industry is both difficult and expensive. Guarantee cGMP compliance is a demanding task for a small scale manufacturer, as it involves taking care of several aspects prior to, during and after production. These include the development of well qualified personnel, use of controlled materials and procedures, availability of qualified equipment, production of the products in designated clean areas, applying validated processes and analytical methods, full documentation of the process and release of the final product by a qualified person. Application of clean room requirements in radioisotope laboratories in general and hot cells in particular (Fig. 1.3), is technically demanding to be compatible with the requirements for both radiological and pharmaceutical safety. The Agency assists its Member States to improve the radiopharmaceuticals production to meet cGMP as adaptable to radioactive products by providing appropriate documents, conducting training courses and supporting technical cooperation Project. [22]

Figure 1.3: Hot cells with manipulators used for radioisotopes/radiopharmaceuticals production available from commercial sources. [13]

7. In the latest decade has seen an increase in the use of PET in regular diagnostic imaging, and a commensurate use of PET radiopharmaceuticals, particularly fluorine-18 in the form of fluorodeoxy glucose (18F-FDG). The associated 511 keV high-energy radiation needs thicker shielding and more sophisticated handling devices. In view of the short half-lives, the emphasis is also increasingly on process validation and strict adherence to approved procedures in handling of all steps of manufacture, rather than relying on the final QC test results alone. Needing for the rapid, remote and reliable synthesis of PET radiopharmaceuticals has led to the

introduction of microprocessor controlled automated synthesis modules. This experiment has also led to the development of similar automated synthesis module systems for other radiopharmaceuticals. [23]

3 Definition and theory

3.1 The Vehicle Routing Problem

3.1.1 VRP Definition

The VRP (Vehicle Routing Problem) depend of a set of customers with known demands. For most cases, it exists only one depot. Each vehicle can carry either a limited or unlimited goods and travel a limited distance. Each vehicle starts from a depot and delivers the goods required, then returns to the depot. Every customer is assigned to exactly one vehicle route. The total demand of any route must not exceed the vehicle capacity. The objective of the vehicle routing problem is to deliver to a set of customers with known demands with minimum-cost vehicle routes originating and terminating at a depot. Figure 1.4 shows a picture of the typical input for a vehicle routing problem. Figure 1.5 shows a possible output. [2]

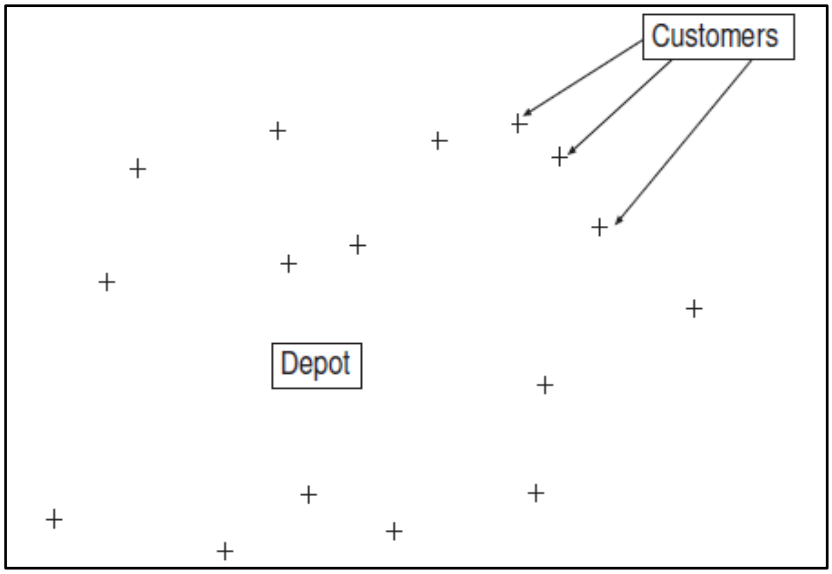


Figure 1.4: An input for a vehicle routing problem [2]

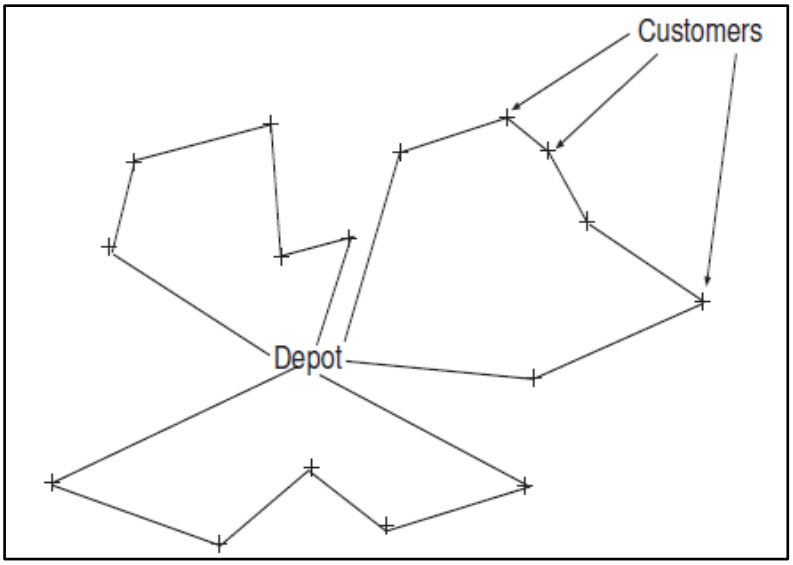


Figure 1.5: An output for the instance in Figure. [2]

A VRP is a well-known integer programming problem which belongs into the category of NP-Hard problems, that mean the computational effort required to solve this problem increases exponentially with the problem size. For such problems it is often desirable to obtain approximated solutions, so they can be found quickly enough and are sufficiently accurate for the purpose. Usually this task is accomplished by using various heuristic methods, which rely on some insight into the nature of the problem.[2]

The Vehicle Routing Problem arises naturally as a central problem in the fields of transportation, distribution, and logistics. In some market sectors, transportation means a high percentage of the value is added to goods, Therefore, the utilization of computerized methods for transportation often results in significant savings ranging from 5% to 20% in the total costs. [2], In some real world vehicle routing problems there are often side constraints due to other restrictions. Some of the well-known models are:

- Each customer has to be supplied within a certain time window **(vehicle routing problem with time windows - VRPTW).**
- The vendor uses many depots to supply the customers **(multiple depot vehicle routing problem - MDVRP).**
- Customers may return some goods to the depot **(vehicle routing problem with pick-up and delivering - VRPPD).**
- The customers may be served by different vehicles **(split delivery vehicle routing problem - SDVRP).**
- The demands, service time and/or travel time are random **(stochastic vehicle routing problem -SVRP).**
- **Satellite facilities** are used to replenish vehicles during a route - **(VRPSF).**

- A VRP in which customers can demand or return some commodities **(vehicle routing problem with backhauls -VRPB).** [2]

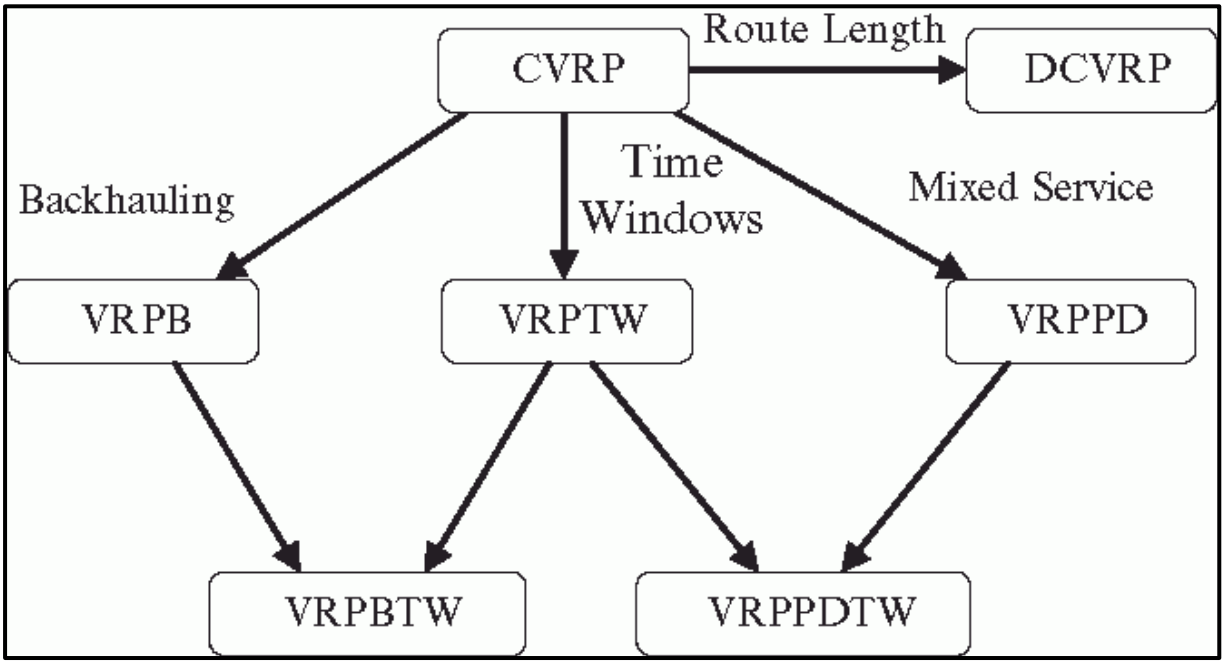


Figure 1.6: different variant of vrp. [32]

3.1.2 VRP in Real life

The VRP is of great practical significance in real life. It shows in a large number of practical situations, such as transportation of people and products, delivery service and garbage collection. For an example such a matter of course as being able to buy milk in a store, arises the use of vehicle routing twice. First the milk is collected from the farms and transported to the dairy and when it has been put into cartons it is delivered to the stores. That is the way with most of the groceries we buy. And the transport is not only made by vehicles but also by planes, trains and ships. VRP is everywhere around! One can therefore easily imagine that all the problems, which can be considered as VRP, are of great economic importance, particularly to the developed nations. The economic importance has been a great motivation for both companies and researches to try to find better methods to solve VRP and improve the efficiency of transportation. [5]

3.1.3 Solution Methods

The VRP belongs to the category of NP-hard problems, meaning that the computational effort required to solve this problem increases exponentially with the problem size in the worst case. A small VRP (up to 100 customers) is solvable with exact algorithms, but for a large VRP (more than 100 customers) it is desirable to obtain approximate solutions using heuristics as well as metaheuristics. These methods often give routes of sufficient quality within reasonable computation time when exact algorithms are not fast enough. More detailed information about

exact approaches, heuristic methods and metaheuristics is given below in the same order used by Dorronsoro. [5]

Exact Approaches

For small problems, exact approaches are proposed that evaluate implicitly, every possible solution to obtain the best solution. A well-known exact method is the branch and bound method, which consists of a systematic implicit enumeration of all feasible solutions. Using lower and upper bounds on the optimal objective value, more and more subsets of the feasible solutions will be rejected, such that the optimal solution appears. Another exact approach is the branch and cut method, a hybrid of the branch and bound method and the cutting plane method. The cutting plane method adds linear inequalities, called cuts, to the problem in order to define an as small as possible feasible set of the objective values. To prevent a slow convergence to the optimal value, the structure of the problem can be used to generate very good cuts. One of the most efficient exact methods in the literature is proposed by Baldacci et al, who combine three different bounding procedures to compute the optimal solution. [5]

Heuristic methods

To approximately solve larger problems within reasonable computation time, heuristic methods are used. Heuristic methods perform a relatively limited exploration of the search space and typically produce good quality solutions within modest computation times. The heuristic methods can be divided into two-phase methods and constructive methods. Two-phase methods are decomposed into two natural components, on the one hand clustering of customers into feasible routes and on the other hand actual route construction (like solving a Travelling Salesman Problem (TSP)), with possible feedback loops between the two stages. Constructive methods consist of route construction heuristics and local improvement heuristics. A route construction heuristic quickly builds a feasible solution, but usually not the optimal one. The most well-known route construction heuristic is the greedy nearest neighbor heuristic. This heuristic starts at an arbitrary customer, subsequently it chooses the nearest customer as the next one to visit and so on, until a feasible solution is obtained. Another well-known route construction heuristic is the Clarke-Wright savings heuristic. This savings heuristic starts with an initial allocation of each customer to a separate route. Then for each pair of customers the cost savings of joining those customers on one route are calculated. Based on the values of these savings, the customers are sequentially joined into routes starting with the customer combination yielding the largest cost savings until no further savings can be achieved. Local improvement heuristics are heuristics based on simple route modifications. The most well-known local improvement heuristic is the k-opt heuristic, which removes k arcs and reconnects the routes in another possible way to find a better solution. Other well-known local improvement heuristic is the Or-opt heuristic (an intermediate heuristic between 2-opt and 3-opt), the crossover heuristic (two routes are mixed at one point), the customer relocation heuristic (a customer located at one route is changed to another one) and the customer exchange heuristic (two customers of two different routes are interchanged between the two routes). [5]

Metaheuristics

Metaheuristics are more generic solution schemes that combine several heuristic methods. In these methods, the emphasis lies on performing a deep exploration of the most promising regions of the solution space. The quality of solutions produced by these methods has the potential to be higher than that obtained by only one heuristic method. A metaheuristic is a search strategy that first constructs an initial solution using a route construction heuristic. The initial solution is then improved by an iterated local search procedure, with phases of intensification and diversification. Intensification is achieved through finding an optimum in the neighborhood of the initial solution using local improvement heuristics. When a local optimum has been found, several diversification mechanisms like a random restart, or penalizing moves, may be invoked to 'jump' to an unexplored part of the search space in order to escape from poor local minima. More detailed information about local search is formulated by Aarts and Lenstra (1997). [5]

3.1.4 Classical types of Meta-heuristics

There are a large types of metaheuristics like Greedy Randomized Adaptive Search Procedure (GRASP), Ant Colony Optimization (ACO), Variable Neighborhood Search (VNS), Simulated Annealing (SA), Tabu Search (TS) and Evolutionary Algorithm (EA). An overview of these types is given below and more information including references on specific metaheuristics can be found in Gendreau et al.

- **Greedy Randomized Adaptive Search Procedure (GRASP)** focuses on the route construction heuristic.
- **Ant Colony Optimization (ACO)** is another metaheuristic with focus on the route construction
- **Variable Neighborhood Search (VNS)** is a metaheuristic with focus on the iterated local search procedure, which exploits neighborhoods to escape from local minima
- **Simulated Annealing (SA)** is another metaheuristic with focus on the iterated local search procedure, which allows bad solutions to escape from local minima.
- **Tabu Search (TS)** is like VNS and SA another metaheuristic with focus on the iterated local search procedure.
- **Evolutionary Algorithm (EA)** uses techniques inspired by Darwin's principle of natural selection such as recombination (also known as genetic crossover), mutation, selection (also known as survival of the fittest). The most popular type of EA is Genetic Algorithm **(GA). [5]**

3.1.5 Algorithmic complexity:

The complexity of any problem implies the estimation, in the worst case, of the number of instructions to perform in order to solve an instance of the problem. Algorithmic complexity is

used to measure the effectiveness of an algorithm for solving a problem. Indeed, the more the number of instructions is important the more the problem is complex and therefore the algorithmic complexity is proportional to the number of instructions of the problem. We deduce that the more complex the algorithm, the less efficient it is. Several types of complexity can be distinguished, we quote the constant complexity o (1) which is independent of the size of the problem, the logarithmic complexity o (log (n)), the linear complexity o (n), the quadratic complexity o (n2), the cubic complexity o (n3), the polynomial complexity o (np), the exponential complexity o (exp n), and finally the factorial complexity o (n!). [3]

According to the criterion of complexity, we distinguish essentially four classes of problems:

- Class P

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

- Class P-Complete

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

- NP class

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

- NP-hard class

This class is also called NP-hard. It brings together all the problems that the complexity is exponential or doubly exponential. This class of problem is distinguished by the difficulty of optimal or exact resolution of large bodies, in a polynomial time. In this regard, we are content to find solutions known as good quality in a polynomial time. Among the problems of this class, we quote tree partitioning problems, connectivity in a routing graph, the problem of vehicle tours is one of the problems of routing. Lenstra and Rinnooy Kan have proven that the VRP is an NP-HARD problem. The problem of vehicle tours is an NP-HARD problem. Its resolution by an exact method is inappropriate for large instances. It is therefore inevitable to proceed to its resolution by heuristic approaches, which provide feasible and appreciable solutions in a reasonable time. [3]

4 Conclusion

In this chapter in this chapter we tried to introduce what is the radiopharmaceuticals by selecting a few definitions and also we have presented the problem of the VRP brevity. Passing Scopes of definitions and theories, and we have listed the variants of vrp. In adding to that we have known that the problems of touring vehicles belong to the NP-hard problem. Their resolution can only be done in an approximate way, because to date, there is no algorithm allowing their resolution in a polynomial time. The mathematical form that will be developed in the next chapter are an alternative for problem solving of this kind.

Chapter 02
Problem description

1 Introduction

Every day, it is necessary to create delivery schedules. Current practice is the delivery schedules to be generated manually by experienced personnel in the radiopharmacies and as a result they are not optimized. This may result in two major issues: (a) the delivery of some orders may arrive at a customer location later than the actual injection time specified in the contract, resulting in extra cost for the pharmaceutical company as it needs to replace the order free of charge in a later time or day, and (b) since the delivery routes are not optimal they may cost more to the pharmaceutical company simply because the vehicles travel longer distances. Apart from the increased monetary cost, longer delivery routes result in the release of more greenhouse gas emissions which negatively affect the climate and human health.

The main contribution of our work is to address the above limitations by defining an efficient mathematical optimization model that determines the routes that are the most cost effective and guarantee that all doses arrive at a customer location before their injection times. To the best of our knowledge this is the paper that deals with the distribution of radiopharmaceuticals.

