Capstone Project

The Vehicle Routing Problem

By: Hajar Nibou

Supervised by: Dr. Ilham Kissani

May 1st, 2017
THE VEHICLE ROUTING PROBLEM

Capstone Interim Report: Hajar Nibou

Approved by the Supervisor

Dr. Ilham Kissani
ACKNOWLEDGEMENTS

I would like to express my thankfulness to my academic supervisor Dr. Ilham Kissani, whom I have already worked with during my internship, for her close assistance, guidance, encouraging lectures and support from the initial to the final level during my capstone project.

Moreover, I am so grateful for all the people who have helped me in my capstone project, especially Khalid Thillai Rajah, a CS student, who helped me code with Matlab, and Oussama Ayach a master student in Web, Multimedia and networks at South Bretagne University.

Last but not least, I would like to thank my family and friends, because without them I could have never done this project. This project will help my university whereby I have grew and developed my intellectual curiosity as well as established a firm foundation in science and engineering.
CONTENTS

ACKNOWLEDGMENT ........................................................................................................... 3

CONTENTS .......................................................................................................................... 4

ABSTRACT ........................................................................................................................... 5

Literature Review ................................................................................................................. 6 - 10

STEEPLE Analysis ................................................................................................................ 10 - 12

Methodology ....................................................................................................................... 13 - 15

Algorithm ............................................................................................................................. 15 - 17

Dijkstra’s algorithm ........................................................................................................... 17 - 19

Solving the salesman manually ......................................................................................... 19 - 23

Salesman using Matlab ...................................................................................................... 23 - 32

Salesman using javascript ................................................................................................. 32 - 41

Conclusion ......................................................................................................................... 41 - 42

References ......................................................................................................................... 43-45
ABSTRACT

This capstone project will attempt to find the optimal set of paths for Vermousil Tanger’s company vehicles, from the enterprise to the other stations respecting the constraints that are mainly the distance, capacity, and time.

The vehicle routing problem project involves three main parts. The first part of the project will be talking about the literature review of the issue, its explanation and main causes. While the second part, will be about the data generation that is mainly searching for the constraints of the VRP in Vermousil Tanger’s company, that are the number of stations visited by the vehicles, the time window constraint, the capacity, and the distance. Lastly, in the third part, I will be doing the implementation of the algorithms using the software Matlab for the distance-constrained vehicle routing problem, while my other teammates will be working on the capacity and time constrained vehicle routing problems.

This capstone project is of upmost vitality in that it will help Vermousil Tanger’s company to reduce its expenses in terms of additional mileage that is not needed. This project will aim to determine optimally the shortest path between each station and henceforth establish the optimal solution to this NP problem. By illustrating the stopping stations as nodes with connecting edges, a shortest path will be determined in order to traverse in an efficient and effective manner.
I. Literature Review:

Mirroring the NP-hard problem of the travelling sales problem (TSP) through the vehicle routing problem (VRP), it is of utmost essentiality to juxtapose both problems and realize that the generalization of the TSP through asking the rhetorical question of “What would be the optimal combination of pathways that would be traversed in such a way that would enable us to deliver to a set of customers without repetition and visiting nodes exactly once?” This famous operations research problem stems from the TSP through raising somewhat a similar question of how many nodes can we traverse from a node to another exactly once keeping in mind to return to the first node? (Vehicle Routing Problem, 2017)

To briefly touch upon its history, the VRP was first applied through creating an algorithm concerning petrol deliveries. Its very first appearance in 1959 was by George Dantzig along with John Ramser. Several research have been done in order to illustrate the use of the famous Vehicle Routing Problem; nevertheless, to apply it to a real life situation where various factors come into play is the upmost challenge keeping in mind that these constraints will vary with regards to the objective of such optimizations as well as human factors that would be taking place. (Vehicle routing, 2017)

When exploring the VRP, and through taking into consideration the fact that besides it being a theoretical computer science problem, the logistics in terms of distributing and delivering in an efficient and effective manner, meaning optimally using the least amount of resources and
gaining more profits in the least amount of time, is where the VRP come into play. At Vermousil Tanger’s company, there exist transports along with other conveyance means that would drop off staff members to their home from the company, and again from the company to their home. Such transportation means will require the most accurate, optimal traversal route in order to decrease the expenses in terms of transportation, though it will not be visible in the short term, but such application will help to make such impact in the long term.

Several researches have been conducted with regards to the Vehicle Routing Problem, amongst which the reinforcement of the application of the VRP was deemed essential. There exist several applications to the VRP, but the two highlighted ones are the following: The research conducted on VRP theory applied to logistics transportation, and the research on the VRP theory applied to the optimization based on improved two phase algorithm in the field of electronic commerce.

First and foremost, the logistic transportation problem is one of the most important issue that faces the current age due to the apparent advent of radical increase in technology, especially that is applied to and integrated with the vehicles market. Due to the diversification in terms of transportation means, the increase in complexity in terms of routes or transportation network, when it comes to combinatorial optimization, the logistics transportation problem is considered to be an NP problem. Two researchers in the field of computer science and management engineering, Ruiqi Zhu and Yunkai Zhai, conduct this research entitled “The Application of VRP Theory in Logistics Transportation”. (Zhu et al, 2017) This research explores the VRP in relation to the the physical distribution. It was challenging indeed to conduct such research mainly taking into consideration that it is an NP problem, whereby the
set of decision problems can be potentially solved what is known as theoretical non-deterministic Turing machine. (Zhu et al, 2017)

