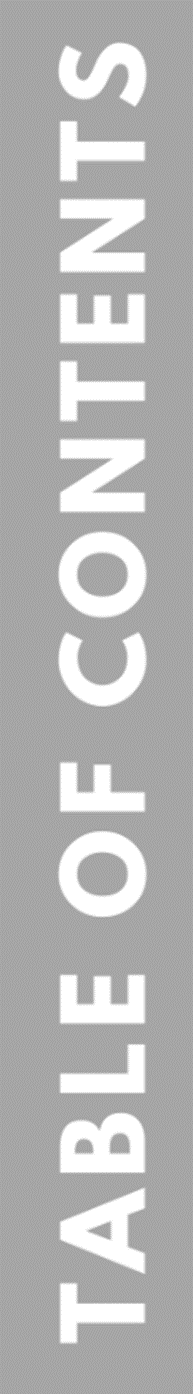
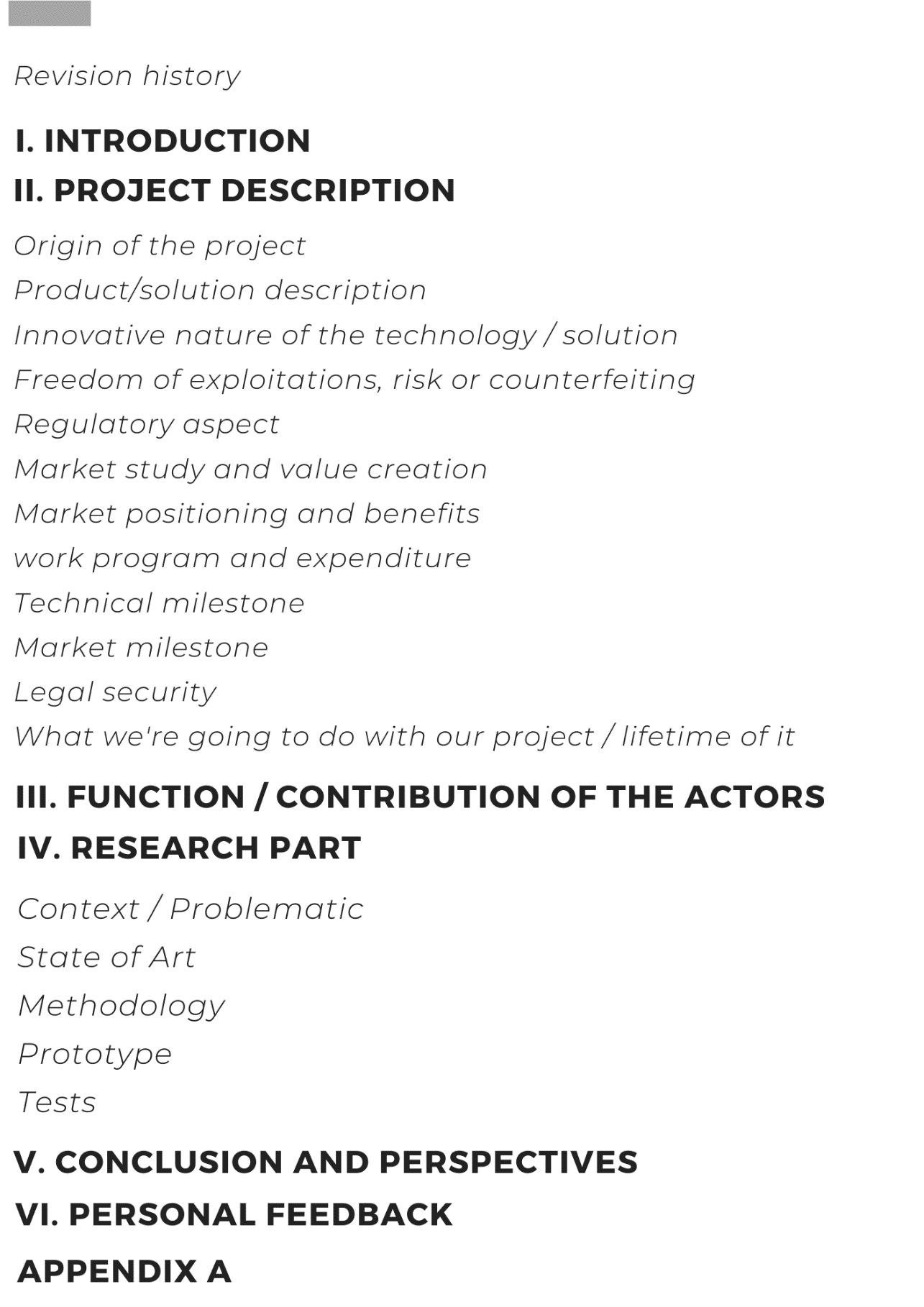