2 The Vehicle Routing Problem with Time Window

The vehicle routing problem with time windows (VRPTW) is a well-known NP-hard problem. Almost all VRPTW methods proposed are devoted to a static problem where all data are known before the route is constructed and do not change thereafter. The advancement of communication and information technology makes entrepreneurs more aware of the importance of just-in-time managerial strategies. In the past decade, express transshipment activities and e-commerce business have experienced a rapid growth. These developments have led to a gradual growth of a new class of problems, known as real-time routing and scheduling problems, where problem size and parameters change after the vehicle routes are constructed [10].

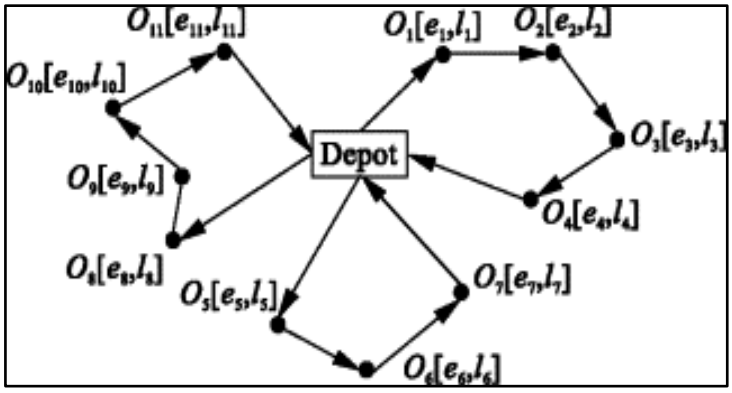


Figure 2.1: VRPTW from one production center to several imaging centers according to appointments. [33]

2.1 The Problem Description

We consider a single production center with multiple imaging centers where RPs are required. On the site of the production center there is also an imaging center. From this point on when we talk about facilities we mean the group of imaging center together with the production center. There are several examinations planned for each imaging center. When a patient arrives at the center, he/she will be injected with a single dose of RP which is within the proper range of radioactivity (measured in millicurie). Whenever it is not possible to meet an appointment, for example when delivering doses would be too costly, the appointment has to be rescheduled, for which we impose a failure cost. We will vary this failure cost and see how this influences the acquired solutions. Below we describe the several aspects of the problem in more detail. [15]

2.1.1 Production

The production center has multiple production lines that can work simultaneously. Using any particular production line requires a fixed fee to be paid, as well as a certain hourly cost. Each production line can produce up to some maximum batch size of dosages with a certain initial radioactivity within some timespan. These parameters can be different for each production line. Immediately after a dosage is produced it undergoes radioactive decay. As a simplifying assumption, we assume that the decay happens every thirty minutes. That is, thirty minutes after production a dosage differs by a certain factor from its initial radioactivity. [16] and as a note this part is not covered in this work it is just for future adding.

2.1.2 Distribution

After production, and in our phase that we interest in my studies, the dosages have to be distributed over the imaging centers. For this we can assign trucks to drive along certain routes, visiting multiple imaging centers along the way. These routes are not fixed and should also be determined. Using a vehicle requires a fixed fee to be paid, as well as a certain hourly and mileage cost. The travel times and distances between two imaging centers are assumed to be fixed and known, though we also consider the case of stochastic travel times. We assume that an imaging center is able to receive a shipment at any time and also multiple times a day. Whenever a shipment arrives, the dosages need to be unloaded for a certain time before both the dosages can be used and the vehicle can move on. In addition to this, we require the dosages to be fully unloaded at least thirty minutes before they are injected. [16]

2.1.3 Stochastic Travel Times

In addition to the deterministic problem, we also consider the case where with equal probability the travel time between two imaging centers takes the normal amount of time, a factor of one and a half as much time, or ten percent less time. By considering this case we can investigate the robustness of our deterministic schedules. [16]

3 The model

3.1 Parameters and Sets

n: number of medical imaging centers placing orders
V: the set of all nodes in the network, **V = {0, 1,… n}**
Vc: the set of all customer nodes in the network, **Vc = {1,… n}**
E: the set of arcs in the network, **E = {(i,j) :** $\forall i, j \in V$}
N: maximum number of available vehicles
Di: the total number of doses ordered by imaging center **i**
F: fixed cost for dispatching a vehicle
cij : the cost of traveling from node **i** to node **j**
dij : the distance of traveling from node i to node **j**
tij : the time of traveling from node **i** to node **j**
pi: the time a vehicle must arrive at a customer prior to the injection time
si: the service time needed by a driver to spend in an imaging center
W_{VEH}**:** the vehicle weight capacity of each available vehicle
W_{CON}**:** the weight of each container that seals a dose
W_i: the weight of the total number of doses ordered by customer **i**
$[e_I, l_i]$**:** the time window a dose must arrive in an imaging center
S: is the nominal time allocated for servicing one dose, typically 3 minutes

3.2 Variables

w_i**:** measures the weight a vehicle has delivered until it reaches customer **i**
xi : the arrival time of a vehicle to customer site **i**
yij : 1, if node i is visited immediately before node **j; 0** otherwise

3.3 Preprocessing

It is common that pharmaceutical companies outsource the delivery of the doses to logistics companies, which are responsible for providing the transportation vehicles together with the drivers and the load and unload equipment, there is a maximum number of vehicles (denoted by **N**) that the logistics company makes available to the pharmaceutical company. [8]

Mathematically the transportation of the doses to the medical imaging centers can be expressed as a general vehicle routing problem with time widows. Let **G =(V,E)** be a complete undirected graph where **V = {0, 1,… n}** represents the nodes of the graph and **E = {(i,j) :** $\forall i, j \in V$} is the set of edges connecting the nodes. The set of nodes consists of the radio-pharmacy, denoted by **0**, and the imaging centers. For convenience we will denote the set of the imaging centers as **Vc = {1, 2,… n}**. Also, throughout the paper we may refer to imaging centers as customers. Every imaging center places an order which consists of a number of doses. Let Di represent the number of doses ordered by the **i-th** imaging center. In addition, the **j-th** dose ordered by the **i-th** imaging center has an injection time, $\mathbf{T_{ij}^{INJ}}$, associated with it. All doses

ordered by a specific imaging center are delivered by the same vehicle. This means that the delivery vehicle must arrive at the imaging center before the earliest injection time, that is

$$T_i^{INJ} = \min \{T_{ij}^{INJ} : j=1, \ldots, D_i\}, \forall i \in V_c \quad (1)$$

Furthermore, every imaging center may require the delivery to arrive during a certain time window, [ei; li], where ei represents the earliest and li the latest arrival times. The latest arrival time may be a certain number of minutes, pi, prior to TINJ defined by (1). Therefore, the vehicle must arrive at the i-th imaging center no later than

$$l_i = T_i^{INJ} - pi \quad (2)$$

It is also possible that some imaging centers do not allow deliveries prior to a certain time. For example, an imaging center may not accept doses to be dropped off before the center opens for business. In this case **ei** will be set to the opening time of the imaging center. In the case where doses can be dropped off any time in the day (even when the imaging center is closed) we set **ei = 0**. The distance and the time it takes to drive between node i and node j are denoted by dij and **tij** , respectively. We obtain distances and drive times by using the geo-coding services ordered by Google [9]. The service can provide the distance and duration matrices of a network of any number of nodes. The distance and duration measures are not symmetric. This means that in general we have **dij ≠ dji** and **tij ≠ tji**. The cost, **cij** , of driving from a location i to a location j is defined as

$$C_{ij} = (m + f)d_{ij} + g \quad (3)$$

where the m is the cost of traveling one mile, f is the fuel surcharge that the logistics company asks for every mile traveled, and g is a flat amount charged by the drivers for every customer site they visit (typical value ranges of m, f and g are \$1.1-\$1.5, \$0.055-\$0.065, and \$10-\$15). Also there is a fixed cost, **F**, associated with every vehicle used.

The fleet of vehicles is homogeneous, meaning that all vehicles have the same weight capacity, denoted by W_{VEH}. During transportation to customer locations the doses are placed and sealed in lead or tungsten containers in order to minimize the radiation exposure. The weight of each container is denoted by W_{CON} (usually a container may weigh 32.5 lbs). Hence the total weight of the order of the i-th customer is defined by $W_i = D_i W_{CON}$. We assume that one vehicle will deliver all doses ordered by a customer, that is, we do not consider split orders. This means that all orders weigh less than the vehicles' capacity, i.e., $W_i \leq W_{VEH}$. If the order of the i-th customer weighs more than the vehicle's capacity, then this customer can be split into the appropriate number of dummy customers, **(i1,…, id)** so that each of them is assigned orders whose total weight is less than the vehicle capacity.

Once a vehicle arrives at a customer location the drivers need to spend some time unloading the containers, signing certain documents and picking up empty boxes. This is called service time of customer i and is denoted by **si**. The service time may be fixed for all customers **(e.g., 30 minutes)** or may be a function of the number of doses ordered by the corresponding imaging center (e.g., $s_i = D_i s$, where s is the nominal time allocated for servicing one dose, typically 3 minutes).

Besides the parameters, described above, we need to introduce a number of variables whose optimal values will be determined by the solution of the mathematical model, described in the next section. More specially, we use the binary variables **yij** to represent the order by which the network nodes are visited by the transportation vehicles, that is, **yij = 1**, if node i precedes node j, and **yij = 0** otherwise.
Since every vehicle has a maximum weight capacity, we are also interested in the total weight carried by it during the complete route. Hence we define the variable **wi** representing the total weight a vehicle has delivered until it has reached customer **i**. We also define the variable **xi** representing the arrival time of a vehicle to customer site **i** (note that, in the implementation of the model, **xi** is measured in minutes after midnight of the day of the delivery). Since the doses have to arrive at the customer before the dose with the earliest injection time we always have **xi** $\leq l_i$, where **i** is the latest time that a vehicle must arrive at a customer site and is defined in (2) .[8]

3.4 Constraints

3.5 Objective Function

The objective of the pharmaceutical company is to determine delivery route which will minimize the total transportation cost and guarantee that all orders reach the imaging centers before the specified injection time of each dose. [8] In the remaining of this section we summarize the variables and parameters used in the mathematical description of the optimization model presented in the following section.

$$\textbf{Min} \sum_{i \in V} \sum_{j \in V} y_{ij}\ c_{ij} + F \sum_{j \in V} y_{0j} \qquad (4)$$

3.6 Distribution

Here we will discuss all constraints related to the distribution part of the problem. In particular, we ensure that the following four constraints hold, which we will describe in detail below:

$$\textbf{S.t} \sum_{j=0, j \neq i}^{n} y_{ji} = 1 \ \forall i \in V_c \qquad (5)$$

$$\sum_{j \in V_c} y_{0j} \leq N \qquad (6)$$

$$W_i \leq w_i \leq W_{VEH}\ , \forall i \in V_c \qquad (7)$$

$$e_i \leq x_i \leq l_i \,,\ \forall i \in V_c \qquad (8)$$

4 Conclusion

In this chapter we have presented the problem of the VRPWT. Passing Scopes of Application and Mathematical Formulation, and we have listed the variants. But the problems of touring vehicles with time windows belong to NP-hard problem. Their resolution can only be done in an approximate way, there is no algorithm allowing their resolution in a polynomial time. The Meta-heuristic approaches that will be developed in the next chapter are an alternative for problem solving of this kind.

Chapter 03

The metaheuristic for the VRP problem

1 Introduction

The VRP is part of the combinatorial optimization problems so their resolution is quite delicate since the finite number of feasible solutions generally increases with the size of the problem, as well as its complexity. This has prompted researchers to develop many methods of resolution in operational research (OR) and intelligence artificial (IA). These approaches of resolution can be classified in two categories: the exact methods and the approximate methods. In this chapter presents the exact methods and then approximates then the algorithm chosen to solve the VRP problem.

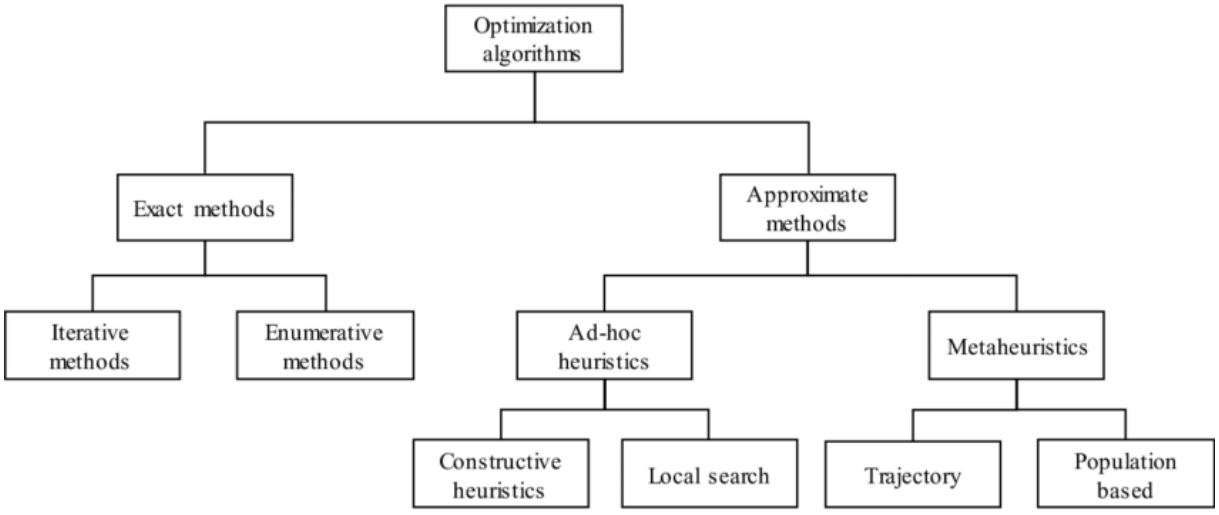


Figure 3.1: Exact methods and approximate methods. [26]

2 The exact methods

The exact methods seek to find the optimal solution in a certain way. Explicitly or implicitly examining the entire search space. However, several methods have been developed. They generally solve effectively up to 50 customers, sometimes more. method solving a problem containing 100 customers has been proposed in (Ralphs 2003). According to (Laporte and Nobert 1987) These generic methods generally of three categories based on a Branch and Bound, a Branch and Cut or on the dynamic programming. [24]

3 The approximate methods or heuristics

The approximate methods (i.e. Heuristics) make it possible to quickly find a feasible solution to a given problem. However, this solution is not necessarily the optimal solution. The goal of a heuristic is to find a solution as close as possible to that an exact method while being faster. The quality of an approximate method is therefore to be calculated in relation to the difference obtained between its solution and the optimal one. More this result is close to the optimal solution, the better is the heuristic. among these heuristics, we find Meta heuristics. [9]

3.1 Metaheuristics

The meta heuristic word is derived from the composition of two Greek words: meta, from Greek μ meaning beyond (or at a higher level) and heuristic. Indeed, these algorithms are want generic methods that can optimize a wide range of different problems, without requiring deep changes in the algorithm used. Meta heuristics are in general non-deterministic, they may not find the optimal solution, and again less prove the optimality of the solution found. We can distinguish the meta heuristics that evolve a single solution on the search space at each iteration and meta population-based heuristics of solutions. In general, meta heuristics based on unique solution are rather focused on exploiting the search space, so we are not never sure to get the optimum. Meta-heuristics based on population are rather exploratory and allow a better diversification of the research. [24]

3.1.1. Single-solution meta heuristics

In this section, we present meta-heuristics based on unique solution, also called trajectory methods. Unlike population-based meta-heuristics, single-solution meta heuristics begin with a single initial solution and move away, building a trajectory in the research space. Trajectory methods essentially include the descent method, the method of simulated annealing, taboo search, GRASP method, variable neighborhood search, iterated local search, and their variants. [21]

a) **Method of descent:** a descent direction algorithm is an algorithm differentiated optimization (the optimization we are discussing here is a branch of mathematical) intended to minimize a differentiable real function defined on a space Euclidean or, more generally, on a Hilbert space.

b) **Simulated annealing:** Simulated annealing adapted to continuous problems was used to solve the problem of segmentation by the parametric approach. Simulated annealing allows to approach near the optimal solution of the problem faster than an exploration exhaustive in the research area.

c) **Taboo search method:** taboo search is a meta heuristic of optimization presented by Fred Glover in 1986.there is often the name search with taboos in French. This method is an iterative meta heuristic described as local research in the broad sense.

d) **Variable Neighborhood Search:** (VNS) is a recent meta heuristic for the solving combinatorial and global optimization problems, whose basic idea is the systematic change of neighborhood within a local search. [21]

3.1.2. Meta heuristics with population of solutions

Unlike algorithms starting from a singular solution, meta heuristics with population of solutions improve at every iteration a population of solutions. We distinguish in this category, the evolutionary algorithms which are a family of algorithms derived from the theory of evolution by natural selection, stated by Charles Darwin in 1859, and swarm intelligence algorithms which, in the same way that evolutionary algorithms come from analogies with phenomena natural biological [9]

a) **the genetic algorithms:** the algorithms are optimization algorithms relying on techniques derived from genetics and mechanisms of evolution of the nature: crosses, mutations, selections, etc. ... they belong to the class of the algorithms evolutionary.

b) **Swarm intelligence:** Particle swarm optimization has been applied to the segmentation of images in 2005 thanks to Du Feng's work. The meta heuristic was used to maximize Shannon's two-dimensional entropy in order to segment Infrared images. The OEP algorithm implemented is the one developed by Kennedy.

c) **Ant colonies:** ant colony algorithms are algorithms inspired by the behavior of ants, or other species forming a super organism, and constitute a family of optimization meta heuristics. [9]

4 Genetic algorithms for the resolution of the VRP

Genetic Algorithms (GAs) are stochastic optimization algorithms based on the mechanisms of natural selection and genetics. They have been adapted to the optimization by John Holland, also the works of David Goldberg have largely contributed to their enrichment. The vocabulary used is the same as that of the theory of evolution and genetics, uses the term individual (potential solution), population (set of solutions), genotype (a representation of the solution), gene (part of the genotype), parent, child, reproduction, crossbreeding, mutation, generation, etc. [11]