On the other hand, the second known VRP applied research is entitled “Optimization Based on Improved Two Phase Algorithm under e-Commerce,” conducted by Wang Xiao-bou, Li Yi-jun, and Sun Jin-ying. (Wang Xiao-bou et al, 2017) One of the main constraints that the logistic distribution of e-Commerce is facing now is the radical increase in terms of, firstly, its development, and secondly the relationship between the role that the VRP theory plays on the implementation and the development of a given logistic distribution system. (Wang Xiao-bou et al, 2017) In the logistic distribution system, there are several components related to the VRP. Besides customers’ positions and the large order invoices and forms, there are little collections and various redundancies in terms of routes, meaning there exist numerous repeated routes. (Wang Xiao-bou et al, 2017) When applying any other methods that could be considered classical or traditional in order to solve the issue of repeated routes, several side constraints will come into play. Forth most, it is rather obligatory to rely on the NP problem in order to eliminate repeated routs. The three researchers applied the improved two-phased algorithm in order to solve the issue of VRP under e-Commerce. (Wang Xiao-bou et al, 2017) A clustering method in terms of customer encapsulation was partitioned in the first phase. The second phase on the other hand consists of receiving the customer’s point solution in each group through the use of the “improved genetic algorithm.” (Wang Xiao-bou et al, 2017) Henceforth, when they analyzed it, they came to the conclusion that they were dealing with various single TSP models. It was challenging since they had to deal with several groups of clusters, but at the end, they were able to the validity of this algorithm.
Given these two researches that have been conducted in order to show some of the possible applications of VRP in diverse fields, it was not an easy task to be accomplished. This is due to the various factors that needs to be taken into account with any specific problem that should be solved using VRP. For instance, there is a well-known blood collection problem that requires VRP, and the number of factors that will need to be taken into account range from the scheduling of the operation rooms to the technical equipment along with the schedules of the nurses and the doctors. (Anonymous, 2017) These problems are considered to be complex since they are usually considered as a set or a group of combinatorial of NP problems. Therefore, developing any effective and efficient solutions are considered to be of challenging tasks. (Anonymous, 2017)

Last but not least, in order to establish the theoretical framework of this problem, a separate, challenging case study needs to be mentioned. There is a conducted research paper written by Joaquim A. S. Gromicho, Sameh Haneyah, and Leendert Kok under the title “Solving a Real-Life VRP with Inter-Route and Intra-Route Challenges.” (Gromicho, J et al., 2017) Their paper face upmost challenges that were categorized into three ones, which are: inter-route restrictions, unmatched pickups and deliveries, and priority orders. (Gromicho, J et al., 2017) This problem is apparent at large firms for their distribution activities. Inter-route privileges and restrictions take place due to the requirement of high priced tools and equipment that such international distribution requires. The equipment will also be moved from one depot to another, and this causes the second challenge of unlatching equipment in a given depot. (Gromicho, J et al., 2017) Their research generates plans and follows intensified searching procedures in order to implement the VRP algorithms and the entire research did not fully eliminate the problems raised by VRP, but only decreased the distance travelled by 5%. (Gromicho, J et al., 2017)
Given the above existing case studies along with conducted researches, it is safe to stress on the challenges that the VRP brings about when taking into account real life factors. In this paper, the constraints that will be taken into account are time and distance, and given these constraints, will tend to find the optimal routes to traverse the given set of depots.

The main objective of this capstone project is to identify, optimally, the set of routes that would minimize the expenses along with directly impacted variables that are time and distance, when traversing the scattered stopping points and stations from the starting point which is the university itself. For my part of the project I will be focusing on the distance constraint, while other groups will be focusing on the time window constraint and the capacity.

The geographic location of the stations need to be analyzed as well as mapped at the level of nodes and edges in order to accomplish, using the VRP algorithm, the purpose of this capstone. There exist several algorithms such as the heuristic method, metaheuristic, and exact method, and in this capstone, I will be implementing my code using the exact method in MATLAB using the integer linear programming respecting the precisely the parameters that concerns the traversals that will take place between the set of nodes.

The capstone project will be partitioned into three main sections. The first section will concentrate on the definition of the variables through conducting the STEEPLE study in order to determine the right constraints that were pointed out earlier in the paper. The second section will mainly focus on the algorithms that could be potentially used in order to solve this particular problem. Finally, the third part will include the methodology that will be
implemented using MATLAB in order to optimally find the shortest traversal path that could be used in order to solve the issue in terms of effectiveness and efficient use of its transportation means.

II. **Definition of the heuristic, metaheuristic, and exact algorithm.**

Heuristic function, otherwise called heuristic algorithm or simply heuristic, is an algorithm that is mainly used in computer science domain, artificial intelligence, or arithmetical optimization. When we are generally facing an NP-hard problem that is hard to solve, and that does not have any exact method, even if a method is found, its complexity in terms of big O will be significantly high; henceforth, we generally opt for the heuristic method. In fact, besides the fact that the exact methods are so slow to be executed, they are considered to be very expensive. The heuristic method is a shortcut, since it is relatively fast and easy for finding an estimated solution that is close to the exact one derived from the classic methods. Heuristic methods unfortunately do not always assure an exact solution to the asked problem. In fact, even when algorithms sometimes might find the exact solution for a given problem; nonetheless, they are still entitled heuristic until the solution found is proven to be the best alternative. One of the main reasons behind the inaccuracy of this algorithm is that it discards some of the problem’s demands in order to make it fast to execute. (Bianchi et al., 239)

Heuristics algorithms might not always be producing an accurate result, but they are mostly used in combination with other optimization algorithms in order to improve the effectiveness and efficiency of the solution generated. Those types of functions are the mostly used algorithms in NP-hard problems that are executed in real-word applications, and in theoretical
computer science because they are the only practical option for diverse complex optimization problems. (Bianchi et al., 239)

The metaheuristic algorithms on the other hand are approaches used to monitor the search process. Just like the heuristic functions, or any iterative method, not problem-specific, that is close to the exact solution, the metaheuristic algorithm is usually non-deterministic. Metaheuristic algorithms are known to be used in solving NP-hard problems in order to give an estimation of a solution, when the exact solution is hard to find, slow, or hugely expensive to be executed. Those NP-hard problems again are used in computer science and mathematical optimization. The problems dealt with using the metaheuristic algorithm have often unfinished defective information that are limited in computation ability. The solution generated from this optimization method is reliant on the random variables generated.