Revision history

|  |  |  |
| --- | --- | --- |
| Name | Date | Changes |
| 1rst version | 07/11/2019 | Creation of the document (p.1-p.6) |
| 2nd version | 08/11/2019 | Project descriptions + Tasks Distributing |
| 3rd version | 12/12/2019 | Technical feasibility |
| 4th version | 01/04/2020 | Final version |





I. Introduction

*(Parts in violet concern the research part of our project)*

A liberal nurse has a daily morning routine of visiting 25 to 30 patients a day at their place. This accounts for about a thousand kilometers they must drive weekly, for the sole purpose of getting to their patients’ home. The driving part represents a good third of their time.

We then asked ourselves: is there any way to give time to liberals, for them to do what is meaningful? This is with that in mind that we designed Trav’Help, and that we started looking for a way to optimize liberals’ schedules.

This optimization problem is related to the Traveling Salesman Problem, which is a famous NP-complete one\* that has been sought by many people for a long time. However, it has a less known variant: The Traveling Salesman Problem with Time Windows. This problem is about finding the shortest possible route that visits each dot, while respecting some time constraints for each dot. This problem fits exactly our optimization problem for liberals’ schedule, where every dot – in our case, patient – must be visited when available.

There are already some implementations of solutions to this problem. However, as we need to find the travel time between two patients, we must use an API. The implementations we found did not quite match our expectations. Indeed, they would call all needed distances, regardless of whether they are needed or not. It gets expensive to call those APIs that much. Furthermore, we used different parameters than them.

Because of that, we decided to implement one solution ourselves. Its explication is further up in the document.

This project aims at tackling different societal aspects. First comes the social one: liberals would stop spending their time in driving, rather than in their actual job: meaning services of higher quality. It also has an economic aspect, where liberals become more time efficient as they drive less, thus consume less. Finally, reducing liberals’ car consumption also helps in decreasing greenhouse gases emissions, as well as noise pollution.

This document will give complete information about our project, Trav’Help, and its modifications. We will regularly recap our different missions and put a glossary with reference sections so it will be easier to understand our project.

Happy reading.

*\*refer to the glossary*

Reference documents

|  |  |  |  |
| --- | --- | --- | --- |
| **Document** | **Number** | **Attached?** | **Application** |
| Document name | Code, number, version | Yes/No | The role of the document relative to the CDC |

Glossary

**Terms**

|  |  |
| --- | --- |
| NP-complete problem | A problem is said to be NP-complete when it is easy to find if a solution is the best, yet difficult to find this solution. |
| Clustering | Clustering is grouping a set of objects in a way that objects in the same group (called a cluster) are more like each other than to those in other groups. |

**Acronyms**

|  |  |  |
| --- | --- | --- |
| **Acronym** | **Meaning** | **Explanation** |
| TSPTW | Traveling Salesman Problem with Time Windows |  |

II. Project Description

## Origin of the Project

We have met several nurses in liberal profession. They want to reduce the time spent in driving. We want to optimize their schedule to see each of their patients with different addresses in a minimum of time.

## Product/solution description

The solution to this problem is a mobile application which takes every address, as well as each patients’ availability, to find the best route which visits every address once and go back to the nurse's home in the shortest time.

## Innovative nature of the technology/ solution

The innovative nature of the project lies in the implementation of the TSPTW problem. This part is defined further up in the document.