4.1 Operating a genetic algorithm

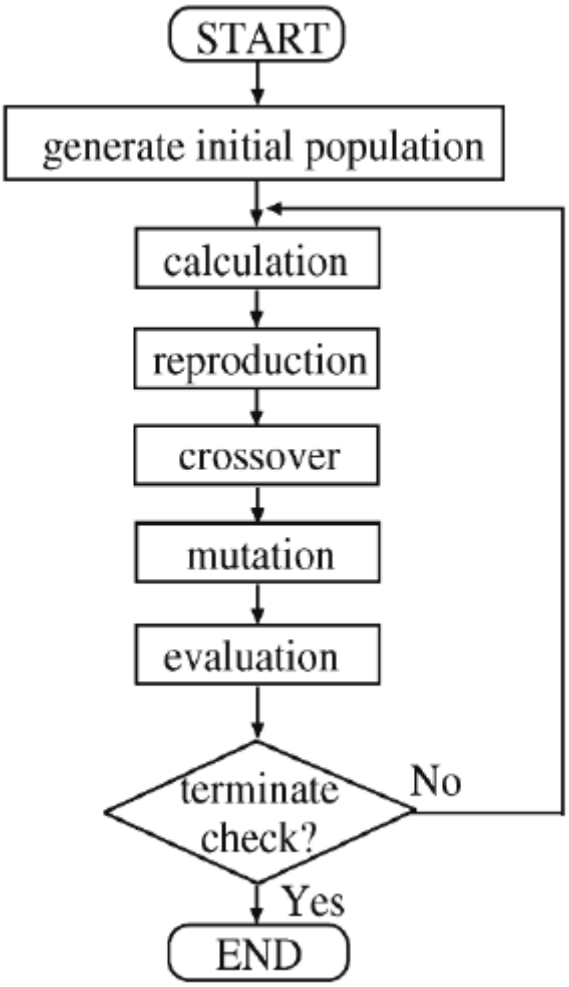


Figure 3.2: How a genetic algorithm works. [27]

Their operation is extremely simple, starting from a population of potential solutions (chromosomes) initial, arbitrarily chosen. We evaluate their relative fitness. On the basis of these performances we create a new population of potential solutions using simple evolutionary operators: selection, crossing and mutation. Some individuals reproduce, others disappear and only the best adapted individuals are expected to survive. This cycle is repeated until a satisfactory solution is found. Indeed, the genetic inheritance through the generations allows the population to be adapted and thus to meet the criterion. [25]

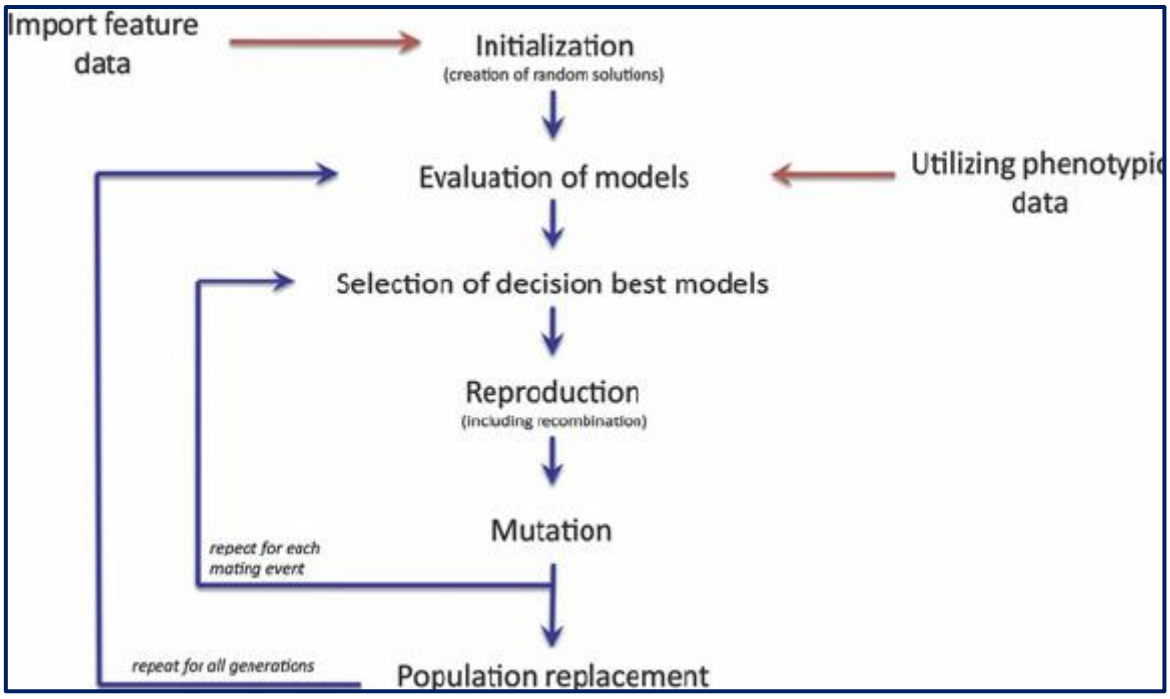


Figure 3.3 illustrates the main steps of a genetic algorithm. [28]

A genetic algorithm searches the extrema or extrema of a function defined on a space of data. Its implementation requires:

1. **START**
2. Generate the initial population
3. Compute fitness
4. Repeat
5. **Selection**
6. **Crossover**
7. **Mutation**
8. Compute fitness
9. until population has converged
10. **STOP**

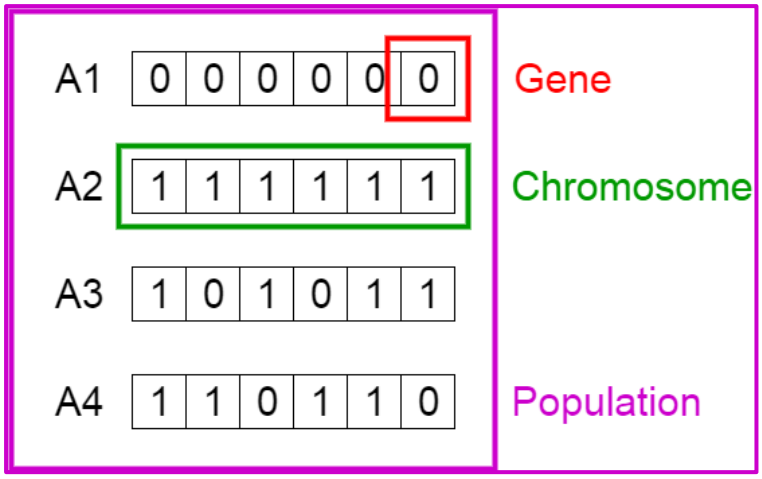


Figure 3.4: Population, Chromosomes and Genes. [34]

4.1.1 Coding of the data

The first step is to properly define and code the problem. This step associates with each point of the search space a specific data structure, called genotype or set of chromosomes, which will characterize each individual of the population. Coding of each individual in sequence is essential in developing a genetic algorithm which in particular depends on the implementation of transformation operators. So, this phase determines the data structure that will be used to encode the genotype of individuals Population. The coding must therefore be adapted to the problem addressed. [14]

4.1.2 The Creation of the Initial Population

The initial population will be created randomly provided that each individual in the created population is a solution of the problem. The size of the initial population must be reasonably large by taking into account both the quality of the solutions found and the execution time of our algorithm. [11]

4.1.3 Adaptation function (Fitness)

The evaluation of fitness is usually the stage in which one measures the performance of each individual. To judge the quality of an individual and thus compare it to others, a common measure of evaluation must be established. No rules exist to define this function, its calculation can thus be any, whether it is a simple equation or a linear function. In the vocabulary of genetic algorithms, we call "fitness" (English translation of the word adaptation) the criterion to be optimized. [14]

4.1.4 Selection

The selection operator is responsible for defining which individuals of p will be duplicated in the new population p 'and will serve as parents (application of the crossing operator) [30]. Let n be the number of individuals of p, we must select n / 2 (The crossing operator allows us to go back to n individuals). This operator is perhaps the most important since it allows individuals to population to survive, reproduce or die. As a general rule, the probability of survival of an individual will be directly related to its relative effectiveness within the population. There are basically four types of different selection methods: [14]

- **The Roulette Selection**

Parents are selected based on their performance. Best is the coded result by a chromosome, the greater its chances of being selected. We must imagine a kind of casino roulette on which are placed all the chromosomes of the population, the place given to each chromosome being related to its adaptation value. Then, the ball is launched and stops on a chromosome. The best chromosomes can be drawn several times and the worst are never selected. [14]

- **Selection by Rank**

The previous selection has problems when the adaptation value of the chromosomes vary enormously. If the best function of evaluating a chromosome represents 90% of the roulette, then the other chromosomes will have very little chance of being selected and we would arrive at a stagnation of evolution. Row selection sorts first the population by their scores. Then, each chromosome is associated a rank according to of his position. So the worst chromosome will have rank 1, the next 2, and so on to the best chromosome that will have the rank N. The selection by rank of a chromosome is the same as roulette, but the proportions are related to rank rather than to the value of the evaluation. With this method of selection, all chromosomes have a chance to be selected. However, it leads to a slower convergence towards the good solution. This is because the best chromosomes do not differ greatly from worse. [14]

- **The Tournament Selection**

It consists in randomly selecting two or more individuals and selecting the strongest. This process is repeated several times until N individuals are obtained. The advantage such a selection is to prevent a very strong individual from being selected more than once. [14]

- **Elitism**

When a new population is created, there is a good chance that the best chromosomes are lost after hybridization and mutation operations. To avoid this, we use the elitism method. It consists of copying one or more of the best chromosomes in the new generation. Then, we generate the rest of the population according to the usual reproduction algorithm. This method greatly improves the algorithms genetic, because it allows not to lose the best solutions. [21]

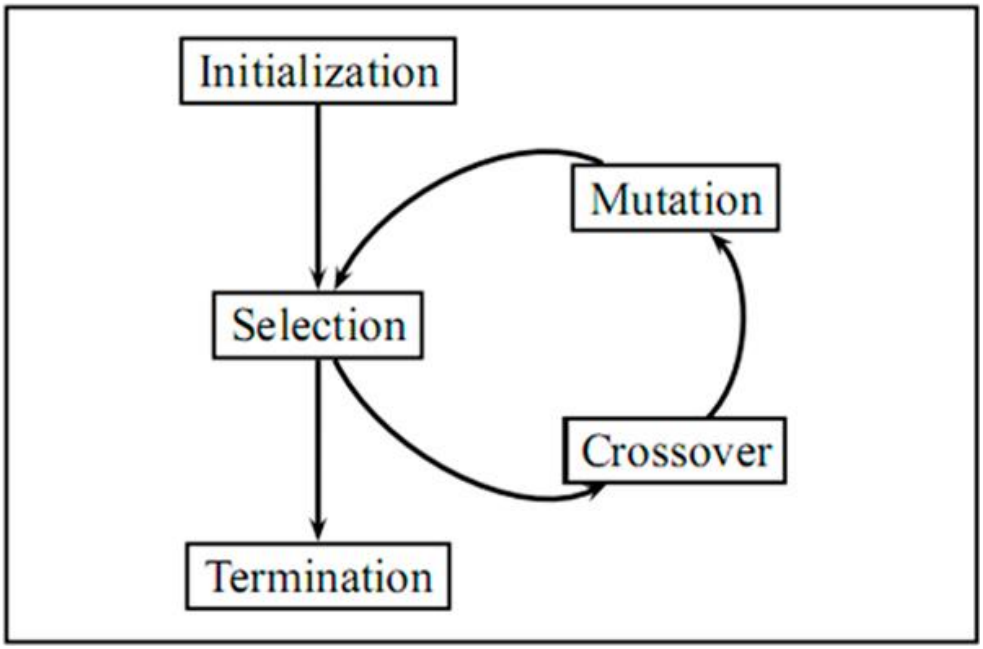


Figure 3.5: Tree Basic steps of GA: selection, crossover and mutation. [29]

4.1.5 the crossing

Once selected individuals, they are made to reproduce between them, for this we use the crossover operator. This is the essential operator for finding an algorithm genetic. He combines the genotypes of two individuals to obtain two new ones. With this operator, genotypes are seen as a string of binary numbers. Many crossing methods are used:

- Crossing at one point: we randomly select an identical break point on the two genotypes and exchange fragments after the cut-off point for give the two new genotypes.
- Two-point intersection: two crossing points are chosen randomly and one exchange the fragments between these two points.
- Crossing in k points: generalization at k cut-off points of the methods preceding. [14]

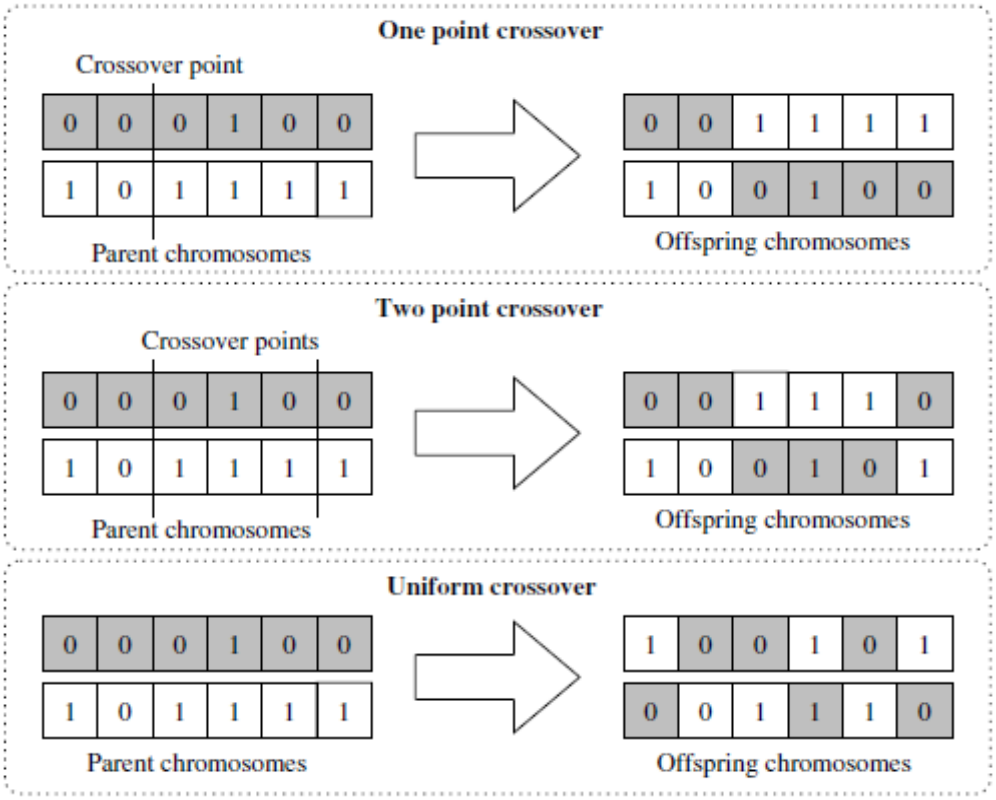


Figure 3.6: crossover work. [30]

4.1.6 the mutation

This operator consists in changing the allelic value of a gene with a very low probability pm, generally between 0.01 and 0.001. We can also take pm = 1 / lg where lg is the length of the string of bits encoding our chromosome. A mutation simply consists in the inversion of a bit (or several bits, but considering the probability of mutation is extreme rare) being in a very particular locus and also randomly determined; we can summarize the mutation as follows:

We use a function that is supposed to return true with a probability pm. [11]

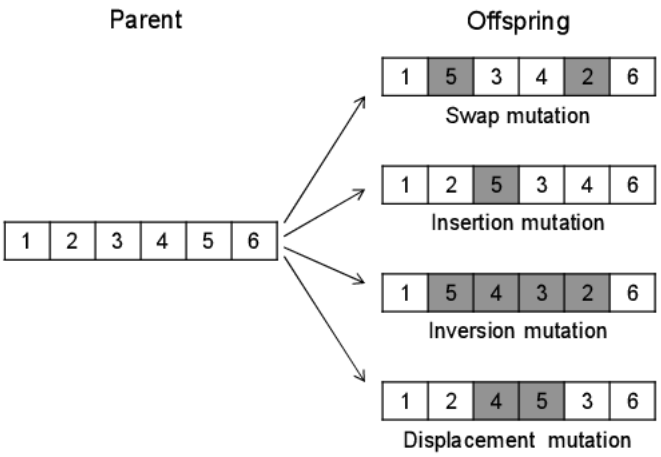


Figure 3.7: mutation work. [31]

Algorithm of mutation

For each locus to do

Call on the function

/ If this function returns true then

/ / we reverse the bit at this locus

/ End if

The mutation operator therefore modifies in a completely random way the features of a solution, which allows to introduce and maintain diversity within of our population of solutions. This operator plays the role of a "disruptive element", he introduces "noise" into the population. [11]

5 Advantages and disadvantages of genetic algorithms

5.1 Advantages

- Elimination of invalid solution.
- Allows you to deal with important research areas (a lot of solutions, no exhaustive path envisaged).
- Number of important solutions.

- Relativity of the quality of the solution according to the degree of precision required.

5.2 Drawbacks

- Need more calculations than other meta heuristic algorithms (especially the evaluation function).
- Parameters difficult to set (population size, % mutation).
- Choice of the delicate evaluation function.
- Not sure that the solution found is the best, but just an approximation of the optimal solution.
- Local optimums problem if parameters are poorly evaluated. [14]

6 Conclusion

In this chapter describes the different methods of problem solving optimization. We have found that the exact methods allow to reach the optimal solution, but they are too greedy in terms of computing time and memory space required. However, the approximate methods require reasonable search costs. But, they do not guarantee the optimality of the solution. Optimization by the genetic algorithm is a meta heuristic classified in the family of evolutionary methods. It presents a great number of peculiarities notably in its functioning and its flexibility in the VRP problem. In the next chapter we will present new methods and technologies for the global and optimal management of the logistics chain.