The difference between the metaheuristic and the heuristic method is that the heuristic functions paraphrase the problem with paying attention to the particularities of the issue. Nevertheless, as they are often too picky, they mostly crash while finding their optimal solution and thus fail most of the time to get the desired optimal solution. On the other hand, metaheuristics are techniques that do not depend on the problematic of the problem. Hence, they miss generally the details of the problem, and do not mind some deterioration of the solution. They are used as black boxes as you cannot see its implementation. This technique allows them to explore more thoroughly the solution, and hence reach the optimal solution.

For our case, we will be using the exact method of the vehicle routing problem. Even though that this method takes a longer time, it is the very exact one. The other group on the other
hand who is dealing with the capacity constraint will be working with the heuristic algorithm. Because when they have tried to implement their code using the exact method it crashed.

III. STEEPLE Analysis:

There exist two types of analyses, the SWOT and the STEEPLE. The STEEPLE analysis is the best alternative in that it exhibits more external fields and is considered to be the more advanced analysis in the sense that it provides a broader overview of numerous external fields. It is an acronym of the following fields: Social, Technological, Economic, Environmental, Political, Legal, and finally Ethical analyses. (Conner, 2017)

Through this external environment analyzer, we will be able to determine the variables that will be branching out the constraints that will be taken into account throughout this capstone project.

First and foremost, with regards to the social or the socio-cultural aspect, there is no major impact on the cultural aspect or society; nevertheless, the proposed solution will inevitably reduce the time of the driver as well as the staff members when they are being dropped off. This is mostly relevant in the case of an emergency. For instance, if the driver or one of the passengers working in the company encounters a personal problem and he needs to get to a certain location in order to deal with an urgent matter, then this strategy will enable them to reach their destination as soon as possible.
On the other hand, when analyzing the technological aspect of the matter, the technical infrastructure, vehicles used, and the lifetime mileage of such buses used for transportation will determine the qualitative logistics distribution. Meaning that advanced technologies will indeed lead to better and faster traversal between stations in Tangier. Also, the optimal use of the brakes will last longer. This will affect positively the cost meaning by that the expense, the distance, and the time that will be reduced.

Economically speaking, rate of inflation along with interest rates will impact the expenses to a certain extent in terms of fluctuations in the price of oil and the fuel used in the vehicles. Regardless of the variation in the expense of the fuel, the company is going to save money and consequently time.

Environmentally speaking, it will not affect the transportation or the speed of the traversal between nodes since there are no specific regulations against the amount of carbon emission or any other regulations that would directly impact negatively or positively the speed or any factors of traversing. In fact, diminishing the distance will decrease the amount of the fuel burnt; hence it will cause less harm to the environment. The result of decreasing the amount of combustion of fuel will decrease the presence of carbon dioxide in the air that is one of the main causes of global warming.

Political aspects would be in terms of the transportation capacity; however, in our case, regulates its own transportation restrictions and terms and would not be affected by external political environment. In fact, optimizing the trajectory will not have a direct impact on the political aspect. Unless if in a certain city or country, there are either strikes or other political
issues that would close the optimal road. In this case, the vehicle will have to change the road and it will certainly be longer.

Moreover, the legal aspect would affect the speed by which the vehicles should operate due to the speed limit that is imposed as well as weight restrictions on the bus. Since the higher the speed the higher the amount of fuel burnt, and the costly it is going to be. Finally, ethically speaking, the carbon emission is the main ethical responsibility that would question the ethical part of this project. Also, there is a possibility of ethical misconduct for the drivers. They might incorrectly declare that they stopped in a station, when they not.

Last but not least, concerning the distance constraint, the STEEPLE analysis revealed that the technological and legal regulations are the only potential aspects that might impact this specific variable to a certain extent; nevertheless, distance is a measurement that is fixed between the nodes, and in this case between the stations or the stopping points. This means that it is the set of traversals that will be made between the distances is the constraint in the sense that we will need to find the shortest path without revisiting the nodes twice and hitting all the needed stations. This means that the total distance is the one to be minimized through minimizing each traverse of the subparts of the total visits.

IV. Methodology:

Algorithm:

When subjected to networking problems, the model should be illustrated in terms of nodes and edges. There exist several networking problems such as transhipment, assignment, and maximal flow, and they all use the same approach in order to be solved in terms of applying VRP. In our case, which is transportation, the design of the route that will be used in order to
traverse all the stops once and exactly once in the shortest distance possible will be solved using VRP. The following is the methodology that will be implemented in terms of solving the distance constraint:

- The number of buses used to transport students between a school and the pick up stop, that is in our case the company: k
- Each bus is dedicated to a single path and it uses the same path for both picking up or dropping off the staff
- Each bus has an upper bound on the total amount and a lower bound of the distance it may traverse
- Each pick up point is visited only once and should not be visited by another bus
- What is the minimum number of buses required to transport all of the students and their corresponding routes so as to minimize the total cost of transportation?
  - We consider a complete graph \( G=(V,A) \) with \( V \) nodes and \( A \) once
  - Distance between each node pair is characterized by a symmetric distance matrix \( D=\{d_{ij}\} \) when \( d_{ij} \) is the distance required to traverse from node \( i \) to node \( j \) (also from \( j \) to \( i \))
  - Node set: \( V=\{0\}UI \) such as node 0 is depot (ex: school) and \( I \) set of intermediate node. Knowing that each node \( i \) included in \( I \) has a number of students (or anything) to be picked up from stops
- We want to determine \( k \) paths connected to the depot such that:
  - Total capacity doesn’t exceed the given limit: \( Q \) (This part is the other team’s constraint)
  - Total length doesn’t exceed the given limit: \( T \)
• We will introduce a “dummy” node \( d \) to which all the last nodes of each edge or path will be connected to.