## Freedom of exploitation, risk of counterfeiting

We are free to exploit this idea, as it is new.

Risks of counterfeiting are low because our idea is original, and not spread. We have not published our researches yet.

## Regulatory aspect

There is not any issue, political or ethical, in France or internationally, that would be an obstacle to our project.

It does not involve any tendentious topic; idea is new and has no other impact than reducing liberals’ amount of driving.

## Market study and value creation

Our research has shown that there is no existing patent on the exact same topic, and no custom solution to liberals was designed yet.

Some patents already exist on a somehow related issue, but they propose a wholly different solution than ours.

## Market positioning and competitive advantages/ Environmental cost/ benefit

It would help to reduce pollution by keeping liberals from using unnecessary transportation.

They would have more time for their work, which means higher quality of cure, and better relations with the patients themselves.

## Work program and expenditure

Expenditure-wise, we have not spent any money yet during the test period.

We have a partnership with Google. They lent us 300€ worth of API calls for a year. This amount is more than enough for the testing part.

If we were to commercialize the application further, however, we would need an actual contract with Google API for our application to work. We need the distances and times they provide us for our application to work.

## Technical milestone

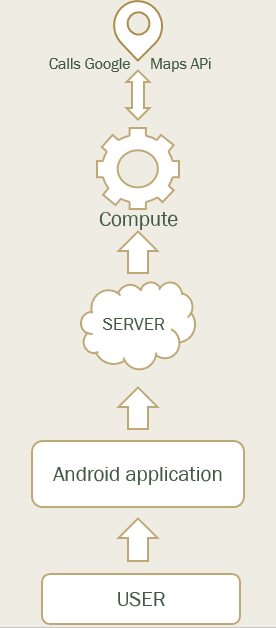
Our first goal is to implement the algorithm defined further up in the report.

We want our solution to be able to compute the best route for a theoretical case, and to display it. We also want to be able to test this application with different sets, to do performance reports.

In a second part, we would like our final solution to be as close as possible to the architecture defined below.

Liberals could use an Android application to save their patients and communicate with the server to retrieve the fastest route. It could display the route in an intuitive and aesthetic interface.

The server would compute the route, using the implementation of our algorithm, and the distances given by Google API Distance.



## Market milestone

After having conducted the project to its end, we shall look for partners in the liberals’ world by introducing them our application, and possible common benefits both parts could take out of this project.

According to their answer, we will rethink the way we designed the project.

If they think the project is worth sharing, we may make it available to public, most probably thanks to the Google Play Store platform.

## Legal security

For the moment, we have not registered our algorithm in a Soleau envelope, nor have patented it.

However, given we have not published our results yet, we do not see any legal security issue in the near future.

If the project is to grow bigger, we will rethink our position towards this point.

## iWhat are you going to do with the product/solution at the end of the project? / What is the lifetime p roduct /solution?

It has a double goal.

First, on a practical aspect: it depends on liberals’ contacts we may have. If they consider our project worth sharing, we definitely will consider doing it. If they fail to see how our project may help them, we will probably put the project to its end.

Secondly, this project has a research aspect. Indeed, the research we have done is new and might be useful. This is here to stay.

Regarding PFE: we do *not* consider pursuing this project into a PFE one. It is not about the project in itself, rather that 5 of our 6 members will go for an exchange semester into another country.

III. Function and contribution of the actors

Karla FLATRES

Background:Finance

Function and contribution:

* Take care of the research and documentation part
* Design of both posters
* Write deliverables

Motivation and personal commitments:Curious about the Salesman Problem, I was wondering how we could approach the problem. The fact that we studied clustering in finance made me think that it could be possible to solve our problem. With my VP communication skills, I can also bring my help for all the designs we have to make.

Marc-Antoine HERVÉ

Background:Finance

Function and contribution:

* Theory and implementation of the algorithm
* Finalize algorithm, tests and research part

Motivation and personal commitments:Initially interested in programming and machine learning, really thrilled about the ideal of looking for an old math problem and willing this Project the further it can get

Pierre LELIÈVRE

Background:Information Systems (SI)

Function and contribution:

* Project manager
* Finalize algorithm and research parts
* Code the server part
* Code the mobile application part
* Write deliverables

Motivation and personal commitments:Initially, the idea came from the will to enhance liberal profession's working routine, as Pierre has some family directly involved in this field. Furthermore, he enjoys his team manager position and wishes this project will result into a great experience.