Chapter 04

Implementation of genetic algorithm and analysis of results

1 Introduction

In this chapter, we present the results of our experimentations. We start with the choice of the programming language, the description of the implemented software then the parameters of the genetic algorithm. The implemented genetic algorithm is tested on the problem successfully, reflecting the vrptw.

2 The programming environment

We developed the application using the programming environment: Microsoft Visual Studio Profissional 2017 ,Microsoft NET Framework version 4.7 under Windows 10 Operating System.

3 The programming language

We developed the application using the programming language C #. It has been available in successive beta versions since the year 2000 before being officially available in February 2002 at the same time as the Microsoft .NET 1.0 platform to which it is linked. C # can only work with this runtime environment. This makes available to the programs that run within it a very large set of classes. As a first approximation, we can say that the .NET platform is a runtime environment similar to a Java virtual machine. There are two differences, however:

Java runs on different OS (windows, unix, macintosh) since its inception. In 2002, the .NET platform ran only on Windows machines. In recent years the Mono project allows to use the .NET platform on OS such as Unix and Linux. The current version of Mono (February 2008) supports .NET 1.1 and elements of .NET 2.0.

The .NET platform allows the execution of programs written in different languages. It is sufficient that the compiler of those-cisache produce code IL (IntermediateLanguage), code executed by the virtual machine .NET. All .NET classes are available in .NET-compatible languages, which tends to erase differences between languages as programs use these classes extensively. Choosing a .NET language becomes a matter of taste rather than performance.

In 2002, C # was using the .NET 1.0 platform. C # was then largely a "copy" of Java and .NET a class library very close to that of the Java development platform. In the context of language learning, we went from a C # environment to a Java environment without being really out of place. There were even tools for converting source code from one language to another. Since then, things have changed. Each language and each development platform now has its own specificities. It is no longer as immediate to transfer one's skills from one domain to another…

C # and the . NET Framework 4.7 is included in Windows 10 Creators Update and Microsoft has added support in Visual Studio 2017 to target this version of the framework. You will be able to start creating .NET Framework 4.7 applications after installing Windows 10 Creators Update and Visual Studio 2017 Update, which has just been announced. You must select the .NET Framework 4.7 development tools included in the Visual Studio 2017 update. [33]

4 Application Architecture

In our application we developed (04) classes, whose (04) classes are grouped in a package that has the name "vrpw" (see figure 4.1). The last is the main class named "vrptw".

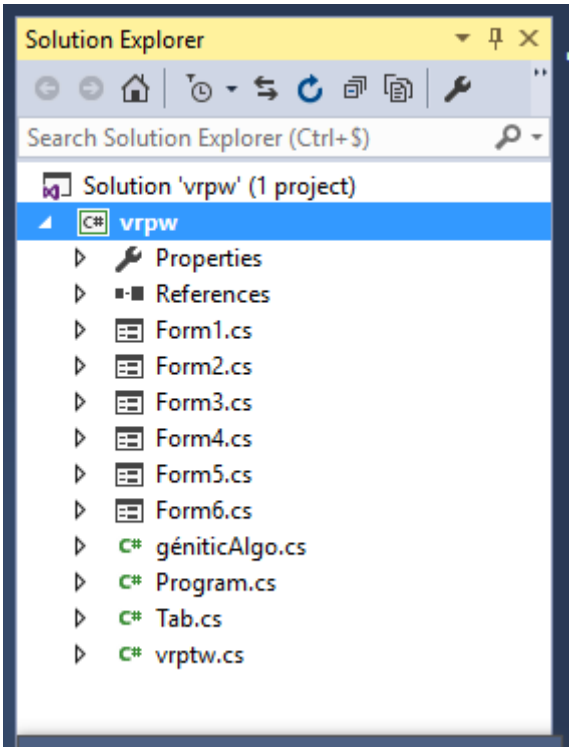


Figure 4.1: The vrpw package

4.1 The classes:

The class « vrpw »:

The class «**vrpw**» Is the principle class in our Project. It allows to manipulate the communication between classes.

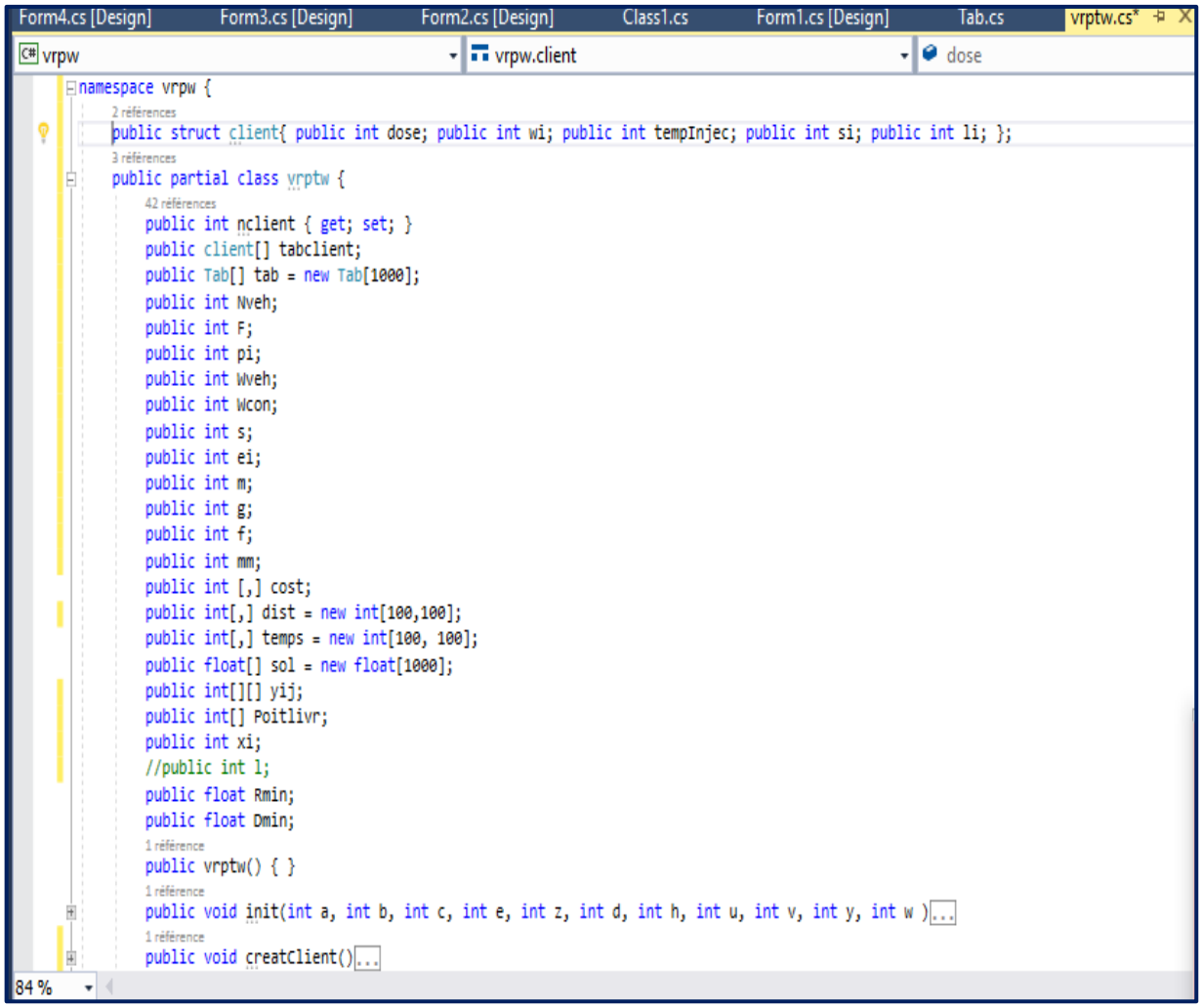


Figure 4.2: The class vrpw

The class « geneticAlgo »:

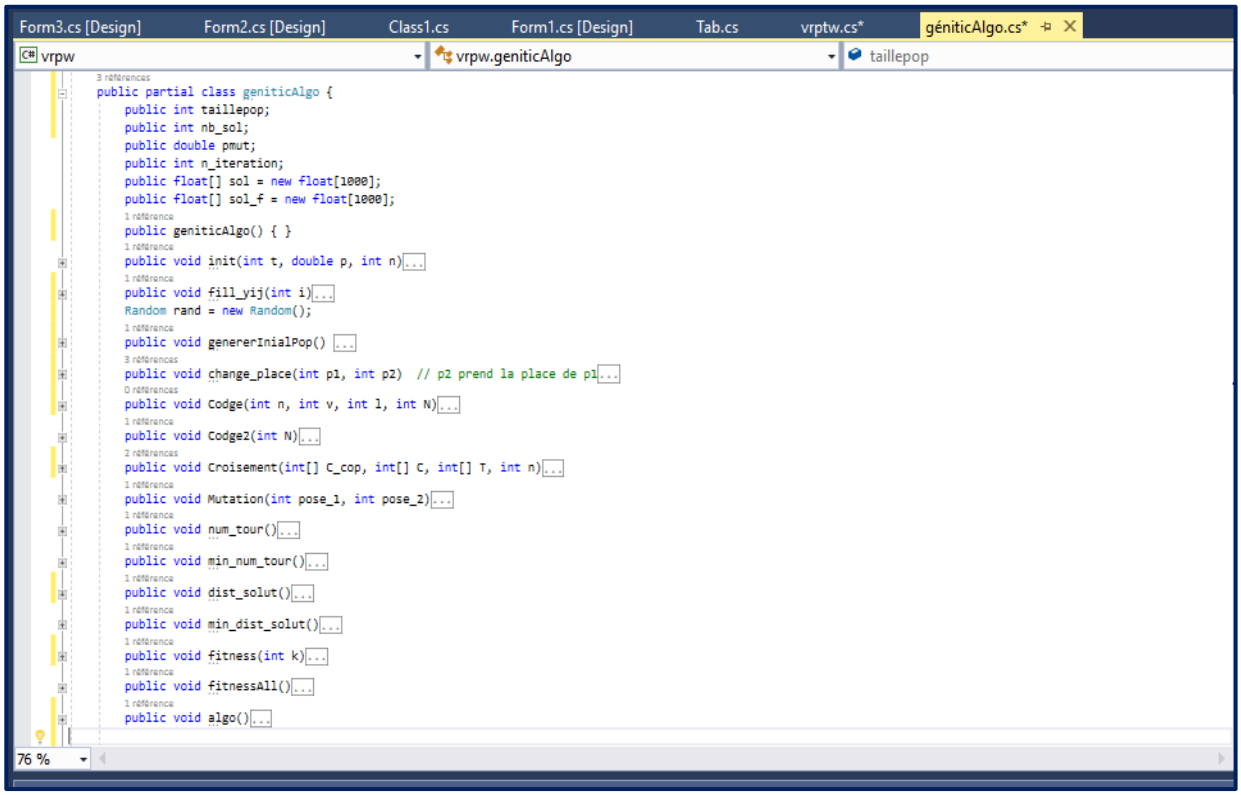


Figure 4.3: the functions of genetic class

The class «tab »:

This class contains 2 tables and 4 variables:

The table T: it is a number vector of the customers

The table V_cop: it is a vector which indicates for each customer the number of the vehicle which serves it.

The variable I: it is a counter that allows us to determine the box to which we want to apply (implement) the methods (set, get).

The variable R is the number of tours in the current solution.

The variableD_solu: the total distance in the current solution.

Variabledefitnesse: allows to measure the quality of an individual.

(See figure 4.4)

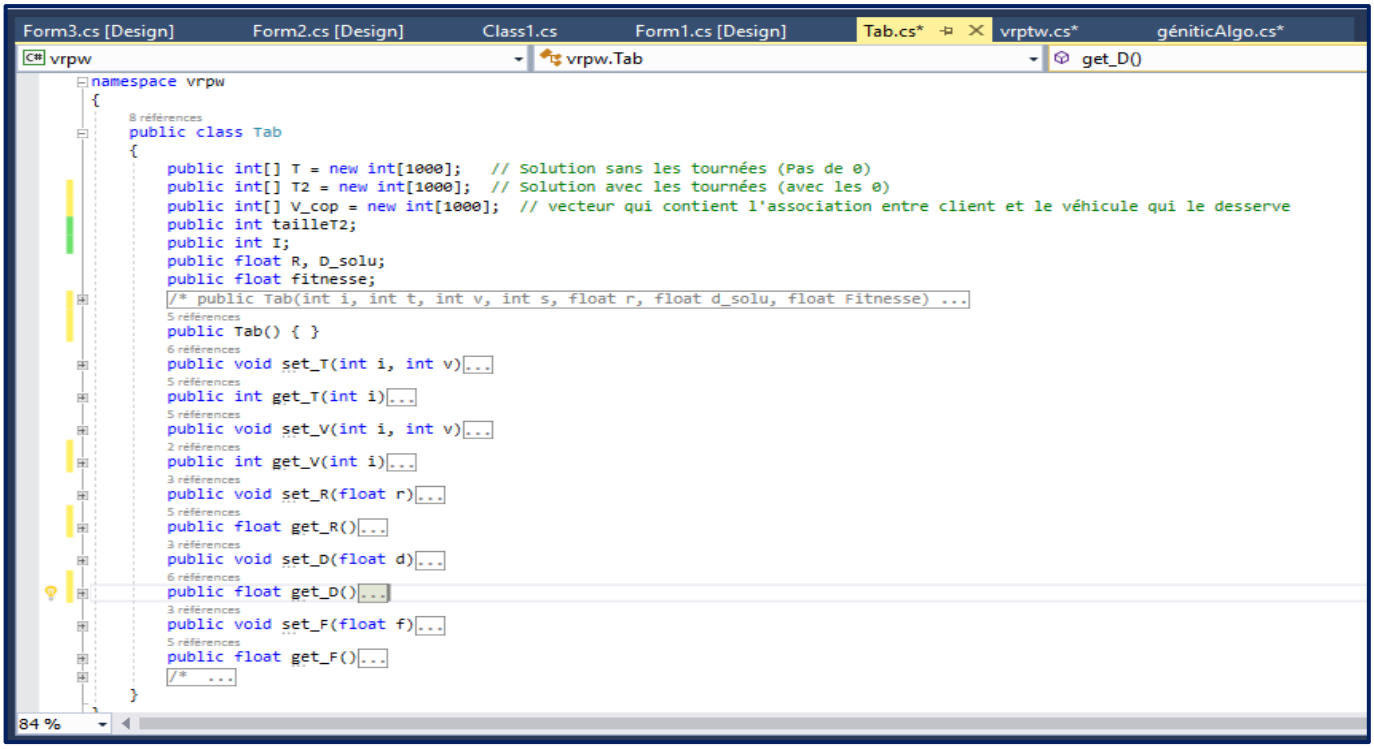


Figure 4.4: the class Tab

4.2 The instances:

From the instances of this class we can cite:

ga: This instance allows to manipulate the methods of genetic classe (calls the methods of genetic class)

tab: allows to create 2 tables T and V_cop the table T is the victor of numbers of clients and V_copT the victor of numbers of vehicles.

v: This instance allows to manipulate the methods of vrpw class.