• **Problem reduces to finding \( k \) node disjoint paths between 2 points on an expanded graph \( G'=(V',A') \) such that:**
  
  o All nodes are visited exactly once
  
  o \( V' = V \cup \{d\} \)

It is rather vital to pay attention to the Subtour constraint that is present when implementing the TSP or the VRP problem. Since in the vehicle routing problem there is the initial point that we will be calling the depot, in our case it is going to be the company. The transport should come back to the company’s parking. While running the code, it will generate other routes that we call subtours. Those certain routes will inevitably exist; hence a subtour constraint should be added, in order to have an optimal solution vacant from subtours. To clarify the concept, a subtour is in deed when there exists a round trip, meaning a route that starts and ends up at the initial depot; however, such round trip would not satisfy all the demands at the level of stopping at each given station. Since the initial requirements of a VRP problem is to visit each independent node exactly once, leave the station exactly once, and eliminating any existing subtours. Since the prefix of subtour is “sub,” it means that we have to solely deal with subsets of stations or depots.

When talking about subtour elimination, we need to take into account all the subset of the cities or depots that exist. The way to initialize such elimination is by verifying that there exists an arc that is leaving a depot that is part of the subset and entering another one that is not part of the subset. By following such algorithm, we come to perceive that there is an
exponential number of subtour elimination constraint. Only a small number of subtours should be present in order to eliminate them.

**Dijkstra’s algorithm:**

A Dutch computer scientist called Edsger Dijkstra in 1956 perceived the Dijkstra’s algorithm. This algorithm is a graph search algorithm that finds the shortest path from one node to another for the whole graph with positive edge path costs, in order to make a shortest path tree. Dijkstra's algorithm is usually used in routing and in our case it was part of our code in order to solve the salesman algorithm problem. (Zhan, F. Benjamin; Noon, Charles E, 1998)

Here is a manual way finding the shortest path using the Dijkstra’s algorithm. It can be used also by matlab, as it is in the salesman or any vehicle routing problem.

In the following table, if the nodes are not directly linked we use the symbol ∞.

<table>
<thead>
<tr>
<th>V</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0_A</td>
<td>4_A</td>
<td>3_A</td>
<td>∞</td>
<td>7_A</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>C</td>
<td>0_A</td>
<td>4_A</td>
<td>3_A</td>
<td>14_C</td>
<td>7_A</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>B</td>
<td>0_A</td>
<td>4_A</td>
<td>3_A</td>
<td>9_B</td>
<td>7_A</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>E</td>
<td>0_A</td>
<td>4_A</td>
<td>3_A</td>
<td>9_E</td>
<td>7_A</td>
<td>∞</td>
<td>12_E</td>
</tr>
</tbody>
</table>
To find the shortest path from one node to another we use the above table.

Taking for example the shortest path from node A to node B:

A → B (Distance = 4)

The shortest path from node A to node C:

A → C (Distance = 3)

The shortest path from node A to node D:

A → E → D (Distance = 9)

The shortest path from node A to node E:

A → E (Distance = 7)

The shortest path from node A to node F:

A → E → D → F (Distance = 11) % % In this case it is going be shorter for the driver to pass by E and D then F rather than directly to F.

The shortest path from node A to node G:

A → E → G (Distance = 12)

**Solving the salesman problem manually:**

Given the distance between the stations as below, our salesman should find the optimal route to take starting from station A, passing by all the stations exactly once and coming back to the first station that is A:
First step:

We are going to perform the raw minimization that is summarized in taking the minimum number of each row, and subtracting the corresponding raw numbers by this integer as you can see below:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_</td>
<td>7 - 4 = 3</td>
<td>6 - 4 = 2</td>
<td>8 - 4 = 4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>_</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>2</td>
<td>_</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>_</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>_</td>
</tr>
</tbody>
</table>

Second step:

In this step we’ll be performing the column minimization that is similar to the row minimization. The only difference is that instead of taking the minimum number of each raw, we will be taking it from each column and subtracting the corresponding column numbers by this number as the following:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>2 - 0 = 2</td>
<td>_</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0 - 0 = 0</td>
<td>2</td>
<td>_</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>
Third step:

In this step we compute the penalty of each zero in the grid. If we take the example of the first 0 corresponding to the row A and the column D, its zero penalty is computed as following:

We take the minimum number of its matching row and add it to the minimum number of the matching column. In this case $0 + 1 = 1$. After that, we reduce the matrix by deleting the row and column corresponding to the zero with the highest penalty.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

After deleting the row corresponding to station B, and the column corresponding to the station D, we get the following reduced matrix:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
After performing the preceding operations, we should consider mentioning the deleted row and column in the following format because it will help us with finding the optimal road. B→ D

Again we do the column minimization, followed by the zero penalty to get the following grid:

<table>
<thead>
<tr>
<th>V</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_</td>
<td>1</td>
<td>0⁰</td>
<td>0¹</td>
</tr>
<tr>
<td>C</td>
<td>0⁰</td>
<td>0⁰</td>
<td>_</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>_</td>
<td>0¹</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>0⁰</td>
<td>0⁰</td>
<td>1</td>
<td>_</td>
</tr>
</tbody>
</table>

After that, we keep repeating the same steps:

After finding the highest penalty that is 0¹, we delete both the column and the row corresponding to it, and write down A→ E. We after get the matrix below:

<table>
<thead>
<tr>
<th>V</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0¹</td>
<td>0⁰</td>
<td>_</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>_</td>
<td>0²</td>
</tr>
<tr>
<td>E</td>
<td>_</td>
<td>0⁰</td>
<td>1</td>
</tr>
</tbody>
</table>

After that we delete the column and the row corresponding to 0², and make sure to write the corresponding annotation: D→ C

<table>
<thead>
<tr>
<th>V</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0⁰</td>
<td>0⁰</td>
</tr>
<tr>
<td>E</td>
<td>_</td>
<td>0⁰</td>
</tr>
</tbody>
</table>
After writing the reduced matrix,

Similarly to the previous steps done before, we are going to reduce the matrix further by omitting C → A and E → B, since the road from E to A is impossible.

We then compute the distance using the data given:

\[ A \rightarrow E \rightarrow B \rightarrow D \rightarrow C \rightarrow A = 4 + 6 + 5 + 9 + 6 = 30 \text{ km} \]

**Salesman using Matlab:**

MATLAB was used in order to implement the VRP problem. The reason for choosing MATLAB in order to implement the solution for this problem is due to its richness in terms of toolboxes and functions that could be used in order to simplify such complex problem. It allows us to break down the problem into sub problems and linearly solve them. Indeed, such toolboxes exist in other programming languages; however, why not use MATLAB if it has all the necessary toolboxes to solve VRP.

- **Al Akhawayn salesman code**

In order to start coding for the vehicle routing problem, I have begun by coding the salesman problem of Al Akhawayn University issue. I have gathered the information of the stations from the transportation department and gathered them in an excel sheet. For my part, I had only to know the visited stations that are Al Akhawayn University, Down Town Residence, the market, Pam/Slaoui, ifrane school residence, Annex Residence Ifrane School, Atlas 1 Temaddekin. The other groups were mainly concerned by the capacity, and time window
As showed in the excel sheet, the vans here in Al Akhawayn university do not follow the salesman rules, since the stations are visited more than once.

Before starting my code I have searched at the coordinates of each station using google maps (longitude and latitude).

The coordinates of each station are the following:

Al Akhawayn University, being the depot: (33.5393881, -5.1079893)

Down Town Residence: (-5.108355, 33.530303)

The market: (-5.115607, 33.526001)

Pam/Slaoui: (-5.1177443, 33.5258333)

Hay Atlas 1: (-5.1212247, 33.536493)

Ifrane school residence: (-5.1355597, 33.527736)

Annex Residence Ifrane School: (-5.1355047, 33.5246802)

Hay Atlas 2: (-5.1209667, 33.540424)
In order to make sure that the coordinates I have picked from google maps were the right ones, I have drawn them using Matlab using the following code:

```matlab
%%
Y=[33.530303, 33.526001, 33.536493, 33.540424, 33.5246802, 33.527736, 33.539388 ];
X = [-5.108355, -5.115607, -5.1177443, -5.1212247, -5.1209667, -5.1355047, -5.1355597, -5.1079893 ]
plot(X,Y,'+g')
plot_google_map
%%
```

This code draws the coordinates of the stations visited by Al Akhawayn University in green. As you can see in the image below, the coordinates were drawn in green in Ifrane’s city exactly in the mentioned stations. Hence, our longitude and latitude coordinates are exact.

![Map of Al Akhawayn University and its surroundings](image)

After that I have found the optimal trajectory for Al Akhawayn University van using the code below:

In fact, this following code consists of several blocks and each need to be explained in order to give an overview and a broad understanding of the implementation. The following is an explanation of the functions, methods, and documentations used in order to implement the salesman problem of Al Akhawayn university van.
• Setting the depot to be [33.5393881, -5.1079893] that are the coordinates of the university. After that allocating for each \( x(i) \) the coordinates of Ifrane’s station using the following loop: For a given index that has a value assigned, which in this case is 1 for \( i \), the block statement between for and end is being executed \( n_{stop} \) times, which was initialized as an integer at the beginning of the code and given a value of 8. Meaning the statements will be executed 8 times and then exiting the loop the end of the fourth round.

```matlab
for i=1:n_{stop}
    X_coordinates(i)=X(i);
    Y_coordinates(i)=Y(i);
end
n=1;
```

• The below while loop differs from the for loop in the sense that in the while loop, the inner block of statements will be repeated until the loops holds false. Meaning that the block will be executed as long as it holds true, which in this case \( n \) being less than or equal to \( n_{stop} \), and \( n \) was initialized as one. We can see that it is being incremented by 1 at the end of the statement before starting the next round. When \( n \) becomes eight, the statement will hold false and hence automatically exiting the loop since it is not true anymore. And when leaving the loop it will be writing for example in the case of \( i = 1 \): the 1 point coordinates are \(-5.923230e+00,3.571376e+01\).

```matlab
while (n<=n_{stop})
    fprintf('The %d point coordinates are %d,%d \n',n,X_coordinates(n),Y_coordinates(n));
    n = n + 1;
end
```
• Then as explained earlier the plot function is going to draw for us the coordinates of the eight stations. In this case, we are using hold on that is a function that exists in the library that is used in order to recall the plot line and then add another plot to the existing graph. In other words, it adds a stem to a certain given domain. At the end, hold off is used in order to make the replacement and append to the old plots.