Robinson MATHIEU

Background: Information Systems (SI)

Function and contribution:

* Theory and implementation of the algorithm
* Code the mobile application part
* Write deliverables

Motivation and personal commitments:interested in the Salesman’s problem and about all the mathematical formulas and knowledge that we can use to resolve this problem. It remembers him the period of his preparatory class.

Quentin ROCHA PINTO

Background:Embedded System (SE)

Function and contribution:

* Theory and implementation of the algorithm
* Code the mobile application part
* Write deliverables
* Work on tests and research parts

Motivation and personal commitments:His main motivation in this project comes from the research aspect. Working on this problem of optimization is very interesting and motivating because it is a concrete problem that can affect all of us. This "Salesman Problem" makes it possible to reach many interesting areas such as algorithmic mathematics and the use of new languages.

Diego SANCHEZ

Background:Energy and Environment (EE); International student from Mexico. Joined the group in February 2020.

Function and contribution:

* Code the server part
* Make the video
* Write deliverables

Motivation and personal commitments: His motivation is to be able to have more knowledge about how to develop an app, in addition to being able to generate value for society with the project we are working on.

IV. Research Part

**Problematic**

As said in the introduction, we want to find a solution to the Traveling Salesman With Time Windows that would satisfy two points :

* Finding an exact solution;
* Work with a data set like a liberal’s journey.

**State of Art**

*An Exact Algorithm for the Time Constrained TSP* by Edward K. Baker (1983)

This research paper was the first one introducing the Time Constrained Travelling Salesman Problem (TC-TSP) using the same kind of model we are using in our research which is a different number of cities (patients house for us) that have to all be visited with the shortest distance possible while taking care of the upper and lower bound of the city which represent the time interval where they have to be visited.

However, it is using a different way of solving the problem which is a bit too simple. In the first place it is using an algorithm to find the distance between all cities with the help of the triangle inequality and referring to the Hamiltonian path. Which lead to the shortest path possible without considering the time constraint of each cities in a first way. Moreover, in this algorithm the next city in the sequence will always be the one that is the closest to the actual city.

This is already different than our algorithm idea because we are using considering the time windows of every “city” at the very beginning of our algorithm and we are not looking for the shortest distance between the two cities at every node but we take all the possibilities and process them with our criteria to get rid of the ones that are irrelevant.

Then, on the second place, the algorithm from Baker’s research receive a shortest distance sequence and will now have to process it by taking the time windows in count. The way it is dealing with those constraint is the following, if a node/city can be visited and is the next one in the sequence, it will be, if not, the traveler will have to wait at the city for the lower bound of the interval to be reached and if a cities upper bound has been outdated then the algorithm will not find a solution.

Our algorithm works in a different way here because at each sequence and at the end of the algorithm we will be reducing the waiting time at the maximum by not taking in count the cities where the lower bound have not been reached and will get rid of the sequences where the upper bound of a city is outdated is the city is not in the sequence yet.

*An Optimal Algorithm for the TSP* by Yvan Dumas, Jacques Desrosiers, Eric Gelinas, Marius M. Solomon (1995)

This paper presents the development of new elimination tests which greatly enhance the performance of a relatively well-established dynamic programming approach and its application to the minimization of the total traveling cost for the traveling salesman problem with time windows. The tests take advantage of the time window constraints to significantly reduce the state space and the number of state transitions. These reductions are performed both a priori and during the execution of the algorithm.

The branch-and-bounds solution given by researches like Baker’s where effective but only for a small number of nodes (50 max) and with very reduce width, this is why other searchers were looking for a more optimal solution of the TC-TSP that could work with larger bounds and with a greater number of nodes at the same time.