4.3 Presentation of the interface

login window :

When we launching the application the following window appeared:

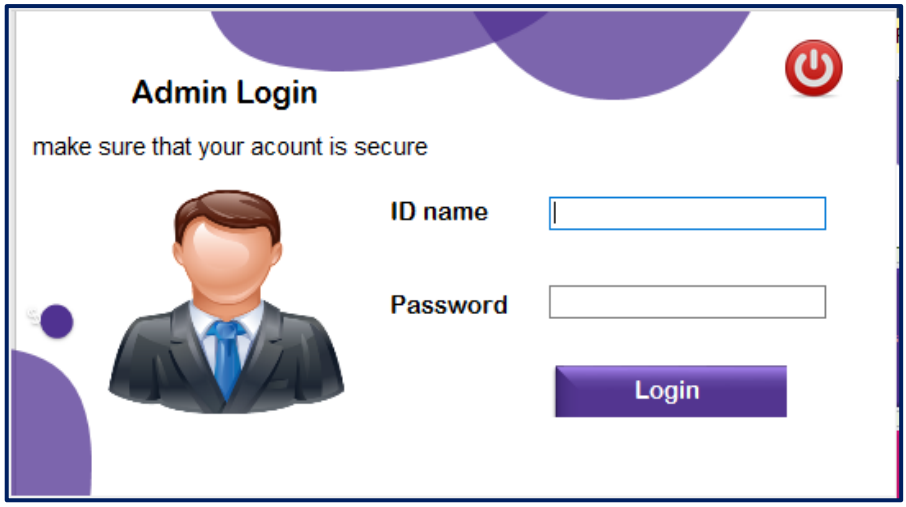


Figure 4 .5: login window

The main form:

The main components of this interface are:

Figuire 4.6: The principal interface

The insert parameters button:
allows to insert the parameters of vrptw and the algorithm parameters or even (see figure 4.7)

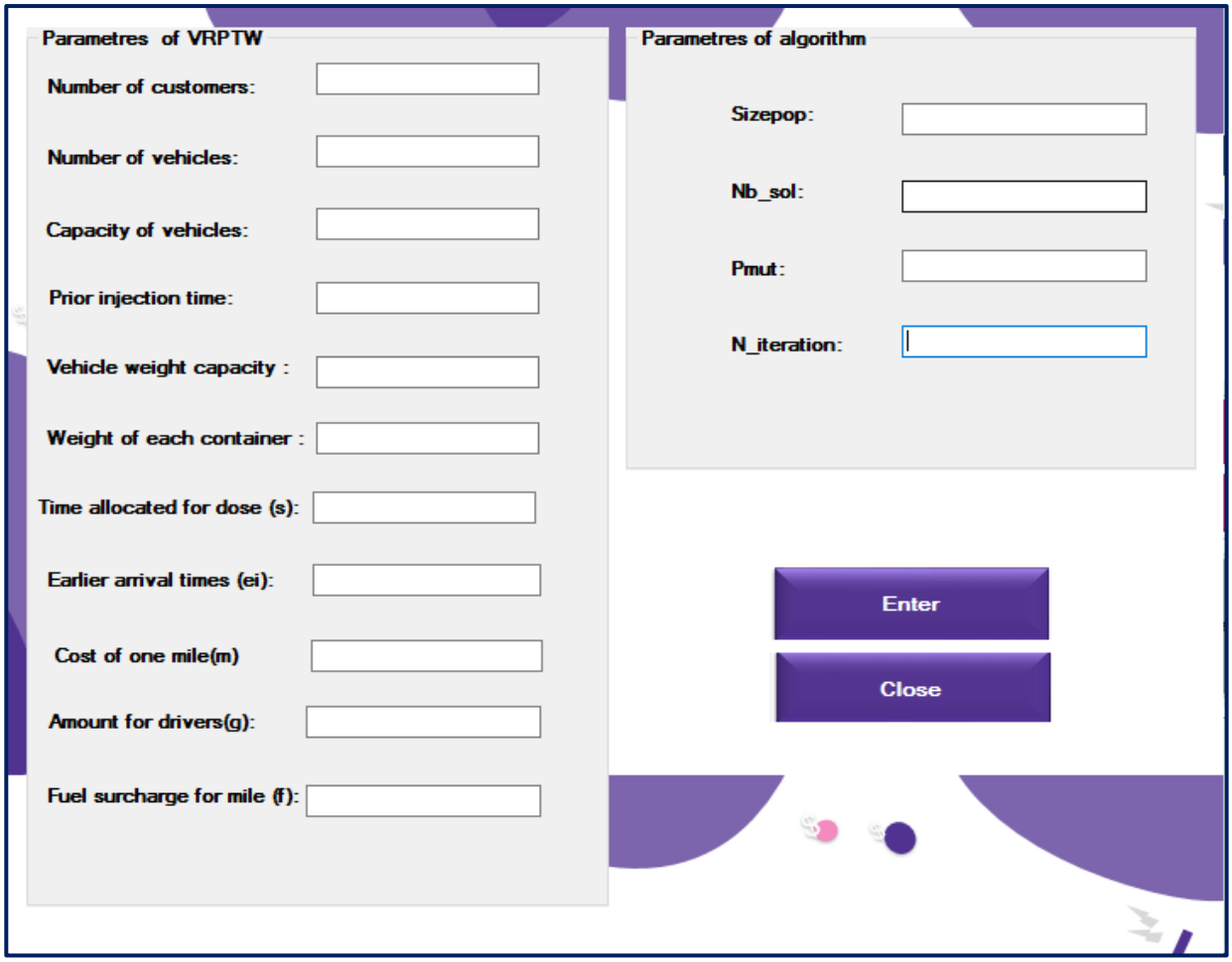


Figure 4.7: insert the parameters of problem

The button to insert the Customers' Requests:
the requested dosage and the total weight Delivered, time of injection, the service time and the latest time of the arrival (see figure 4.8)

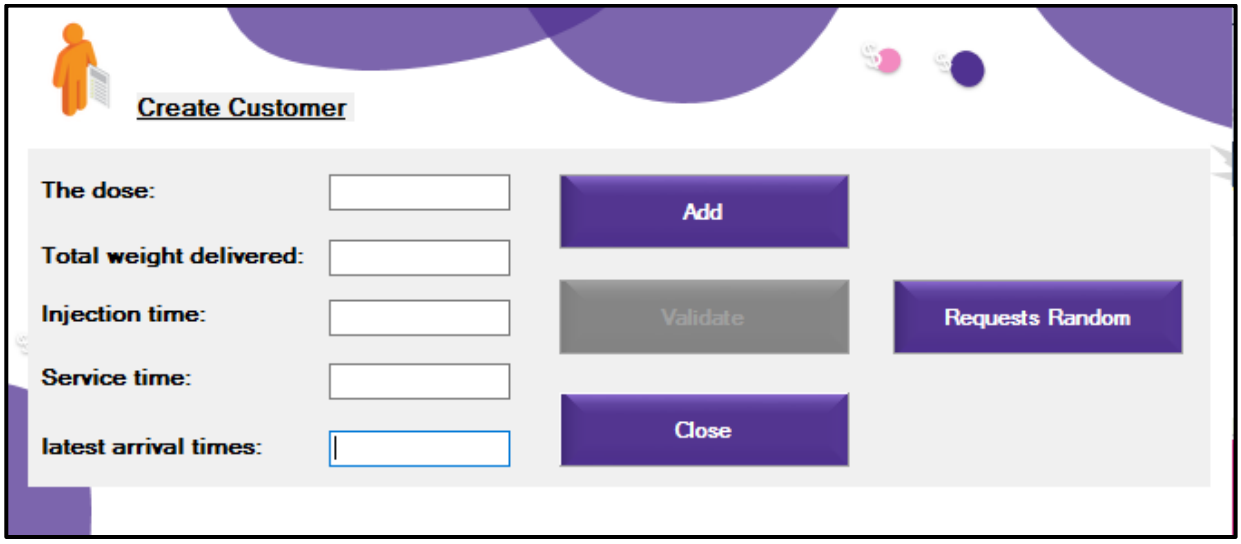


Figure 4.8: create a costumer

The Random Distance Matrix button:

allows you to insert the distances between the customers randomly (see Figure 4.9)

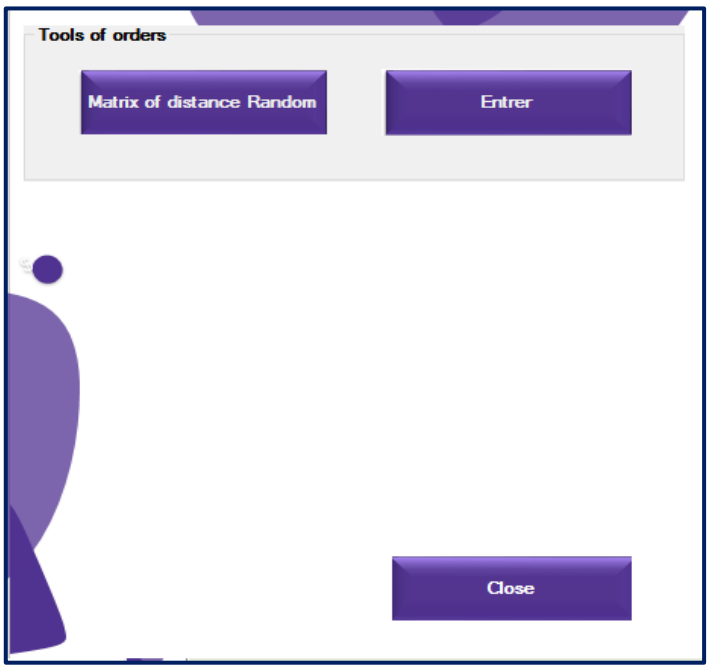


Figure 4.9: The insert buttons for The distance matrix

The Random Temps Matrix button:

allows you to insert the temps between the customers randomly (see Figure 4.10)

Figure 4.10: The insert button of The time matrix

5 Implementation of the Genetic Algorithm:

5.1 Initial population:

For the application of the algorithm we have chosen to take an initial population of individuals or each is representative by a path, the generation of a population is done randomly. In the literature the often adapted size of the population is taken to be double the number of client. The written method below

```
Random rand = new Random();
1 référence
public void genererInialPop() {
int[] client_cop = new int[100];
    for (int i = 0; i < taillepop; i++)
    {  int c, n_cop, j;
       n_cop = Form1.v.nclient;
        for (j = 0; j <= Form1.v.nclient; j++) client_cop[j] = j + 1;
        for (j = 0; j < Form1.v.nclient; j++)
        { c = rand.Next(0, n_cop - 1);
            Form1.v.tab[i] = new Tab();
            Form1.v.tab[i].set_T(j, client_cop[c]);
            for (int k = c; k < n_cop; k++)
            {
                client_cop[k] = client_cop[k+1];
            }
            n_cop--;

        }
    }
}
```

Figure 4.11: The Initial Population Method

5.2 The crossover operator:

Creates new combinations representing the new chromosomes. We chose in our case the use of the operator Order Crossover (OX), The crossing method is written below

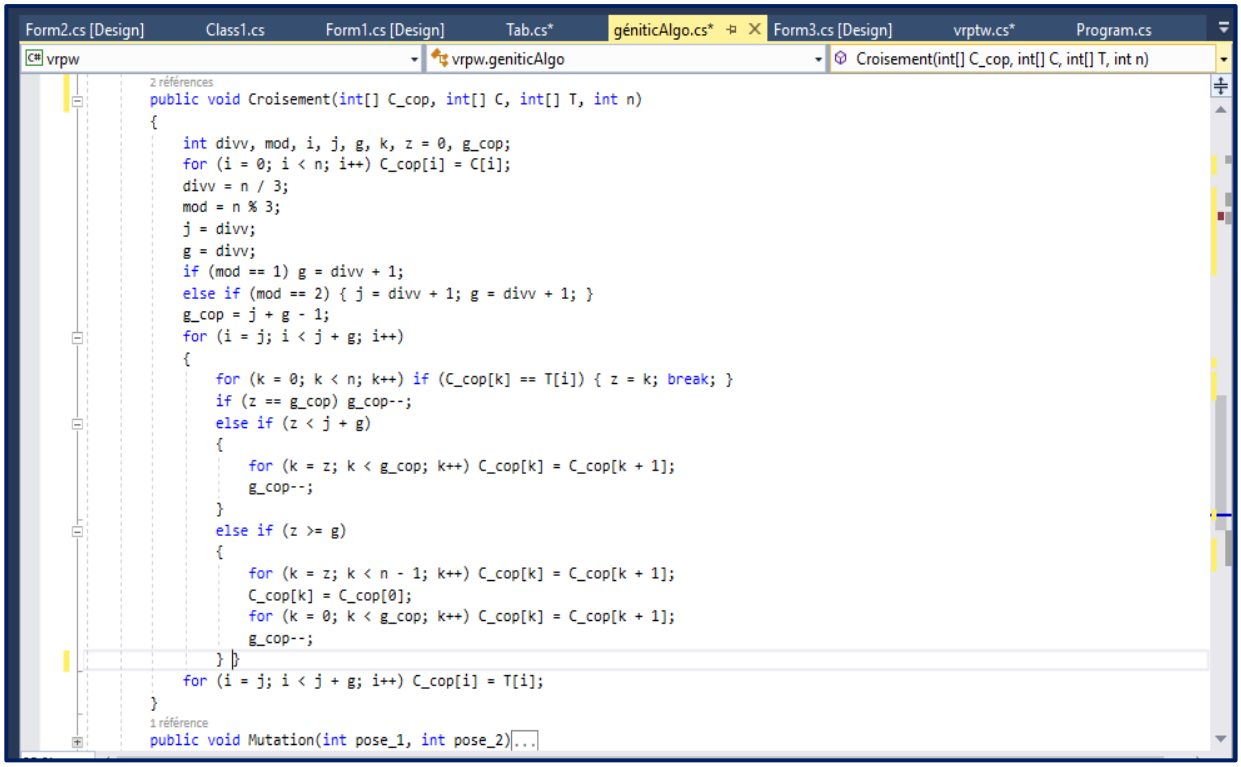


Figure 4.12: The crossing method

5.3 The operator of mutation:

The mutation operator is an operator unary, generating a single individual from another and for our application we use mutation by permutation,The mutation method is written below

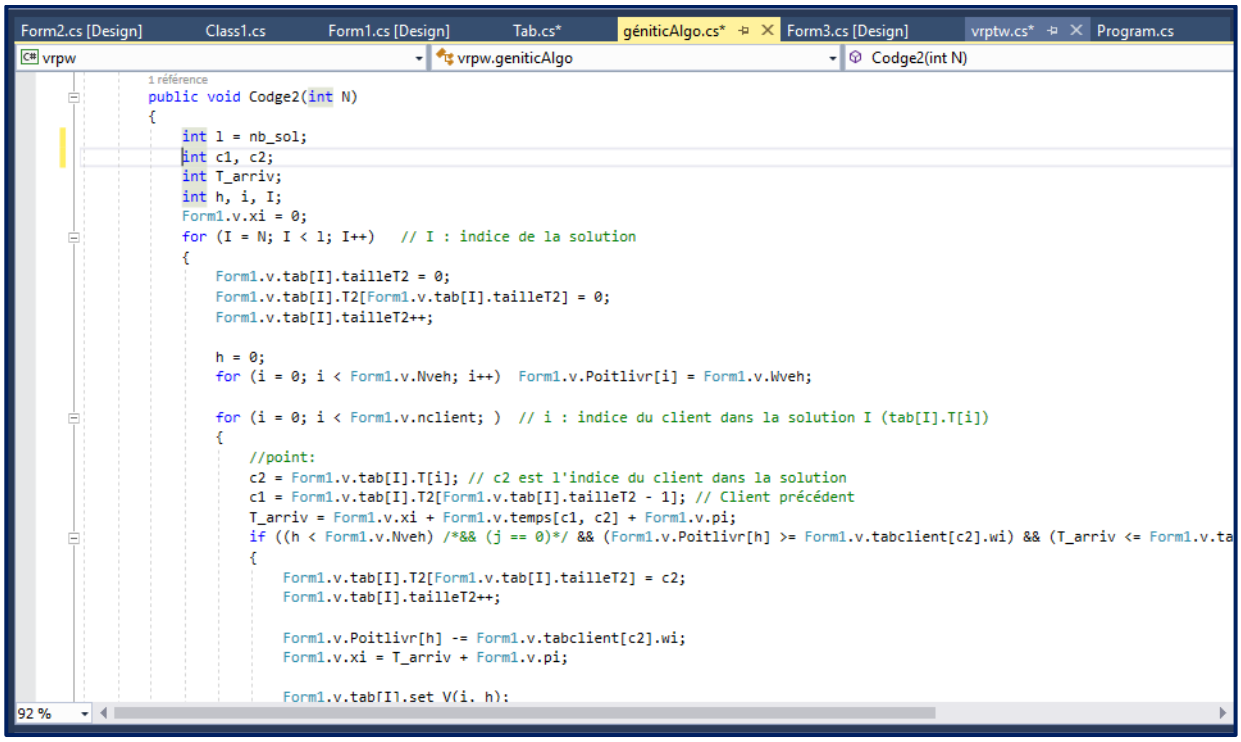
```
public void Mutation(int pose_1, int pose_2)
{
    int p1, p2, c, i;
    int[] T = new int[100];

    p1 = rand.Next(1, Form1.v.nclient - 1);
    do p2 = rand.Next(1, Form1.v.nclient - 1);
    while (p1 == p2);
    for (i = 0; i < Form1.v.nclient; i++) T[i] = Form1.v.tab[pose_1].get_T(i);
    c = T[p1];
    T[p1] = T[p2];
    T[p2] = c;
    for (i = 0; i < Form1.v.nclient; i++) Form1.v.tab[pose_2].set_T(i, T[i]);
}
```

Figure 4.13: The Method for the Mutation with permutation

5.4 Coding:

A chromosome represents an element of the research space The coding method is written below



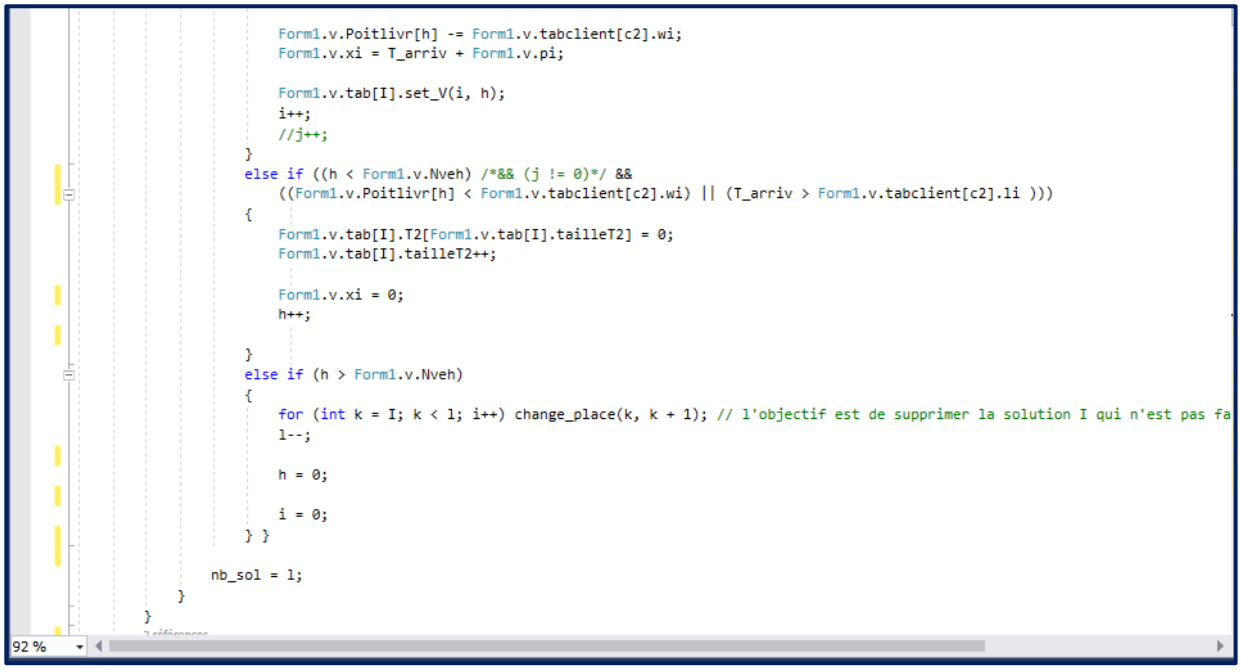


Figure 4.14: The coding method

5.5 The Filling method:

this method allows to fill the Yij matrices, so we can put the tours. filling method is written below

```
public void fill_yij(int i)
{
    for (int k1 = 0; k1 <= Form1.v.nclient; k1++)
        for (int k2 = 0; k2 <= Form1.v.nclient; k2++)
            Form1.v.yij[k1][k2] = 0;
    for (int k1 = 0; k1 < Form1.v.tab[i].tailleT2-1; k1++)
    {
        Form1.v.yij[Form1.v.tab[i].T2[k1]][Form1.v.tab[i].T2[k1+1]] = 1;


    }

}
```