    hold on
    plot(X_coordinates,Y_coordinates,'+g') hold off

• \texttt{Nchoosek} is a function that is included in the library which returns a binomial coefficient. In this case it will be returning it to the variable combination. The binomial coefficient that is being returned is in the form of defined as \( \frac{n!}{(n-k)! \, k!} \). This means that it returns the possible combinations given \( k \) times in our case it is being the number of stations. This will be assigned to the variable combination. All the combinations will then be displayed using the display function.

    combination = \begin{bmatrix}
        1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 2 & 3 \\
    \end{bmatrix} \\
    combination=nchoosek(1:nstops,2); \\
    display (combination)

• Hypot function is another function within the library that computes the hypotenuse by taking the first row of \( y \) coordinates from the combination given and distract them from the second raw. Again the first row of \( x \) coordinates from the combination given and distract them from the second raw. It uses that equation in order to avoid underflow or overflow. This function gives a linear distance using the Pythagorean theorem by square rooting the sum of the squares of the other edges in the triangle. Then using the function display again we get the distance.
\[
\text{dist} = \text{hypot}(Y\text{-coordinates}(\text{combination}(:,1)) - Y\text{-coordinates}(\text{combination}(:,2)), \ldots \\
X\text{-coordinates}(\text{combination}(:,1)) - X\text{-coordinates}(\text{combination}(:,2)));
\]
\[
\text{Dist\_numb} = \text{length}(\text{dist});
\]
display(\text{dist})
\[
\text{fprintf(''}\text{the number of trajectories possible is \%d \n',\text{Dist\_numb})}
\]

- In this case, the spones is a sparse matrix that replaces the 0 elements with one. It creates the same sparsity as combination, and then assigns it to Aeq. Hence, it creates a column that is n-combination long that is full of zeros. Then we set the A-array to be nstops long and ncombination large the number of non-zero entries are defined to be n*n-1, given n the number of stops that is in our case equal to 8.

\[
\text{A\_array=spones(1:length(combinationB\_array=nstops;))}
\]
\[
\text{A\_array = [A\_array;spalloc(nstops,length(combination),nstops*(nstops-1))];}
\]

- Once again, a for loop is being used. First if the coordinates from the combination are being used one is allocated in the matrix optimal solution, anf if the coordinates from the combination are not being used it allocates a zero. In this case we have the sparse function which basically squeezes out any zero elements and generates an encapsulated full matrix. This is done mainly to save memory since zero elements will take up storage space.

\[
\text{for j = 1:nstops}
\]
\[
\text{optimal\_combination = (combination == j);}
\]
\[
\text{optimal\_combination = sparse(sum(o}ptimal\_combination,2)); A\_array(j+1,:) = optimal\_combination';}
\]
\[
\text{end}
\]

- This contains two functions: the zeros and the ones. If the van is going to take a certain path. Upper bound will put a one in a matrix using the function one, if not a
zero is going to be put using the function ones

```matlab
B_array = [B_array; 2+ones(nstops,1)];
intcon = 1:Dist_num;
lower_bound = zeros(Dist_num,1);
upper_bound = ones(Dist_num,1);
```

- This will create an optimization for the intlinprog with the display parameter:
  dist,intcon,[],[],A_array,B_array,lower_bound,upper_bound,opts. And saves in a
  matrix in order to plot it

```matlab
opts = optimoptions('intlinprog','Display','off');
[x_tsp,costopt,exitflag,output] = intlinprog(dist,intcon,[],[],A_array,B_array,lower_bound,upper_bound,opts);
```

- This block contains a function of type function assigned to returns to lh under the
  name updateSalesmanPlot. It give it the parameters or the arguments lh, x_tsp,
  stopslon, stopslat, which in itself plots function for tsp_intlinprog.

```matlab
hold on
segments = find(x_tsp);
lh = zeros(nstops,1);
lh = updateSalesmanPlot(lh,x_tsp,combination,X_coordinates,Y_coordinates);  
title('Solution without Subtours');
```

- The below snippet of code is where the detection of the subtours take place. It is a
  function called detectSubtours that take as arguments the two parameters which are x-
  tsp and combination. It is then assigned to the function subTours, but in this case to
  the variable numtours. It will return a cell array of subtours. After executing the
  function, it will display the number of subtours through the function fprintf and
  displaying the # of subtours and replacing the %d by the number returned and
  assigned to the variable numtours. The spalloc function then allocates space for the
  sparse

```matlab
tours = detectSubtours(x_tsp,combination);
numtours = length(tours);
fprintf('# of subtours: %d
',numtours);
A = spalloc(0,Dist_num,0);
b = [1];
```
Finally, this is a nested loop consisting of while, for, and for loops consecutively and respectively. The zeros function along with the spalloc function are being used again. The numtours variable, which was previously assigned to contain the length of the tours through invoking the function length (tours) is being conditioned in the while loop. As long as the numtours is strictly bigger than 1, the loop will hold true, executing the nested loops and their statements. Once the numtour decreases to becoming zero, and we can see that it is appended to the previous result right before fprintf, then the statement will hold false thus exiting the loops. This is mainly because as was mentioned previously, we need to eliminate all the subtours, and it will be done though this nexted loop and hence exiting when it becomes zero.

```matlab
while numtours > 1
    b = [b,zeros(numtours,1)];
    A = [A;spalloc(numtours,Dist_num,nstops)];
    for r = 1:numtours
        rowIdx = size(A,1)+1;
        subTourIdx = tours(r);
        variations = nchoosek(1:length(subTourIdx),2);
        for s = 1:length(variations)
            whichVar = (sum(combination==subTourIdx(variations(s,1)),2)) & ...  
                        (sum(combination==subTourIdx(variations(s,2)),2));
            A[rowIdx,whichVar] = 1;
        end
        b(rowIdx) = length(subTourIdx)-1;
    end
    [x_tsp, costopt, exitflag, output] = intlinprog(dist, intcon, A, b, A_array, B_array, lower_bound, upper_bound, opts);
    lh = updateSalesmanPlot(lh, x_tsp, combination, X_coordinates, Y_coordinates);
    tours = detectSubtours(x_tsp, combination);
    numtours = length(tours);
    fprintf('# of subtours: %d\n', numtours);
end
```