The largest difference with our algorithm is the use of a dynamic method that will not provide the most optimal solution for sure and the fact that they are calling all distances at the beginning of the computing process whereas our algorithm provides the exact solution by construction and only calls the distances if there is no other way to compare sequences with each other to find the exact one.

However, this algorithm can compute up to 200 nodes and more than a 100 with a large width of bounds.

*Dynamic Programming Strategies for the TSP with Time Window and Precedence Constraints* by Aristide Mingozzi, Lucio Bianco, Salvatore Ricciardelli (1997)

In this paper, the searchers were looking for an exact way of solving the problem using a solution that is similar from Mr Dumas solution which means by using dynamic programming in order to solve the problem efficiently. Indeed the Dumas’ solution lack some condition to be more effective and some of those solution are related with our criteria to get rid of some sequences that are not optimal : for example the fact that is a node as not been visited yet but the upper bound is outdated we get rid of the sequence.

However, this is still dynamic programming, so it has a major difference with our algorithm even if it improves the dynamic solution of Mr Dumas.

*A compressed-annealing heuristic for the TSP with time windows* by Jeffrey W Ohlmann, Barrett W Thomas (2007)

This paper describes a variant of simulated annealing incorporating a variable penalty method to solve the traveling-salesman problem with time windows (TSPTW). Augmenting temperature from traditional simulated annealing with the concept of pressure (analogous to the value of the penalty multiplier), compressed annealing relaxes the time-window constraints by integrating a penalty method within a stochastic search procedure.

Computational results validate the value of a variable-penalty method versus a static-penalty approach. Compressed annealing compares favourably with benchmark results in the literature, obtaining best known results for numerous instances.

This paper shows how more recent researches about the TSPTW has evolved, with the use of stochastic methods and the use of penalty with the annealing method. This is now of majority of searchers are working to solve the TSPTW but once more, this algorithm will be using the Monte-Carlo method also used in the world of finance to predict stock variation. However because it uses Monte-Carlo which is a stochastic approach it can solve a greater number of nodes but it is not an exact solution (fluctuate with the variation of the k parameter which is the cap ratio).

We clearly see here the different path that the research is taking with this stochastic method and our try with an exact but more “old school” method.

## Methodology

## We consider patient to have the following characteristics: [lower bound, upper bound, address]

## Let us consider the following problem:

## 

Construction of all time-intervals we are going to work on

Each time interval has a lower bound and an upper bound.

Process to sort the array [minA, maxA, minB, maxB, ...] where minA is the lower bound of patient A (the earlier you can visit the patients considering his time constraint) and maxA upper bound of patient A (the end of the patient time constraint)

It leads to this kind of construction:



Where each interval will be a loop of the program during computation, first one with [8,10] then [10,11], etc.

Loop over the different time-intervals

Between each interval of the sorted array, we:

Consider available patients for the time period

For each of those patients, we try building sequences using them.

In our example, patients available in the first time-interval are : [2, 5, 7, 8]

But for simplicity, let’s consider there are only three patients, 1, 2 and 3.

We try out every possibility/combination of sequence (including those that do not take all patients available in it).

In our case, we get :



(With 4 patients, we would have had 65 different sequences).

In some sequences, some patients are missing. That is because, they are still available in the next time interval : meaning, they could be visited afterwards. We have to consider this case in order to do all the cases.

For simplicity again, let us consider that patients 2 and 3 *have* to be cured in this first time interval. We then have to remove every sequence that is not built out of patient 2 and 3.

The remaining sequences are :



We just finished the first iteration of the loop. Let’s move onto the second passage in the loop.

Second loop

Let’s consider there is another patient, 4, that *has* to be visited during this second time interval, as well as patient 1.

We are going to build all new sequences, taking this information into account, using the same process as last time.

However, this time, the new sequences will be built from the one we had last time ; we simply add all possible new sequences, to the one already existing.



In this case, we notice that some sequences are necessarily redundant, and we are sure that we will not choose this sequence to be the final one.

Identifying similar sequences

Indeed, sequences [3, 2, 1, 4] and [3, 1, 2, 4] are *similar*.

We consider two sequences as similar if they :

* start with the same patient;
* finish with the same patient;
* visit the same patients.