Figure 4.15: The filling method

5.6 The num-tour method:

The method of calculating the numbers of rounds in a solution is written below

```
public void num_tour()
{
    float y = 1;
    int z = 0, i, I;
    for (I = 0; I < nb_sol; I++)
    {
        z = Form1.v.tab[I].V_cop[0];
        for (i = 0; i < Form1.v.nclient; i++)
        {
            if (z != Form1.v.tab[I].V_cop[i])
            {
                y++;
                z = Form1.v.tab[I].V_cop[i];
            }
        }
        Form1.v.tab[I].set_R(y);
        y = 1;
    }
}
1 référence
```

Figure 4.16: The method of calculating the number of rounds in a solution

5.7 The minimum tour solution method:

The method of calculation the minimum number of tours in the population is written below

```
public void min_num_tour()
{
    int i;
    Form1.v.Rmin = Form1.v.tab[0].get_R();
    for (i = 1; i < nb_sol; i++)
        if (Form1.v.Rmin > Form1.v.tab[i].get_R()) Form1.v.Rmin = Form1.v.tab[i].get_R();

}
```

Figure 4.17: The method of calculation the minimum number of tours in the population

5.8 The distance method

The method of calculating the distance in each solution is written below

```
public void dist_solut()
{
    int p, i = 0, I, z;
    float Dist = 0;

    for (I = 0; I < nb_sol; I++)
    {
        p = 0;
        z = Form1.v.tab[i].V_cop[0];
        for (i = 0; i < Form1.v.nclient; i++)
        {
            piont:
            if (z == Form1.v.tab[I].V_cop[i])
            {
                Dist += Form1.v.cost[p, Form1.v.tab[I].T[i]];
                p = Form1.v.tab[I].T[i]; }
            else
            {
                Dist += Form1.v.cost[p, 0];
                z = Form1.v.tab[I].V_cop[i];
                p = 0;
                goto piont;
            }
        }
        Dist += Form1.v.cost[p, 0];
        Form1.v.tab[I].set_D(Dist);
        Dist = 0;
    }

}
```

Figure 4.18: The method of calculating the distance in each solution

5.9 The minimum distance solution method:

The method of calculating the minimum distance in the population is written below

```
public void min_dist_solut()
{

    int i;
    Form1.v.Dmin = Form1.v.tab[0].get_D();
    for (i = 0; i < nb_sol; i++)
        if (Form1.v.Dmin > Form1.v.tab[i].get_D()) Form1.v.Dmin = Form1.v.tab[i].get_D();

    Form1.v.sol[Form1.v.mm] = Form1.v.Dmin;
}
```

Figure 4.19: The method of calculating the minimum distance in the population

5.10 The Change place method:

The method of changing the places of solutions in the population is written below

```
3 références
public void change_place(int p1, int p2)  // p2 prend la place de p1
{
    int b, c;
    float K;
    for (b = 0; b < Form1.v.nclient; b++)
    {
        c = Form1.v.tab[p1].get_T(b);
        Form1.v.tab[p1].set_T(b, Form1.v.tab[p2].get_T(b));
        Form1.v.tab[p2].set_T(b, c);
    }
    for (b = 0; b < Form1.v.nclient; b++)
    {
        c = Form1.v.tab[p1].get_V(b);
        Form1.v.tab[p1].set_V(b, Form1.v.tab[p2].get_V(b));
        Form1.v.tab[p2].set_V(b, c);
    }
    K = Form1.v.tab[p1].get_R();
    Form1.v.tab[p1].set_R(Form1.v.tab[p2].get_R());
    Form1.v.tab[p2].set_R(K);
    K = Form1.v.tab[p1].get_D();
    Form1.v.tab[p1].set_D(Form1.v.tab[p2].get_D());
    Form1.v.tab[p2].set_D(K);
    K = Form1.v.tab[p1].get_F();
    Form1.v.tab[p1].set_F(Form1.v.tab[p2].get_F());
    Form1.v.tab[p2].set_F(K);
}
0 références
```

Figure 4.20: The method of change places of solutions in the population

5.11 Fitness evaluation:

We measure the languor of each path and we assign each individual a score proportional languor measured as a distance Euclidean. The best individual corresponds to the individual with the the shortest path languor. The fitness method is written below

```
public void fitness(int k)
{
    fill_yij(k);
    int somCost = 0;
    for (int i = 0; i < Form1.v.nclient; i++)
    {
        for (int j = 0; j < Form1.v.nclient; j++)
        {
            somCost += Form1.v.yij[i][j] * Form1.v.cost[i, j];
            if (i == 0) somCost += Form1.v.yij[i][j] * Form1.v.F;
        }
    }
    Form1.v.tab[k].set_F(somCost);
}
```

Figure 4.21: The fitness method

5.12 FitnessAll evaluation method:

```
public void fitnessAll()
{
    num_tour();
    min_num_tour();
    dist_solut();
    min_dist_solut();

    for (int i = 0; i < nb_sol; i++) fitness(i);

    for (int i = 0; i < nb_sol; i++)
        for (int j = i + 1; j < nb_sol; j++)
            if (Form1.v.tab[i].get_F() > Form1.v.tab[j].get_F()) change_place(i, j);
}
```

Figure 4.22: The fitnessAll method

5.13 the calling method:

```
public void algo()
{   int m = 0;
    int b;
    genererInialPop();
    int[] T = new int[1000];
    int[] T1 = new int[1000];
    int[] T2 = new int[1000];
    Random myrand = new Random();
    int num = taillepop, l, k;
    nb_sol = taillepop;
for (int I = 0; I < n_iteration; I++)   // I est le nombre d'iterations
    { if (I > 0) { num = nb_sol; } /*** coisement ***/
        l = num;
        for (int i = 0; i < num; i++)  // i : est l'indice de la solution (Parcourt de toutes les solutions et Croisement + mutation
        {
     if (I == 0)
            {   Form1.v.tab[l] = new Tab();
                Form1.v.tab[l + 1] = new Tab();
                Form1.v.tab[l + 2] = new Tab(); }
            do
            {
                k = myrand.Next(1, num - 1);
            } while (k == i);

            for (b = 0; b < Form1.v.nclient; b++)
            {
                T1[b] = Form1.v.tab[i].get_T(b);
                T2[b] = Form1.v.tab[k].get_T(b);
            }
            Croisement(T, T1, T2); // Croisement entre la solution i (T1) et k (T2)
```

```
                Croisement(T, T1, T2); // Croisement entre la solution i (T1) et k (T2)
                for (b = 0; b < Form1.v.nclient; b++) Form1.v.tab[l].set_T(b, T[b]);
                Croisement(T, T2, T1);
                for (b = 0; b < Form1.v.nclient; b++) Form1.v.tab[l + 1].set_T(b, T[b]);
                Mutation(i, l + 2);
                l += 3;
            }
            if (I == 0) b = 0;
            else b = num;
            nb_sol = l;
            //Codge2(b);
            Codge2(0);
            /*** Fitness ***/
            fitnessAll();
            /*** choisir solusoin fitnennes ***/
            sol_f[m] = Form1.v.tab[0].get_F();
            sol[m] = Form1.v.tab[0].get_D();
            m++;
        }

        Codge2(0);
        //f.Close();

    }

}
}
```

Figure 4.23: the calling method

The algorithm stops after a certain number of iterations or generations, set or leave.

Stopping the algorithm

6 Results

In this part, we present the obtained results. These results were obtained by applying the developed genetic algorithm.

To solve this problem, we have used data managed randomly as follows:

In this test we use 4, 5,6,7 customers keeping the constant number of vehicles (the vehicles have the same capacity) and the algorithm parameters with the following precise values:

- Number of initial solutions "20"
- Number of iteration "40"
- Population size "30"
- Pmutation "5"
- ✓ **NOTE:** Each time you enter the algorithm parameters, you must keep the same values (20, 40,5,30).

The Amount sum t of Requested Quantities <= Capacity of Vehicles <= Deposit Capacity.

6.1 Parameters of problem:

Appoint mentS	Number of clients(n)	Number of vehicles(Nv)	Capacity of vehicles (Wveh)	Prior injection time (pi)	the vehicle weight capacity (depot)	the weight of each container (Wcon)	time allocated for servicing one dose (s)
A1	4	2	20	5min	20	2	3min
A2	5	3	20	5min	20	2	3min
A3	10	3	20	5min	20	2	3min
A4	50	9	20	5min	20	2	3min

opening time of the imaging center(ei)	cost of traveling one mile(m)	An amount charged by the drivers(g)	fuel surcharge for every mile (f)	Population size (sp)	Number of initial solutions (n-sol)	Pmutation (pmut)	Number of iteration (s)
0	1£	5£	1£	20	40	5	30
0	1£	5£	1£	20	40	5	30
0	1£	5£	1£	20	40	5	30
0	1£	5£	1£	20	40	5	30

Table 4.1: a test values for the parameters of problem

Example for A1:

The dosage	the total weight delivered	Injection n time	service time	latest arrival times
4	8	50	10min	5
3	6	140	10min	5
2	4	170	10min	5
4	8	230	10min	5

Table 4.2: a test values for the parameter of customers

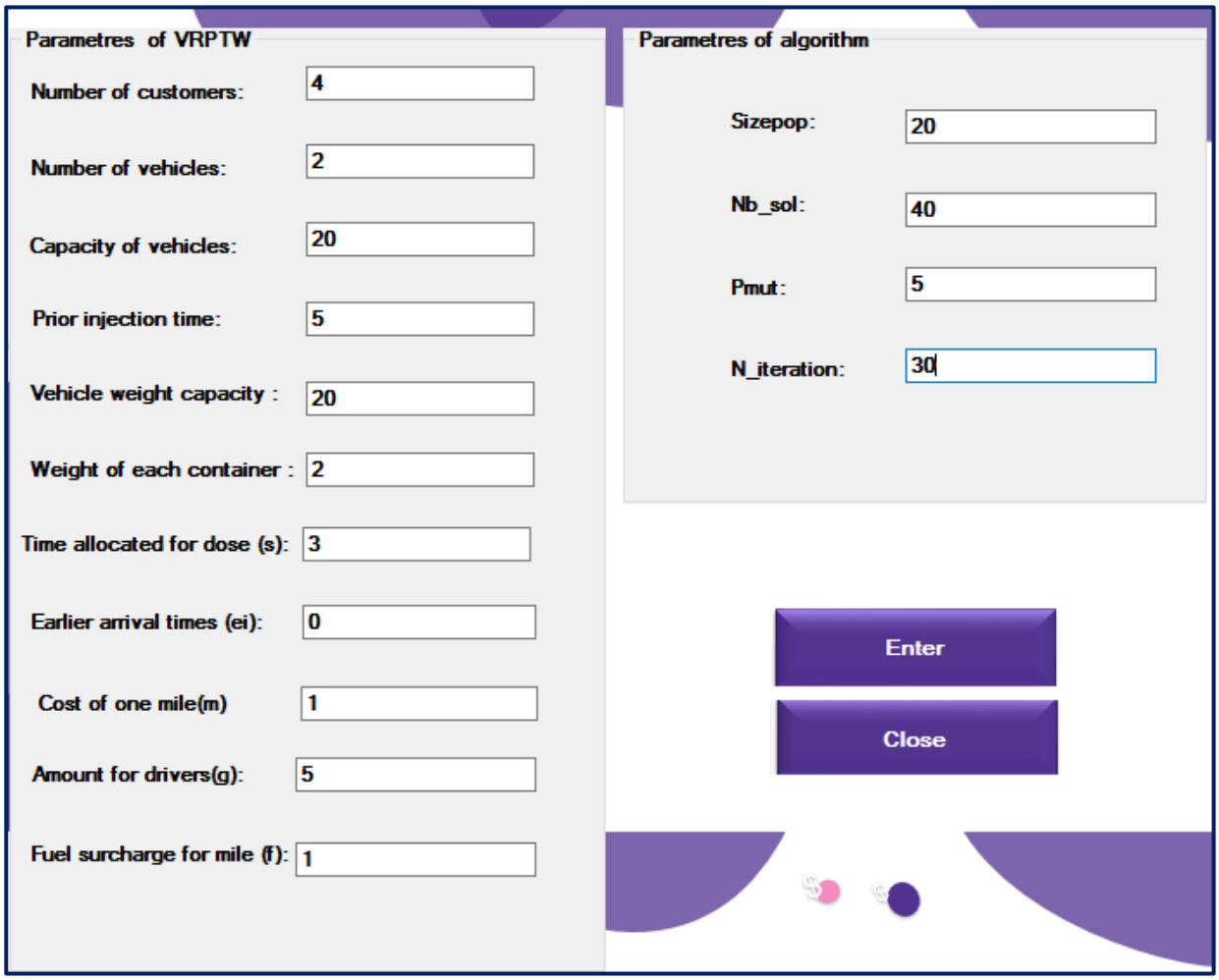


Figure 4.24: parameters of problem for Appointment1

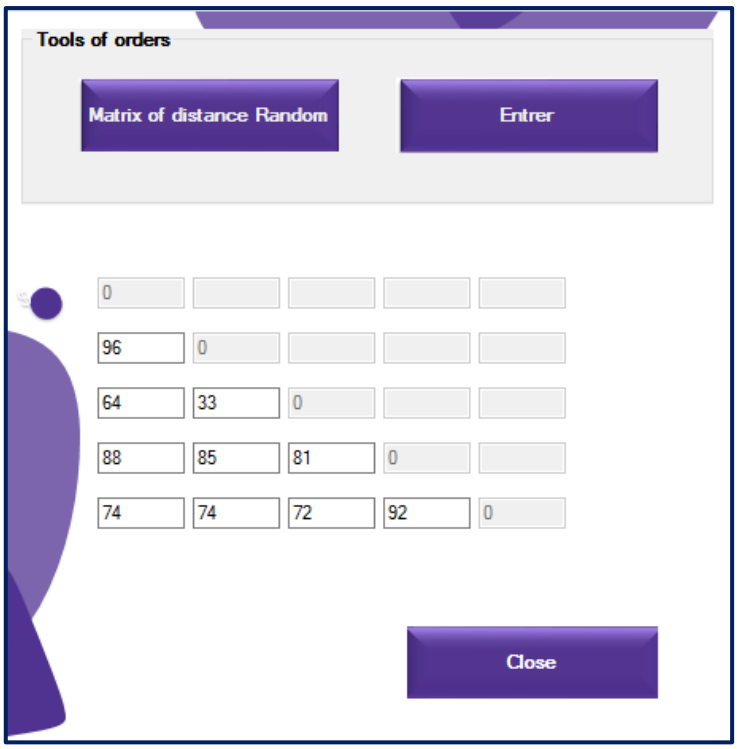


Figure 4.25: random distance matrix for appointment1

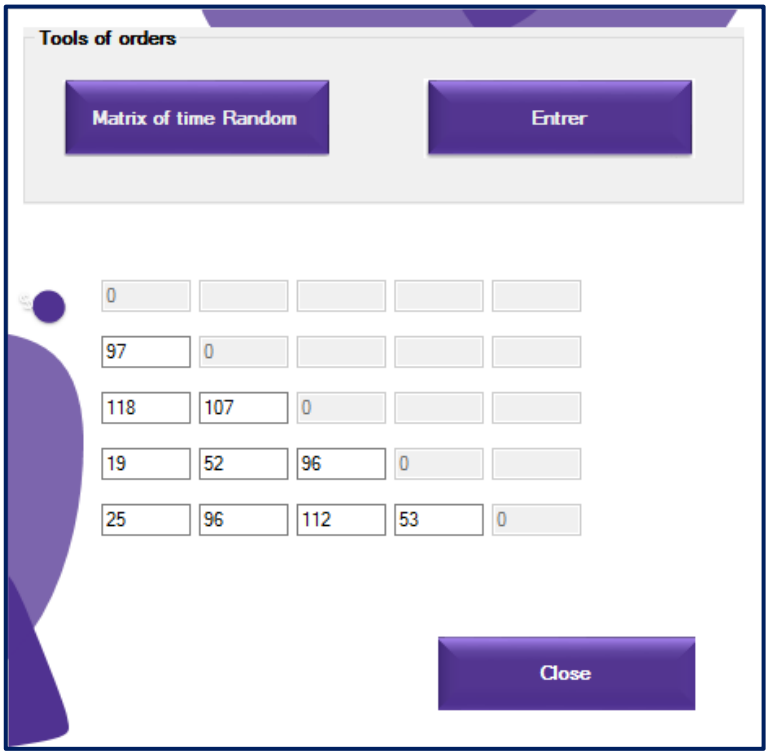


Figure 4.26: random time matrix for appointment1

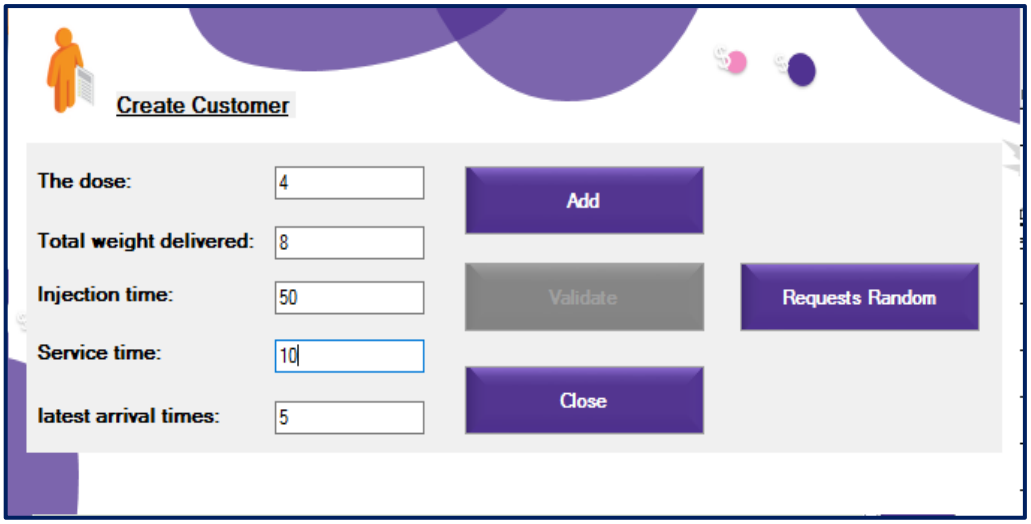


Figure 4.27: customer's parameters for appointment1

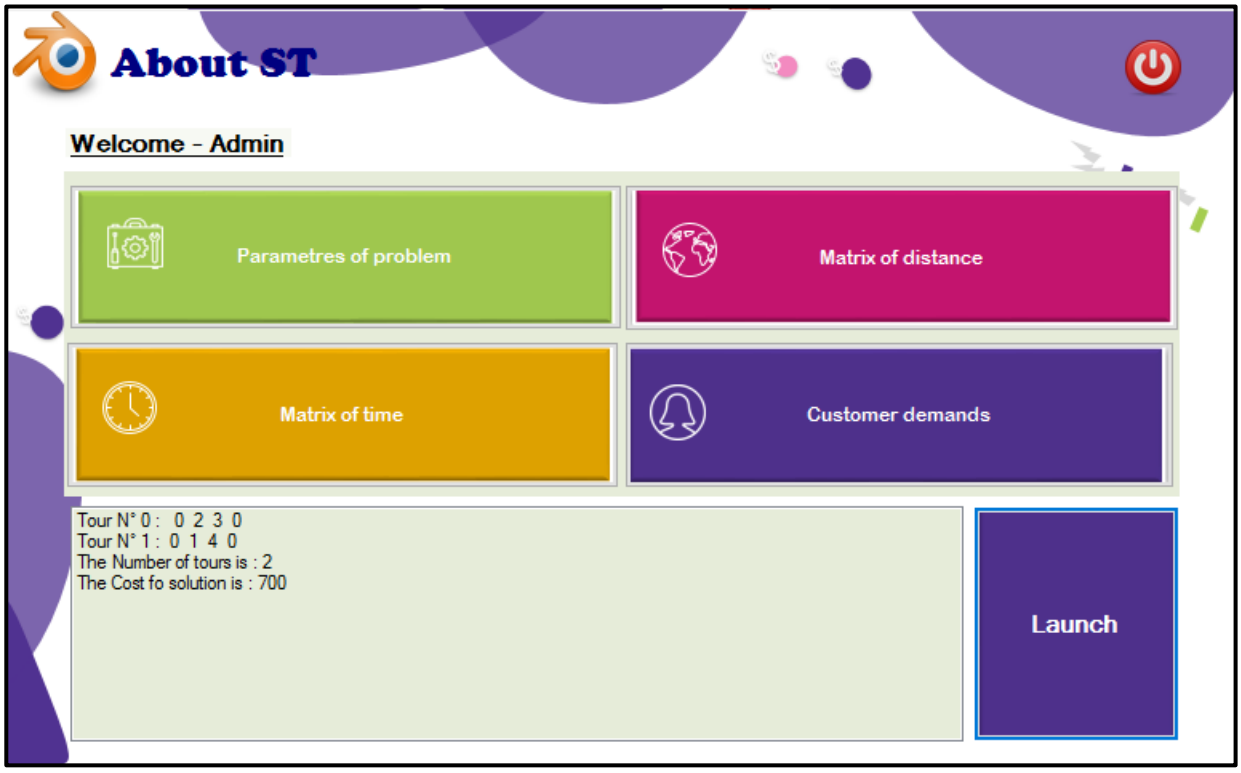


Figure 4.28: results for appointment1

Example for A2:

The dosage	the total weight delivered	injection time	service time	latest arrival times
5	10	80	10min	5min
6	12	120	10min	5min
4	8	160	10min	5min
7	14	260	10min	5min
5	10	300	10min	5min

Table 4.3: a test values for the parameter of customers

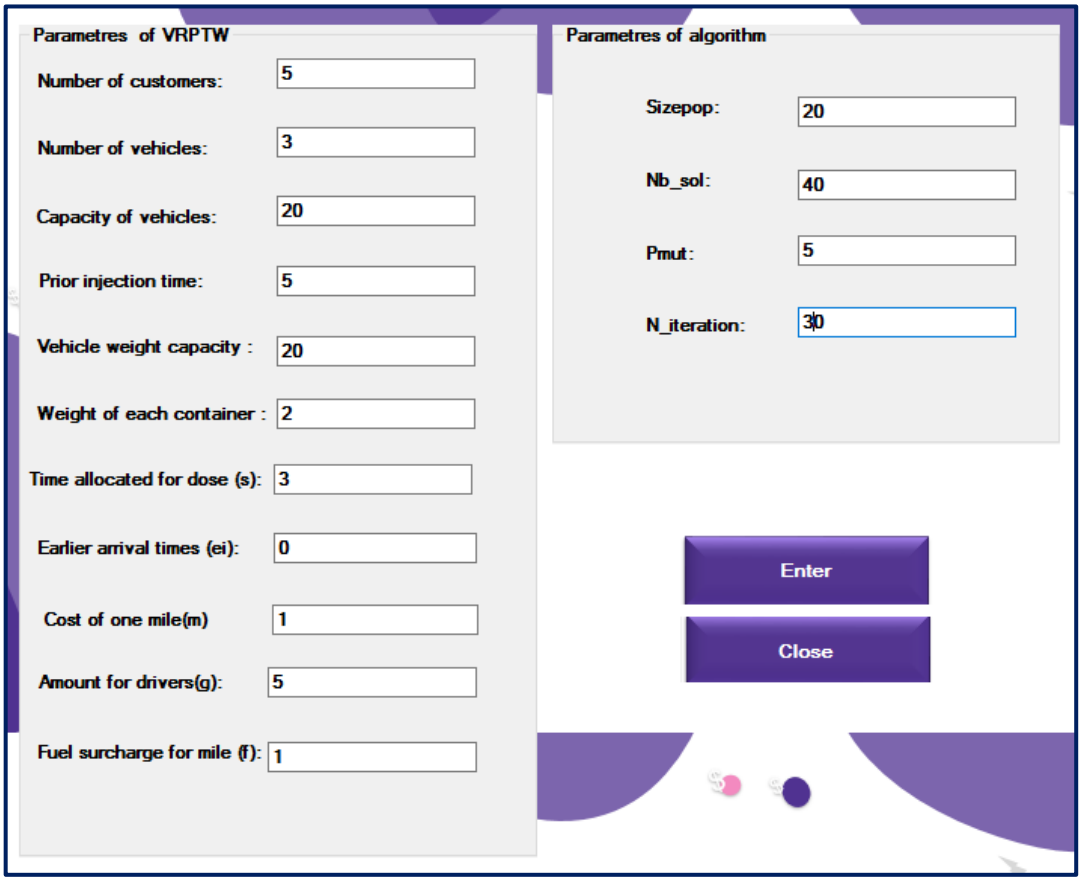


Figure 4.29: parameters of problem for Appointment2

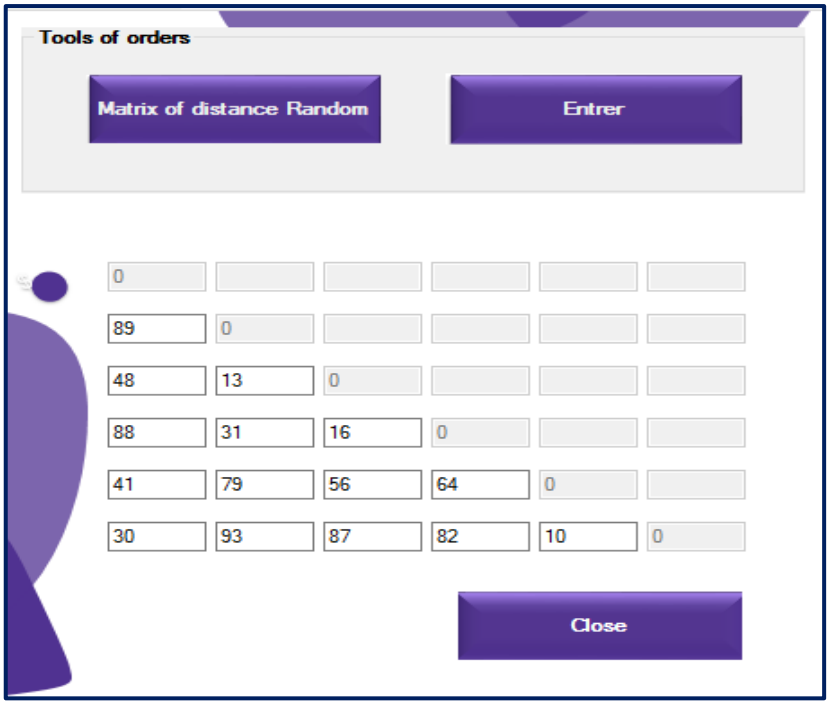


Figure 4.30: random distance matrix for appointment2

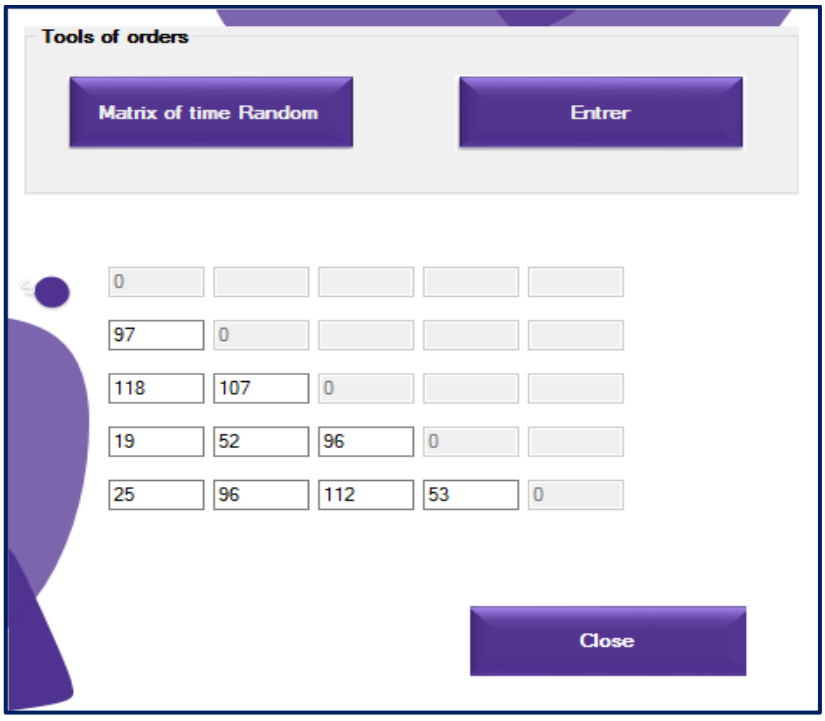


Figure 4.31: random time matrix for appointment2

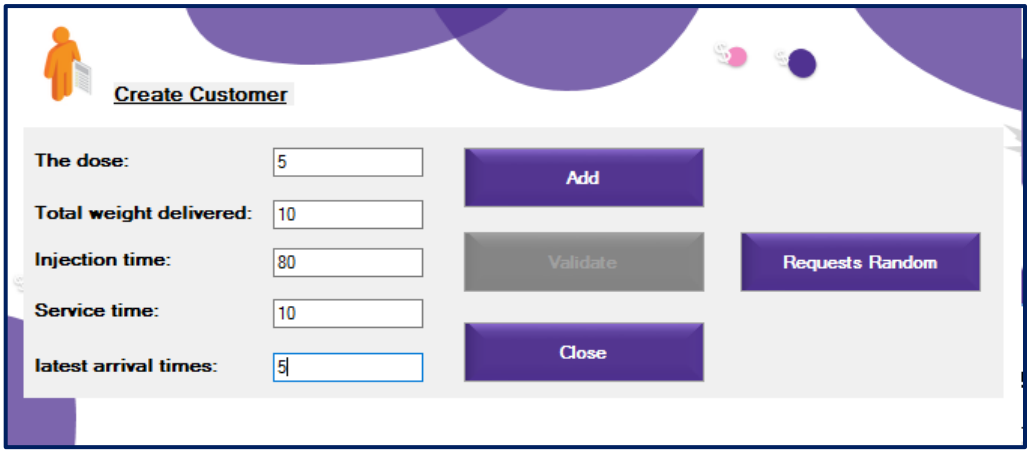


Figure 4.32: customer's parameters for appointment2

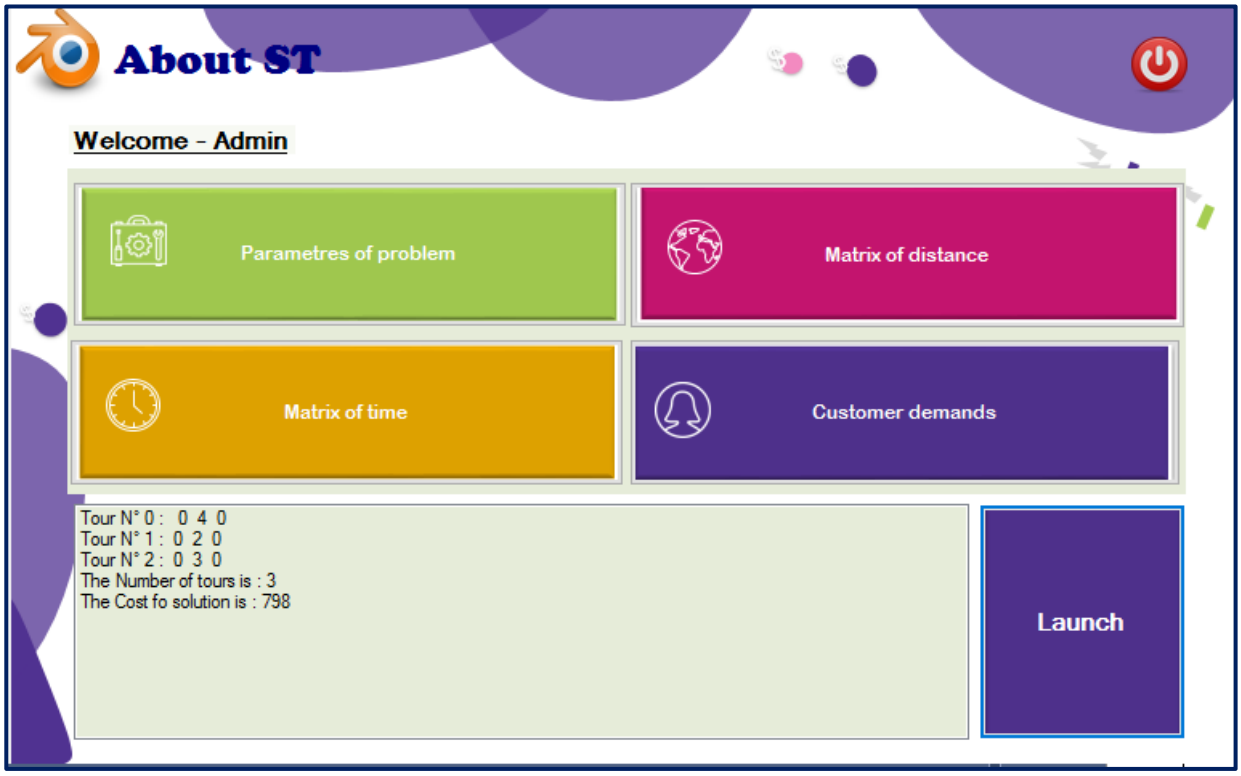


Figure 4.33: results for appointment2

Example for A3:

The customers' requests randomly.

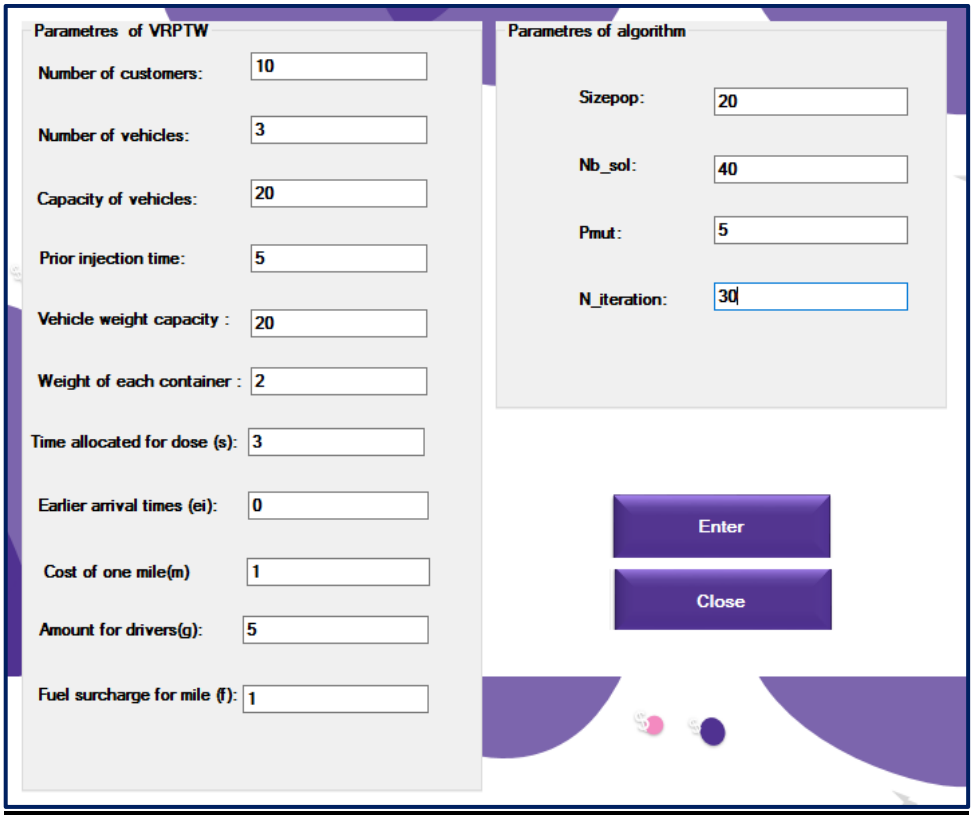


Figure 4.34: parameters of problem for Appointment3

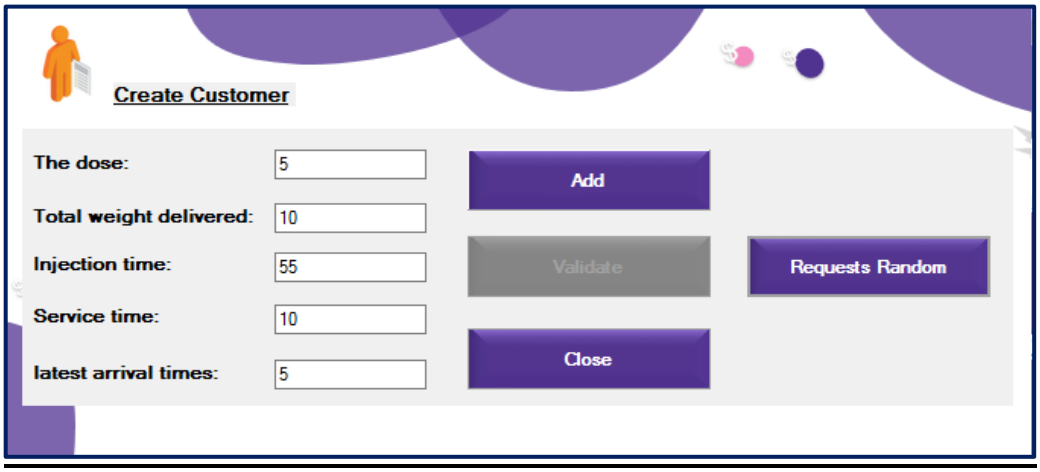


Figure 4.35: customer's parameters randomly for appointment3

Figure 4.36: results for appointment3

Example for A4:
The customers' requests are randomly.

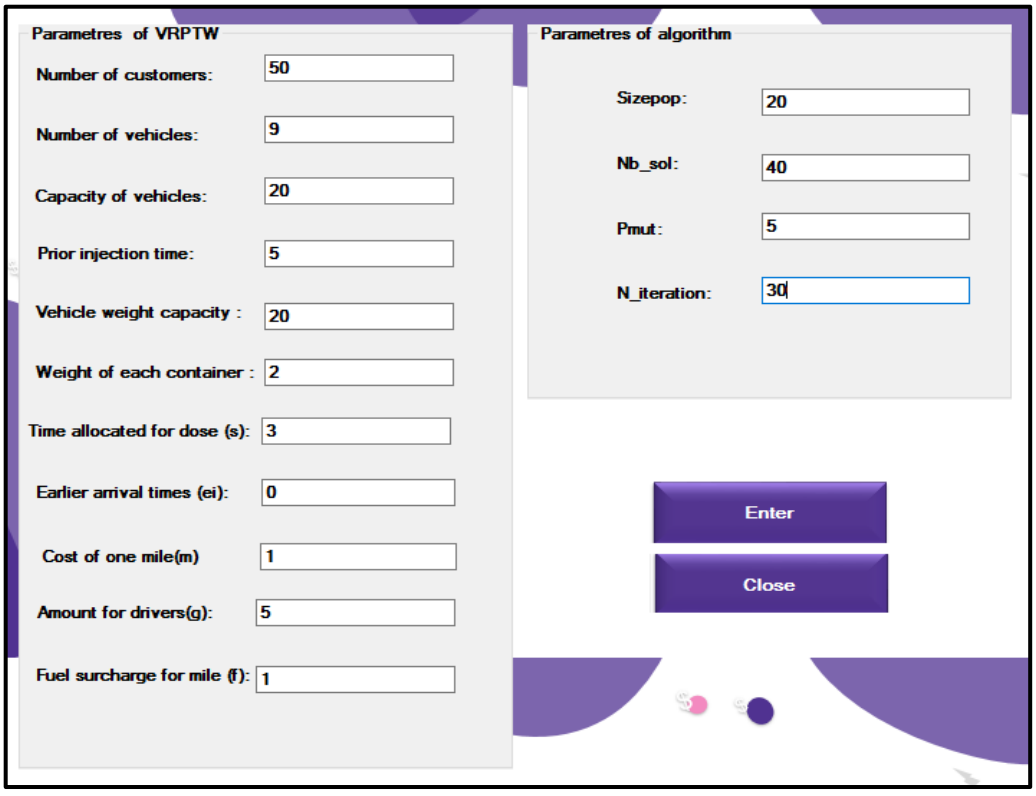


Figure 4.37: parameters of problem for Appointment4

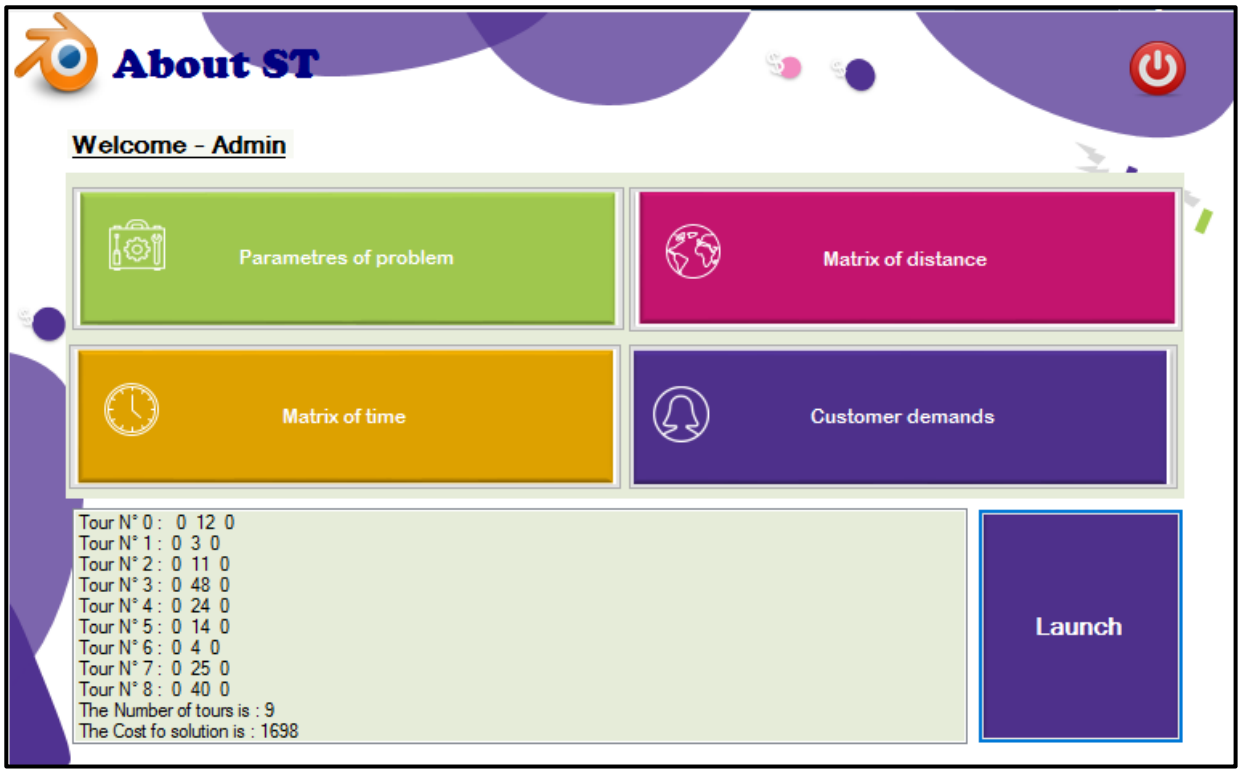


Figure 4.38: results for appointment4

6.2 Final results:

This table represents the results of the number of tours and the total cost for each example (appointments):

Appointment	The number of tours	The cost
A1	2	700
A2	3	798
A3	3	566
A4	9	1698

Table 4.4: the test values results.

7 Conclusion:

In this chapter we proceeded to the implementation of Meta-Heuristics GA to the problem of the VRPTW. We tested on 4 examples. As an experience shows, it is difficult to choose the best suitable parameters for this algorithm.

General Conclusion

A daily activity in today's pharmaceutical, medical, chemical and food industries involves the long-distance transportation of perishable products. The quality of such products may decay at rapid rates immediately after their production and during transportation. If a product does not meet certain quality criteria, the customer may discard it and refuse payment, in which case the manufacturing company may incur substantial loss of profit as well as customer dissatisfaction. In order to make sure their products arrive on time, in good quality and at minimum cost, companies have to carefully choose their transportation routes.

In the context in our work, we were interested in an important variant of these problems which is the vehicle routing problem with time windows (vrptw) with a limited capacity. All requests from customers are known in advance. This problem is to determine the routes for vehicles that transport from an origin to a destination by minimizing the total distance of the tours to minimize the total cost.

Before get into the problem, we presented the various problems of developing vehicle tours. We have detailed the VRP (Vehicle Routing Problem) which is the original problem of the different variants also We have presented the mathematical formulation of one of its variants of vrp problem which is the vrptw.

Then, after recalling the principle and characteristics of the genetic algorithm, we implemented an existing approach to solving the vrptw problem. This approach is based primarily on the genetic algorithm with a new technique that consists of crossover and mutation on the whole individual. This technique makes it possible to avoid the problems of violation of the constraints during the application of the two steps of crossing and mutation, and thus to accelerate the process of evolution.

Finally, we exposed the results of experimentation carried out on a population generated randomly.

Bibliography

[1] Aslaug S, (April 2004), Solving the Vehicle Routing Problem with Genetic Algorithms, Informatics and Mathematical Modelling, IMM Technical University of Denmark, DTU.

[2] Cheng, L. (2005). A GENETIC ALGORITHM FOR THE VEHICLE ROUTING. North Carolina Wilmington: University of North Carolina Wilmington.

[3] Christine Solnon, Résolution de problèmes combinatoires et optimisation par colonies de fourmis.

[4] Enrique Alba, B-D, (2004), Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms, Department of Computer Science, University of M´alaga, Central Computer Services, University of M´alaga.

[5] Ester Stegers, (December 11, 2009) A Solution Method for Vehicle Routing Problems with Time-Dependent Travel Times,Delft University of Technology.

[6] Francis I, P -T, SAMPSON, C.B, (1998) , Textbook of Radiopharmacy, 3rd Edition, ISBN 905699154X.

[7] Henri Luchian, Mihaela E and A- B, (2015) On Meta-heuristics in Optimization and Data Analysis. Application to Geosciences.

[8] Ioannis Akrotirianakis, A. C , (2015), An optimization-based approach for delivering radio-pharmaceuticals to medical imaging centers.

[9] Ilhem Boussaid, (2013) Perfectionnement de métaheuristiques pour l'optimisation continue, UNIVERSITÉ DES SCIENCES ET DE LA TÉCHNOLOGIE HOUARI BOUMEDIENE.

[10] Mei-Shiang C-A, (October 2003) REAL-TIME VEHICLE ROUTING PROBLEM WITH TIME WINDOWS AND SIMULTANEOUS DELIVERY/PICKUP DEMANDS Professor Department of Business Administration Chung Hua University.

[11] Nicolas Durand ,(2016), Algorithmes Génétiques et autres méthodes d'optimisation appliqués à la gestion de trafic aérien .

[12] Potvin, J.-Y. (16 juin 2009). Evolutionary Algorithms for Vehicle Routing. Centre-Ville, Montreal, Canada: Universit´e de Montreal.

[13] organizations, I. M.-M. (2017). Radiopharmaceuticals: Production and Availability,Vienna.

[14] Souquet Amédée, R-F, (21/06/2004) ALGORITHMES GENETIQUES, TE de fin d'année Tutorat de Mr Philippe Audebaud.

[15] Thangiah, S. R. (30 nov 2017). Vehicle Routing with Time Windows using Genetic. In L. Chambers, *Applications Handbook of Genetic Algorithms:New Frontiers*. USA: Artificial Intelligence and Robotics Laboratory Computer Science Department.

[16] TiesWestendorp, J.-T. B. (June 2017). Production and Delivery of Radiopharmaceuticals to Medical Imaging Centers. Enschede,Netherlands: univ-of-twente.

[17] (2006) Advances in medical radiation imaging for cancer diagnosis and treatment, Nuclear Technology Review 2006, IAEA.

[18] (2005) Beneficial Uses and Production of Radioisotopes. 2004 Update. NEA/IAEA Joint Publication. OECD.

[19] (2006) Directory of Cyclotrons Used for Radionuclide Production in member States. IAEA-DCRP/CD.

[20] Radioisotope Handling Facilities and Automation of Radioisotope Production. IAEA-TECDOC-1430 .

[21] The European Contrast Media and Radiopharmaceuticals Markets. Market Research.com. http://www.marketresearch.com/map/prod/1058730.html consulted : 2018/02/18.

[22] BIO-TECH Systems, Inc. Market Research in Healthcare field with expertise in medical imaging and Radioisotopes. http://www.biotechsystems.com/default.asp consulted : 2018/02/18.

[23] SAMPSON, C.B., Textbook of Radiopharmacy, 3rd Edition, 1998, Publishers-Taylor & Francis Inc, ISBN 905699154X.

[24] Batouche Mohamed Chawki, Optimisation Multiobjectif Par Un Nouveau Schéma De Coopération Méta/Exacte, université Mentouri de Constantine.

[25] https://www.researchgate.net/figure/Classification-of-optimization-techniques-adapted-from-4_fig1_317901944 consulted: 2018/04/05.

[26] https://www.researchgate.net/figure/Genetic-algorithm-operation-chart_fig1_236221295 consulted: 2018/04/20.

[27] https://www.researchgate.net/figure/Basic-steps-in-a-genetic-algorithm_fig1_237147466 consulted: 2018/04/13.

[28] https://www.researchgate.net/figure/Genetic-Algorithm-Tree-Basic-steps-of-GA-selection-crossover-and-mutation_fig2_260377604 consulted: 2018/04/13.

[29]https://www.researchgate.net/post/Can_anyone_send_me_Algorithm_for_crossover_and_mutation_ie_pseudo-code_of_crossover_and_mutation_consulted :2018/04/13.

[30] https://www.researchgate.net/figure/Mutation-operators-applied-to-chromosomes-in-the-proposed-genetic-algorithm_fig8_272093243 consulted : 2018/04/13.

[31] https://www.sciencedirect.com/science/article/abs/pii/S036083521400360X consulted: 2018/04/13.

[32] https://www.sciencedirect.com/science/article/pii/S1007021409700586 consulted: 2018/04/13.

[33] https://blogs.msdn.microsoft.com/dotnet/2017/04/05/announcing-the-net-framework-4-7/ consulted: 2018/05/02.

ملخص

هذه المذكرة تناولت مشكلة توجيه السيارة مع نافذة زمنية ، وهي متاحة عند الطلب من الشركات التي تقوم بتوصيل الأشعة الإشعاعية إلى مراكز التصوير

مشكل تحديد مسارات السيارات والأوقات للمركبات التي تنقل طلبات المستخدمين من أصل واحد إلى الوجهة وبالتالي VRPTW، التي تنتمي إلى عائلة المشكلات (NP-Hard) مشكلة توجيه المركبة مع النافذة الزمنية لحل هذا النوع من المشاكل (metaheuristics) الحاجة لاستخدام أساليب تقريبية

لحل هذا المشكل استعنا بخوارزمية يعتمد على مبدا الوراثة والذي قمنا بتجريبها على معطيات تم تشكيلها عشوائيا.

Abstract

This thesis focuses on vehicle routing problem with time window available upon request of companies that delivery the radiopharmaceuticals to the imaging centers.

Vrp is an optimization problem of determining the tours and times for vehicles that transfer the user's requests, from one origin to the destination. The vehicle routing problem with time window (VRPTW) that belongs to the family of NP-Hard problems, hence the need to use an approximate method (metaheuristics) to solve this kind of problems.

In the end to resolve this problem we used a (genetic algorithm), which is tested on randomly generated data.

Résumé

Cette thèse porte sur le problème d'acheminement des véhicules avec fenêtre temporelle disponible sur demande des entreprises qui livrent les radiopharmacetiques aux centres d'imagerie.

Vrp est un problème d'optimisation de la détermination des tours et des temps pour les véhicules qui transfèrent les demandes des utilisateurs, d'une origine à la destination. Le problème de routage du véhicule avec fenêtre temporelle (VRPTW) qui appartient à la famille des problèmes NP-Hard, d'où le besoin d'utiliser des méthodes approximatives (métaheuristiques) pour résoudre ce genre de problèmes.

Pour résoudre ce problème, nous avons utilisé un algorithme génétique, qui est testé sur des données générées aléatoirement.

Keywords: radiopharmaceuticals, medical imaging centers, vehicle routing problem, time windows, logistics, genetic algorithms, operations research.