The salesman code of Al Akhawayn university van generates the following linear path:
After I have found the optimal road using the salesman code, I have computed the distance, cost and time spent during the travel by giving an estimation of both the cost and average speed with the following code:

```matlab
cost = 3.5
Optimal_cost = cost * sum(dist)
fprintf('The optimal cost for the van’s travel given the fuel cost is \$%.2e.', Optimal_cost);

average_speed = 50
Optimal_time = average_speed * sum(dist)
fprintf('The optimal time spent during the travel is %.2e.
', Optimal_time);
```

To get the following:

The optimal distance for the travel is 5.9592

The optimal cost for the van’s travel given the fuel cost is $2.979582e+02.$

The optimal time spent during the travel is $2.979582e+02.$
• The optimal distance that is generated by our code is too small compared by the one given by google maps. We can explain that by the fact that my code uses the Euclidian distance, and hence it is a linear one, while Google Maps uses the real distance that takes into consideration the road. For the reasons listed, I will be trying to work with another software that is Javascript.

Salesman using Javascript:

The code to solve the VRP problem was implemented through the use of JavaScript. There are various ways to exhibit the VRP solution at the level of the implementation, but JavaScript was chosen due to its major advantages.

First and foremost, JavaScript is considered to be a programming language rather than a scripting language, and it runs in most of the modern browsers that includes Chrome, Safari, or Mozilla Firefox. It supports procedural and object-oriented programming. It inherits the name from the OOPL Java but is not related to it, and it is used to control and render the client side web pages. It can also control the server side programs. JavaScript could be used and integrated with AJAX, CSS, or HTML. In this project, JavaScript was used along with HTML code in in order to perform the needed requirements and be run locally through a .html file.

In this code, the Google Maps API was used. In order to understand the misconception of what an API is, one has to grasp the client server model in order to understand why this API was used as part of the code. An API stands for Application Programming Interface. The naming confuses people to a certain extent making them think that it is an application whereas
it has more to do with the remote server of, in this case, Google, since we are using the Google Maps API. (Dincer et al, 2013)

When it comes to interacting with a website, this process is being modeled under the client server model. Meaning that we as users interact with the server where the requests are being processed, and we interact directly with the browser. (Dincer et al, 2013) The browser serves as the client side in the server client model. Whereas the server is just a remote computer, meaning a computer that is located somewhere within the Google company’s headquarters and is optimized to only process the insurmountable requests it gets over the world every second. When the server received the request, it reads the code and displays the corresponding page through rendering it in your browser. Meaning that this request goes out to corresponding website’s remote server, and again, in this case, it would be Google. (Dincer et al, 2013)

When looking at this process from the client’s side perspective, then the remote Google server that the browser is connected to and send the requests to is an API. Meaning whenever requests are sent, you are interacting directly with an API. It is the part within the server that processes the received requests and sends back the corresponding reply. (Dincer et al, 2013)

With the Google Map API, the webpage will be able to collect the corresponding needed information from the Google server through your server since APIs allow your own server to connect to other remote servers. The local server will then receive response from Google server and render the requested information. Through using the Google Map API, then it means that the local server will be acting as the client since it will send the request to the remote Google server. (Dincer et al, 2013)
The browser, by default, expects responses in terms of HTML through following a standard HTTP protocol. When using an API, XML or JSON type of markup languages will tend to translate the functions that are requested in order to make sense and standardize the protocol, hence the interaction between the client and the server. (Dincer et al, 2013) The Google Map API will allow the user to use the publicly available functions and methods and hence completing actions without the necessity of leaving the website. (Dincer et al, 2013) This means that Google had enabled us to use their APIs as developers or customers, in order to make use of their dedicated URLs and have the requested data as pure data responses rather than plain HTML responses. (Dincer et al, 2013)

Therefore, the misconception in API derives from the A which stands for Application, and it mainly tends to make people think that it is an application by itself provided by a specific company. However, in API it means that it is a piece of software that has distinct functions and those functions could be called remotely through the use of a standardized protocol.

Therefore, it is essential to use the available Google Maps API in order to call for the map and pinpoint the locations through the map. This API will allow us to easily connect the depots through the use of the algorithm and visually render the result in a dynamic manner to the user in order to make sense of the routes. This falls through the spirit of not reinventing the wheel and also presenting a dynamic graphical user interface (GUI).

The text editor that was used in order to display the code is Sublime Text 3. There exist several other text editors, but this one is user friendly and contains several add-ons that could installed along with extensions that enables the suggestions bar along with other helpful tools.
When opening the file using a given text editor, the JavaScript integrated with HTML file will be opened and the source code will be visible. Through exploring the code, several headers and functions were used.