We now have to find which sequence to remove.

Computing time needed for a sequence

We will then compute the time needed for each sequence, in order to know which one is to be removed.

Let’s compute traveling time needed for sequence [1, 2, 3, 4] :

T = T(1,2) + T(2,3) + T(3,4)

where T(*x*,*y*) represents the time from *x* to *y*.

If a T(*x*,*y*) has already been computed no need to compute it again, it will be kept in the distance matrix.

We need the useful distances that will be kept in a distance matrix. Some distance will never be used, thus never called. For instance, in our example, we never need to know T(4,2), nor T(4,3).

Removing similar sequences thanks to the distance

Then, we will go through the list of sequences to look for those that can be suppressed. Arbitrarily, and for simplicity, we will choose those distances for each sequence, and remove sequences that are removable.



Removing impossible sequences

We also delete the strings that does not respect the patients bounds.

Let’s consider a sequence of patients [1, 2, 3, 4] on the time interval [10, 11]. Let’s also consider patient 5, that had to be visited between [8, 10].

The sequence above will be deleted because it is too late to visit patient 5 (works in the same way if it is too early)

At the end of the loop

When all of this is done for every time interval, we should have only one sequence remaining : the final solution.

As we construct, by recurrence, all our sequences without omitting any, we are certain the final solution is the only one that fits the schedule, and that is the faster.

## Prototype

### Environment of development

We implemented the solution above using Python.

### Implementation of each function

Every function’s implementation is pretty basic, and uses no complex implementation, other than following the above algorithm. Source code is available in Appendix A.

### Special mention to the “remove similar sequence” function

Indeed, this function accounts for roughly 95% of our program execution time.

Depending on how full the sequence list is, the function takes a really long while to finish.   
  
Its implementation is about going through the list with two indexes *i* and *j,* and comparing if sequence[ i ] is similar to sequence[ j ]. The complexity of this function is O(n²), which is why it takes a really large amount of time to finish.

### How to use it

To use it, you only have to run the source code.

As exposed before, the profile of the journey will greatly impact performances, as well as time needed.

That is why we implemented to ways to create the journey.

*First way: manually*



Those lines here are used to choose manually the journey’s characteristics.

The blue part represents the distance/time between each point. For instance, if you take the number with coordinates [4,3], it represents the time needed to go from patient 4 to patient 3. Please note that the distance/time to go from A to B is not necessarily the same as the distance to go from B to A, depending on how the routes are done.

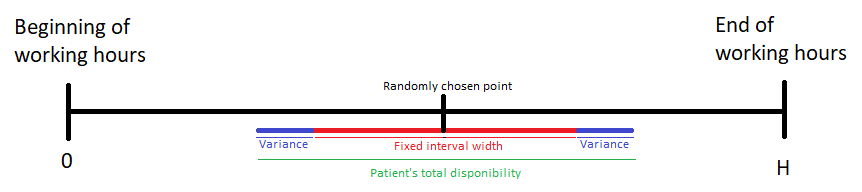
The green part represents the time window in which a patient is available. It means, for instance for patient 1, that they are available from the start until 20 time units.

Using this, we can test custom journeys.

#### Second way: randomly

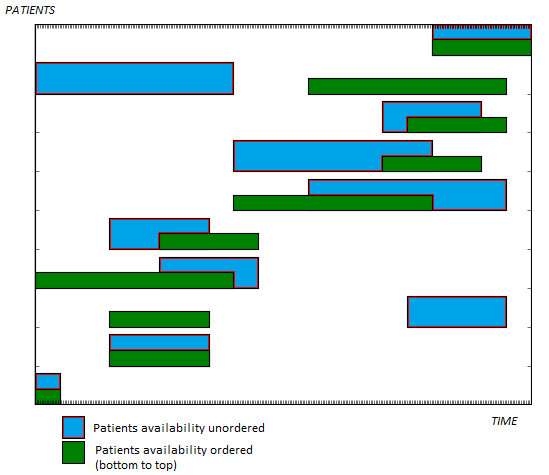
We fill up the same table as above, unless this time it is done randomly, following a certain pattern.

We start by randomly picking a value within working hours. This will be the middle of our interval. Then, we create the width of the interval, using a standard width, to which we add up some variance.



We repeat this process to fill up the journey with each patient’s disponibility.

Eventually, we get a result that looks like this:



**Tests**

Tests have been realized with up to 10 computation for each variation of both the parameters (depending on how long was the computation time).

We decided to evaluate the variation of the compilation time depending on the number of patients and the maximum width of their uptime in percent of the day (the repartition of the width of the intervals have been normally distributed).

The availability of the patients represent a certain time for example if you consider the nurse to be working for 6 hours during her journey, from 8am to 2pm for example, a patient that has a 50% availability is available during an interval of 3 hours during this day (from 9am to 12am for example).So, a 100% availability means the patient has no time restriction at all.

Test were made with an Intel® Core™ i5-7300HQ CPU @ 2.50GHz

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 16% | 33% | 50% | 66% | 83% | 100% |
| 5 | 0.28 ms | 0.33 ms | 0.80 ms | 1.4 ms | 2.5 ms | 3.4 ms |
| 6 | 0.30 ms | 0.46 ms | 0.87 ms | 1.6 ms | 2.7 ms | 4.6 ms |
| 7 | 0.32 ms | 2.0 ms | 44 ms | 1.7 s | 9 s | 10.3 s |
| 8 | 0.32 ms | 2.3 ms | 0.5 s | 3.27 s | 4 min | 7 min |
| 9 | 0.33 ms | 2.6 ms | 0.9 s | 32 min | X\* | X |
| 10 | 0.36 ms | 2.7 ms | 2.7 s | X | X | X |

\*X means that the compilation time was to high to be relevant or even to be measured

We can observe that the computation time stays very low before 7 patients with up to a full day uptime but starts to increase to the point that it cannot compile starting at 83% interval width for 9 patients and 66% at 10 of them.

As an indicator, for 15 patients with a width of 33% it took an hour and 40 minutes to compile.

We have tested out all functions of our algorithm and found out that the remove similar function was taking more than 95% of that computing time because our method ( going through the whole list of sequence every time we have to remove a sequence) was taking way too many time, especially when the length of the list was exceeding 10000.

# V. Conclusions & Perspectives

To conclude, this algorithm is unique because it does not give an approached solution but the best solution by mathematical construction and by recurrence, as no approximation is made across the whole algorithm.

It also reduces the number of data called (here distances), because it only computes distances needed for the algorithm to find the best solution.

However, yet the execution time for it to be effective on real nurses’ journey is too big (up to 25 patients a day for a nurse).

Despite that, it should still work for some liberals that have less patients a day, but more time spent at the patient’s house.

In order to make it usable for nurses as well, in the future the main goal is to reduce the computation time. Indeed, we must search for a different method than using a list to remove our sequences or just finding a way to do it more efficiently.

Another goal of our project is to improve our algorithm for it to be effective with patients with discontinuous availability.

Which means rather they have 2 or more periods of availability time of maybe they have to be treated more than once a day for some reason.

We have managed to implement the algorithm the way we wanted.

Furthermore, the implementation of the full architecture is already started and is not far from being done already.

In this document, we described precisely our project (the main problem and the final goal)

and what we need to design it. We have seen the different documents needed to understand and start in a good way the project. We also described all the actors of this project and their contributions, motivations to the project. Each member has something to add to the project and thanks to all the environment that we are going to implement for the testing phase, we should be able to make and test this algorithm. This algorithm will be the result of the coordination of the team, the researches on the subject, the searches on the code and its optimization but also the guidelines of our mentor.

# VI. Personal Feedback

Karla FLATRES

At the beginning of the project I was worried because this is my first big technical project and I thought that we wouldn’t have the time to finish it.

I felt a little bit lost at the beginning because I didn’t know which new approach we could bring to the Existing Traveling Salesman Problem. For me, the biggest difficulty was to find the documents for our exact problematic, I wasn’t sure about what I was really looking for. Thanks to the help of the group I could go through it and understand the real meaning of our subject at the beginning.

Thanks to the documentations, I learnt a lot about time windows and how it works. The other thing I learnt is, if someone doesn’t give any information about what he does, it slows down a lot the project. We needed to be organized and thanks to our Project Manager, Pierre, it was the case. Knowing that our project has found its way make me feel proud of our work.

Marc-Antoine HERVÉ

Through all this project I have learnt a lot of things and a lot of way to do them. As I am from Finance, a lot of the tools used to code in group were still unknown to me and it was quite a burden to understand how to use GitHub correctly or only to understand (even a little bit how java script or network works). Then, the coding part was very entertaining, and it was interesting to discuss about how to find new solutions to our problems. I also learnt how to document myself with research paper which can only be useful for the future. Overall, this will remain a good experience and I think our subject still has a lot to offer.

Pierre LELIÈVRE

Personally, this project taught me a lot in many fields.

Being the team leader forced me to know a good bit in each field, as I needed to be the bridge between different parts of the project.

It is really demanding to be able to be competent in all fields, but results show up when you manage to explain to different parts what their role is.

There is a lot of pressure, as people usually refer to the team manager for everything they need, which eventually adds up to making a lot of work.

Managing people also isn't easy, with everybody having different schedules, skills, motivation and personalities. But it sure is a really interesting role!

Robinson MATHIEU

Personally, this project made me grow, especially in the way to handle the different steps. More precisely, this project made me learn to take it step by step and not trying to finish the project in one row. For a project like our own, it is clearly impossible. I learned to work in team and to include people in the middle of our project like Diego. And it was also a challenge to find my place in the group of the PPE, each person has different skills and you have to assume and highlight yours to be able to be at the right place. In addition, this project made me realize that even if we have different views of different people, we have to be faithful to our project and not be disturbed by them. We have a goal; we have to be true to plans of the founders.

Quentin ROCHA PINTO

For my part, I found this PPE very interesting and exciting. This is the first time that I have invested so much in such a big project, it was very rich in experience and very enriching. This PPE allowed me to put into practice what I learned during the theoretical courses, and deepen what I practiced in practical work, it being in the purely computer courses but also the project management courses.

The social aspect of the project, through group work, was also very interesting. Having done this project with six people allows us to understand the complexity of teamwork but also, and above all, its strengths. Despite some differences on the way to proceed, to distribute or to code, it seems to me that we have been able to take advantage of this subject in order to build a complex and successful work (as best as possible).

The time to do this project still seemed to me a bit short, given its complexity. But that seems obvious because we always wanted more in order to have the most successful work. I think that by continuing this project a few months we could have a more complete, faster rendering, with more options, more aesthetic and more ergonomic.

For my contribution, the study of the traveling salesman problem and the implementation of the algorithmic solution was also very interesting. However, it is on the coding part of the application on Android Studio that I am mainly focused. This was new to me, I had never coded an application, and therefore never used Android Studio. I learned as this project progressed. It was not simple (a lot of problems due to the lack of knowledge of this coding style, resolved little by little using a lot of tutorials and dedicated website for hours and all year), but very interesting to learn a new way of coding such a concrete project.

I am therefore overall, very satisfied with the progress of this project and the result that we were able to produce as a team.

Diego SANCHEZ

During this project, you can gain new knowledge about areas where I was not so familiar. Being a project focused on another major of mine, I had difficulties on certain solution processes. Despite this, I take the project as a very positive experience due to being something unusual in my academic life. Being an exchange student, with this project, I can take with me knowledge that could be applied in my professional life, but above all, the experience of being able to work in another "field".

What I could learn from the project was to improve my programming skills a bit. As it was a new programming language for me, I had certain difficulties, however with the help of my colleagues I was able to adapt better to the project. What I liked about this project was its objective, since with our solution we would be generating value for society and helping a group of vulnerable people.

# Appendix A. Python Source Code

[https://github.com/Azorlebleu/ECE\_PPE](https://github.com/Azorlebleu/ECE_PPE?fbclid=IwAR0kp8SK22vGgLSBcA1jgE-e6yALBbzKON9dB4VtypZjADLODACueBBVHEc)