The .html document will contain several tags and these tags are considered to be containers. The standard way containing all the necessary code in an encapsulated manner within the headers is:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Document Title</title>
</head>
<body>
  The encapsulated code
</body>
</html>
```

Within the html tag, there is the meta tag which is basically a tag that provides the user with the metadata of about the .html document. The style tag is a tag that provides the information about the style of the .html document, and as could be seen, it includes several functions that specifies the panels’ margins, sizes, its font sizes, and floating points. (W3Schools, 2017)
The div tag is a tag used as part of CSS since it is not supported by HTML5, and it simply aligns and groups other tags in the HTML code and groups elements. (W3Schools, 2017) The select tag is a tag that creates a combo box, otherwise known as the drop down list. The option tags that are being contained inside the select tag give the user the possible options to select from the given list. The <br> tags are used only to break lines between one paragraph and another. The input tag is used in order to allow the user to input data. (W3Schools, 2017)

Now is the time to talk about how JavaScript has been integrated within the HTML code. The standard template of a JavaScript code is as follows:

```
<!DOCTYPE html>

<html>

<head>

<script>

function myFunc(){

    Piece of Code

}

</script>

</head>

<body>

The encapsulated code

</body>

</html>
```
Or it could be embedded within the body of the HTML code, as follows:

```html
<!DOCTYPE html>
<html>
<head>
<title>Document Title</title>
</head>
<body>
<script>
function myFunc(){
    Piece of Code
}
</script>
</body>
</html>
```

The piece of the JavaScript code could be distinguished by its script header, meaning any code in between `<script>` and `</script>`.

A couple of functions are used in the JavaScript code in order to implement the algorithm that would lead to mapping the VRP problem on the map at the left side of the screen. The function

```javascript
function initMap()
```
is used in order to initialize the map and use the API provided by Google Maps. It is done through instantiating the objects google.maps.DIrectionsService from the class directionsService and google.maps.DirectionsRenderer from the class directionsDisplay. These two objects communicate directly with the Google Maps API service and it receives direction requests that are directly requested from the client side, and in this case the requests done through selecting the cities from the lists entitles “Waypoints” and then renders it at the level of the browser. The map is initialized by instantiating the object google.maps.Map which asks the server to receive the map provided by the API and later the maps I being displayed through the use of the method directionsDisplay.setMap(map).

Next up is the following used function:

function calculateAndDisplayRoute(directionsService, directionsDisplay)

The above function mainly creates an array of checkboxArray that gets all the elements with the ID ‘waypoints.’ Then a loop will simply travers all the elements within the check box and calculate the number of waypoints, meaning that it will stop right before reaching the length of the checkboxArray variable.

The directionService method will get the value of the starting and ending points, which are by default Virmousil Maroc. Then set the travelling mode to driving and optimize the waypoints distances. The function that takes as arguments (response, status) will then basically return the status of the travelled distance between the depots or the waypoints in order to design the summary panel that includes the Route Segment as the title of the panel with the corresponding segment number. The following is an example of the output of the summary panel that will be provided once the Submit button has been clicked:
Route Segment: 1

Route 3, Tanger, Maroc to Avenue Moulay Rachid, Tanger, Maroc

6.0 km

This will give a friendly user experience in the sense that it will summarize to the user the depots traversed.

Last but not least, the map will have all the functionalities that are provided by Google Maps API such as zooming in and out of the map, and the user could also switch between plan and satellite visions. The route will be drawn as an output in blue with lettered pins in the map that corresponds to every depot or chosen waypoint as shown below:

A small comparison between the results conducted by Javascript and Google maps:
• Google maps cannot add more than ten stations, while our code can generate the optimal solution of any number of stations.

• Google maps does not find the optimal road of all stations, but does generate it only from one station to another.

• If you change the order of the stations in google maps, you get each time a different optimal solution, while with javaScript the same solution is generated.

• The distance generated by google maps is much bigger than the one found by Jabvascript.

As you can see above the distance generated by Google Maps is 48.7 km

As you can see above the distance generated by Google Maps is: 59.8 kms, while the one generated by JavaScript is equal to 33.4 kms, the sum of the segments below:
After juxtaposing both the results gotten by JavaScript and Google Maps, and in order to make sure of our conclusions. I have made another comparison using other stations.

To get the following paths:
Again, there is a very important difference in the distance between both paths. The minimum trajectory was corresponding to the path generated by JavaScript. In fact, in order for Google Maps to get the same solution generated by the JavaScript code, the driver has to know the road he is going to take. Hence, he will know what station he is going to take first and which one next because as stated earlier it does not take into consideration the optimal distance between all nodes.

**Conclusion:**

In conclusion, the vehicle routing problem, though is an NP hard problem, will determine the shortest sets of routes in order to hit all the depots optimally though decreasing the company expenses on its transportation means while keeping in mind the constraints of capacity, time window, and distance.
Two implementations of the VRP were presented in this paper. The first one through MATLAB and the second one through using JavaScript. The MATLAB output does not in essence give a dynamic routing output in that it displays to the user sets of straight lines from one depot to another. Meaning that it is not as accurate as it should be, though it displays the optimal routing.

On the other hand, the JavaScript implementation gives a real life version of the routing that should be done in order to traverse the selected waypoints. It uses the Google Maps API in order to initialize the real life map and retrieve the corresponding routing through the use of the algorithm that is integrated within the JavaScript code. Since it uses API, it means that the map is ready and the only things that need to be plotted are the waypoint points on the map. Meaning that the output will not be set of lines, but clean routing that traverses through real life roads since an API will provide a dynamically plotted output.

All in all, the juxtaposition of both the JavaScript and the MATLAB implementations were sufficient enough to mirror the VRP problem in real life situation.
Reference Page:

Conner, (2017). STEEPLE Analysis (used with SWOT). [online] Slideshare.net. Available at:


