RAPPORT DE STAGE

Année universitaire 2016 / 2017

Nom et Prénom de l'étudiant :

Elie ABI HANNA DAHER

Année d'études :

☐ L3 INFO ☐ L3 INFO APPR. ☐ L3 MATHS

■ M1 MIAGE ☐ M1 INFO. DECISION ☐ M1 INFO APPR. ☐ M1 MATHS

Organisme d'accueil : LAMSADE

Titre du rapport : Internship Report – UTA Method

Tuteur de stage : M. Olivier CAILLOUX, M. Michel ZAM

Dates du stage : 29/05/2017 jusqu'à 01/09/2017

Université Paris-Dauphine – Département MIDO
Place du Maréchal de Lattre de Tassigny – 75775 Paris Cedex 16 – www.mido.dauphine.fr

EFMD
EQUIS
ACCREDITED

Abstract

This report represents the work done during the internship at LAMSADE Dauphine, under the supervision of Mr. Olivier Cailloux and Mr. Michel Zam. The report gives an overview of the projects completed during this internship.
I have worked mainly on the UTA method, this procedure is an ordinal regression technique, proposed by E. Jacquete-Lagreze and J. Siskos, which adjust a system of additive value function.

Acknowledgment

Contents

Chapter 1

Introduction

Currently I'm completing my master's degree in MIAGE at Paris Dauphine University so I had the opportunity to accomplish an internship at LAMSADE (Laboratoire d'analyse et modélisation de systèmes pour l'aide à la décision). LAMSADE is a Paris Dauphine research laboratory specialized in decision theoretic approaches. Decision theory aims to help individuals who face decision problems by elaborating a study of the reasoning and underlying agent's choices.

From 29th May, 2017 to 1st September, 2017, I completed a research internship at LAMSADE. The internship was held under the supervision of Mr. Olivier Cailloux, a lecturer in computer science and researcher at LAMSADE and Mr. Michel Zam, the founder of KarmicSoft and an associate professor at Paris Dauphine University. I worked on my internship remotely, a weekly meeting was scheduled with Mr. Cailloux to get feedback on the work done. And I communicated with Mr. Zam via mails and calls.

This internship is very important for my professional career because it may be my last internship before I get my Master's degree in Information systems for finance. In addition, this internship allowed me to practice the different theoretical courses learned throughout my academic years.

One of the objectives of this fellowship is to propose the UTA method, that solves a multi-criteria problem by building a value function based on the preferences of the Decision Maker, as an open source software component *uta-calculator* on GitHub. Another major aspect of my internship was the research aspect, using *uta-calculator* we should be able to generate a simulation and make an interpretation of the results found.

In this internship report, I will describe the context of the project which contains an overview of the fellowship. While writing this report, I will describe the projects made: MyDraft, UTA, Linear Program Solver, *uta-calculator* and the research realized that is based on the simulations made. Finally, I will conclude by reflecting on the flow of my internship.

This report was supposed to contain a section called *Organization and Communication*. Since I did do my internship remotely and I had only contact with Mr. Olivier Cailloux and Mr. Michel Zam I didn't have enough information to develop this section. So, as addressed by my internship manager, Mrs. Manouvrier, we replaced the section *Organization and Communication* by *Context of the project* that illustrates a quick background of the projects done.

Chapter 2

Context of the projects

2.1 Introduction to LAMSADE

LAMSADE is a laboratory located in Paris Dauphine university. LAMSADE stands for Laboratoire d'analyse et modélisation de systèmes pour l'aide à la décision. The French laboratory has been established in 1974 by Bernard Roy. The main research subjects studied are operational research, decision aid, artificial intelligence (AI) and decision theory.
The activities of LAMSADE are divided into 3 poles:

- Decision aid

- Combinatorial optimization and algorithmic

- Data Science

LAMSADE is well known for the origin of the multiple-criteria decision analysis (MCDA), and for its approach of the Algorithmic Decision Theory. The laboratory has established itself as the best research center internationally in it domain.
LAMSADE is structured in two dimensions:

- Scientific animation that represents the poles

- Research that represents the scientific projects

Scientific dimension represents the poles, where there is a community of researches invited to exchange throughout meetings and seminaries. The research dimension represents the scientific projects assembled by the researches interested in the subject in question. Both of those dimensions are neither fixed neither permanent, they can be submitted to an evaluation by the Laboratory.

2.2 Objectives

Throughout my internship, I had several objectives to achieve and those objectives changed from when I began the internship.
The following list represents the objectives stated at the beginning of my fellowship:

- Propose an open source software component of the UTA method (Main objective)

- Integrate the UTA open software component as one of the tools in the Decision Deck community

- Document, publish the software and implement a graphical user interface to the UTA by using DecisionCloud, a tool promoted by the Decision Deck

So as stated before, the objectives changed from the beginning of the internship for many reasons. We ended up doing an open source software component of the UTA method that we published on GitHub instead of the DecisionCloud. This gave us the opportunity to evolve the software by doing more functionalities, and allowed us to make a simulation that was the main

reason of our research. The main reason we weren't able to make the tools in the Decision Deck community was because Mr. Zam wasn't available to assist me. So, we took this opportunity to make the software more developed and elaborate a simulation and comparison of the results obtained.
During the internship the objectives were finally re-stated as follow:

- Summarize the UTA method

- Propose an open source software component of the UTA method: uta-calculator

- Generate Random alternatives and criteria

- Generate random Value Function v^R

- Generate value function v^T from UTA algorithm

- Compare the two value function: v^R and v^T and evaluate it mathematically

- Search literature for existing similar studies, and how to generate realistic random alternatives

To sum-up, our main objectives was to implement the method UTA and make simulation to compare the two value functions generated v^R and v^T. The simulation of the software was part of the research that helped us to compare the two value functions.

2.3   Sharing of information

All along the internship, all the work done was submitted on GitHub so it can be seen by my supervisors. We held a meeting approximately each week, Mr. Cailloux and me, to discuss the conduct of the project and to get a feedback from him. Since I didn't have any colleagues to work with and only two supervisors, the communication was straight forward with my mentors. Only when I was having some serious difficulties that where complicating the work, I discussed with Mr. Cailloux via emails to solve it before the meeting.
Since Mr. Zam wasn't available during this summer at Paris, we shared some emails and calls to talk about the progress of the internship and he helped me with MyDraft.

2.4   Conduct of the internship

On the 29th May, 2017 I started my internship. The first two weeks, were dedicated to get to know the platform MyDraft and create some basic applications. MyDraft is a platform created by KarmicSoft, and is specialized in building and running data-driven Web applications. It is a platform for developing business applications the easiest way without complexity. During those two weeks, I did all the tutorials available on the site and in addition I created my own program. The project was called *StockApp* where I recreated a project done during the course of *Marché Financier*: Portfolio Management Project.

After learning about the platform MyDraft, I had my first meeting with Mr. Cailloux on the 15th June, 2017 where he introduced me a little bit to the UTA method and gave me the following book: *Multiple Criteria Decision Analysis*, that contained a chapter explaining the UTA method. I had to understand this method and make a summary about it, so I took this opportunity to learn about LaTeX to create the summary. By the end of July, we created the summary of UTA method in Latex, and created my own problem to illustrate the method. Since one of the steps of the method was to solve a Linear Program, we created an independent project to solve a simple Linear Program (LP) by using the google ortools (Optimization tools) library. Meanwhile, I made some research about literature for existing similar studies and how to generate realistic random alternatives research. During this period, I had a weekly meeting with Mr. Cailloux.

By the end of the month of July, we had a meeting, Mr. Cailloux and me to discuss the software that will represent the UTA method. We designed the architecture, and I coded the program, it didn't take long because during the previous month I had already worked on little components of the software. For the month of august our goal was to finish this document and to make the

simulation and interpretations.

All of the work done is available as open source on my GitHub repository: `https://github.com/elieahd/decision-uta-method`.

Thus, in the next chapter, we will discuss the projects stated above.

Chapter 3

Description of the projects

3.1 MyDraft

3.1.1 What is MyDraft?

MyDraft is a platform proposed by KarmicSoft. The goal of this platform is to develop a data and business driven application built only by click, without the need of a developer for a simple application. The application can be published in the Cloud and have a history of changes made on the data.

Another characteristic of MyDraft is the simplicity of coding which reduces the time required between creating the idea and accomplishing the app. MyDraft goal is to build a cloud-based agile development platform.

MyDraft proposes a unit testing and the possibility to generate UML model of the application. I couldn't test the last feature because of problems in the server located in United Kingdom.

3.1.2 Projects - StockApp

To practice using the platform MyDraft, I worked on my own application: StockApp. The project consists in collecting stock information and calculating the following information:

- Weekly Returns of assets by calculating its mean and standard deviation.

- Correlation matrix of the stock returns

- Correlation of stock returns with the stock market index

- Beta, Sharpe Ratio and Treynor Ratio

- Portfolio optimization scheme of Markowitz

I defined 3 classes in the model:

- Stock

- Data

- Index

The class Stock has the following attributes: id, name, averageReturn, variance, standardDeviation, Beta, expectedReturn, sharpeRatio, growth, averageAdjClose
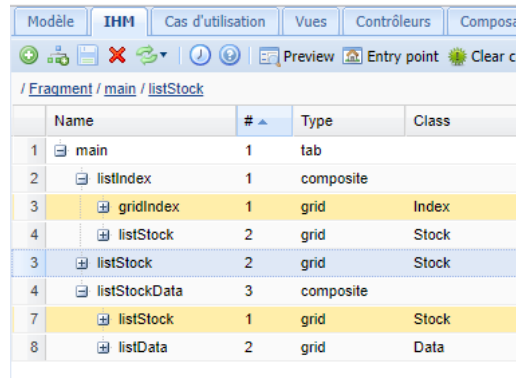The class Data has the following attributes: returns, adjClose, date
The class Index has the following attributes: id, name
The stock has a relation 1 to many with Data, a stock has a list of Data. And the index has a relation 1 to many with Stock.

To achieve this project, I created an operation for the class Stock that calculates all the data needed.

In the IHM, I defined 3 tabs: list of index, list of stock and list of stock and Data.

Figure 3.1 – MyDraft - IHM fragments

I succeeded in doing most of the functionalities of the project except for the last point, Portfolio optimization scheme of Markowitz, this wasn't possible to do in MyDraft since it needed the Macro used in the Excel.

3.1.3 Tutorial

On the Karmicsoft site, we can find a list of tutorials. The tutorials helped the users know how to use the platform. The following list represents all of the tutorials available on the site:

- how to create an account on MyDraft

- how to define classes, objects and fragments

- how to define state and operations

- how to generate a use case diagram

All the mentioned tutorials above were made in French and published to YouTube. So, I took the initiative to create a PDF document in English on how to define classes, object and fragments. I thought that such tutorials would be interesting for non-speaking French and would be beneficial for MyDraft. The tutorials were sent to Mr. Zam.

3.1.4 My remarks on MyDraft

From my experience of using MyDraft, I found this platform very powerful and simple, on the other hand certain categories had some lacks or some bugs. Following is a list representing my remarks and where I think that MyDraft can be improved:

- MyDraft is not consistent with the language, we can find some words in English when the language chosen is French

- not many tutorials are available in the tutorials section, and they are only in French

- most buttons are not labeled, you only find an icon. So, figuring out what each button do is a bit confusing

- there is no community nor documentation to help the users when they face a certain problem or bug

- in certain scenarios, we may need to refresh the page so the changes can take place

- generating UML model wasn't possible during my internship since there was a bug in the main server located in United Kingdom

But overall my experience with MyDraft was very good. This platform can have a real impact on the programming cycle which can lead to some new technology that can help the user to create their own application without the need of a program thus speeding up the process.

3.2   UTA

The UTA method is a multicriteria method that estimate a value function after solving a linear program. The only information needed by the decision maker is the global preference between the alternatives. In the following section, I will summarize this method by introducing the decision theory and then by stating the principles and notation of the UTA method. Then a little numerical example will illustrate the method. Finally, I will conclude this section by explaining what the value function really means.

3.2.1   Introduction to the decision theory

In decision theory, we elaborate a study of reasoning underlying the agent choices. Two branches can be deducted from the decision theory: giving an advice on how to make the best decisions and how existing agents make decisions.
In multiple-criteria decision analysis (MCDA), the following concepts play a fundamental role in decision-making problems:

1. object of the decision, definition of the set of potential actions (alternatives) and the determination of a problem statement

2. modeling a consistent family of criteria

3. defining a global preference model

4. decision-aid or decision support

   Example: Buying a new car
Let's consider we are trying to figure out which car to buy. Using the methodology presented above, we state that the objective of this problem is **buying a new car**. After stating the objective, we can now list the potential actions that represent the list of cars that we may buy. The potential action designates the object of the decision. An action is qualified as potential when it is deemed possible to implement or if it has some interest to the decision aiding process. So, the following list represent the potential actions for this example:

- Peugeot 208 GTi

- Nissan Sentra

- Citroen C4

- Peugeot 308 berline

After listing the list of potential cars, we can define a list of criteria that we will base our decision on. A criterion is constructed to evaluate and compare potential actions according to a point of view. When defining the list of criteria, you should always remember that they must be easy to evaluate (easy to convert to a scale) and should be logical. Let's say we will base our purchase on the following criteria:

- price (in Euro)

- comfort $(0, +, ++, +++)$ *0 being not comfortable and $+++$ very comfortable*

- safety $(1, 2, 3, 4, 5)$ *1 being not safety and 5 safe*

During the decision process, we will determinate the global preferences of the potential actions:

1. Citroen C4

2. Peugeot 208 GTi

3. Peugeot 308 berline

4. Nissan Sentra

Once the global preference is defined, we can start the decision support.

One of the multi-criteria decision analysis methods is the UTA method, which was proposed by E. Jacquet-Lagrèze and J. Siskos in 1982. This method is proposed by the Multi-Attribute Utility Theory (MAUT) that build a utility function based on the DM[1] preferences.

The UTA method is used to solve a multi-criteria problem by building a value function based on the preferences of the DM and solving a linear program (LP). It adopts the aggregation-disaggregation principles: where the model is based on a given preferences.
The UTASTAR, is an improved version of the UTA algorithm. During the internship, we focused on the UTASTAR method rather than the original UTA method.
The aim of the UTASTAR method is to estimate a set of additive value functions which are as consistent as possible with the decision maker's preferences.

At the beginning of the problem, the DM should present the following information

- rank of the actions

- give the criteria he wants to base his decision on

- evaluate the action compared to the criterion

Once those information are presented, the UTASTAR algorithm can be executed.

Through out this document, we designated the term value function instead of utility function for the UTA algorithm.

---

[1] Decision Maker

3.2.2   Principles and Notation

Let's call $A = a, b, c, ...$ the set of potential actions evaluated on a consistent family of criteria G $= g_1, g_2, g_3, ..., g_n$. Where $g_i(a)$ represents the evaluation of an action or alternative a on the criterion g_i with $a \in A$.

We define g_{i*} as the least preferred value of a criterion: $g_{i*} = min_{a \in A} g_i(a)$ and g_i^* as the most preferred value of a criterion: $g_i^* = max_{a \in A} g_i(a)$. So the interval for each value for a criterion g_i is: $[g_{i*}, g_i^*]$.

For every $g_i \in G$ and $\forall a, b \in A$ we have the following relations:

$$\begin{cases} a \succ b \Leftrightarrow g_i(a) > g_i(b) & preference \\ a \sim b \Leftrightarrow g_i(a) = g_i(b) & indifference \end{cases} \tag{3.1}$$

The criteria aggregation model in UTASTAR has the following form:

$$v(g(a)) = \sum_{i=1}^{n} v_i(g_i(a)) \tag{3.2}$$

where $v(g(a))$ is the value of an alternative a equal to $v(g(a)) = v(g_1(a), g_2(a), ..., g_n(a))$ subject to normalization constraints:

$$\begin{cases} \sum_{i=1}^{n} v_i(g_i^*) = 1 \\ v_i(g_{i*}) = v_i(g_i^1) = 0, \forall i = 1, 2, ..., n \end{cases} \tag{3.3}$$

where v_i is a marginal value of a criterion, $i = 1, 2, ..., n$ are non-decreasing real valued function. In UTASTAR we have

$$w_{ij} = v_i(g_i^{j+1}) - v_i(g_i^j) \geq 0 \quad \forall i \quad j \tag{3.4}$$

where w_{ij} is a non-negative constraints Which will allow us to write:

$$v_i(g_i^j) = \sum_{t=1}^{j-1} w_{it} \quad \forall i = 1, 2, ..., n \quad and \quad j = 2, 3, ..., \alpha_i - 1 \tag{3.5}$$

With the evaluation of an action a $g(a) = [g_1(a), g_2(a), ..., g_n(a)]$, we have the following relation:

$$\begin{cases} v[g(a)] > v[g(b)] \Leftrightarrow a \succ b \\ v[g(a)] \sim v[g(b)] \Leftrightarrow a = b \end{cases} \tag{3.6}$$

3.2.3 Development

The updated version of UTA, UTASTAR, proposes a double error function for each action: $\sigma^+(a)$ and $\sigma^-(a)$. The value of each alternative $a \in A_R$ can be written:

$$v^{'}[g(a)] = \sum_{i=1}^{n} v_i[g_i(a)] - \sigma^+(a) + \sigma^-(a) \quad \forall a \in A_R \tag{3.7}$$
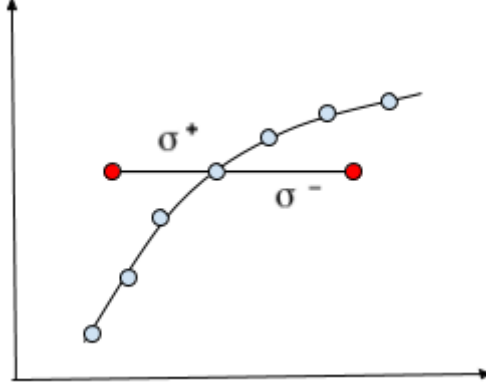

Figure 3.2 − Error function in UTASTAR

For each criteria, the interval $[g_{i*}, g_i^*]$ is cut into $(\alpha_i - 1)$ equal intervals, and the end points g_i^j are given by the formula:

$$g_i^j = g_{i*} + \frac{j-1}{\alpha_i - 1}(g_i^* - g_{i*}) \forall j = 1, 2, ..., \alpha_i \tag{3.8}$$

The marginal value of an action a is calculated by a linear interpolation

$$v_i[g_i(a)] = v_i(g_i^j) + \frac{g_i(a) - g_i^j}{g_i^{j+1} - g_i^j}[v_i(g_i^{j+1}) - v_i(g_i^j)] \tag{3.9}$$

The set of reference actions $A_R = a_1, a_2, ..., a_m$ is arranged by what a_1 is the best action and what a_m is the worst action. Which indicate that we have two possible situations:

- $a_k \succ a_{k+1}$ preference

- $a_k \sim a_{k+1}$ indifference

So if we have that $\Delta(a_k, a_{k+1}) = v^{'}[g(a_k)] - v^{'}[g(a_{k+1})]$ and δ is a small positive number we will obtain the following relations:

$$\begin{cases} \Delta(a_k, a_{k+1}) \geq \delta & where \quad a_k \succ a_{k+1} \\ \Delta(a_k, a_{k+1}) = 0 & where \quad a_k \sim a_{k+1} \end{cases} \tag{3.10}$$

The marginal value functions are estimated by means of the Linear Program with (3.2), (3.3), (3.10) as constraints, and an objective function depending on the σ^+ and σ^-:

$$[min]z = \sum_{k=1}^{m}[\sigma^+(a_k) + \sigma^-(a_k)]$$

subject to:

$$\begin{cases} \Delta(a_k, a_{k+1}) \geq \delta \quad or \quad \Delta(a_k, a_{k+1}) = 0 \\ \sum_{i=1}^{n} \sum_{j=1}^{\alpha_i - 1} w_{ij} = 1 \\ w_{ij} \geq 0, \quad \sigma^+(a_k) \geq 0, \quad \sigma^-(a_k) \geq 0, \quad \forall i, j \, and \, k \end{cases} \tag{3.11}$$

3.2.4   Example - Buying New Car

The implementation of UTASTAR algorithm is illustrated by an example I made previously: **buying a new car**. Another example is available in the Appendices, Choice of transportation, taken from the book: Multiple Criteria Decision Analysis.
The DM is only interested in the following criteria:

- price (in Euro)

- comfort $(0, +, ++, +++)$ *0 being not comfortable and $+++$ very comfortable*

- safety $(1, 2, 3, 4, 5)$ *1 being not safety and 5 safe*

The evaluation of the previous criteria is presented in the following table:

Cars	Price	Comfort	Safety	Ranking of the DM
Nissan Sentra (ns)	17 000	$+++$	4	1
Citroen C4 (c4)	15 000	$++$	2	2
Peugeot 208 GT (p208)	25 000	$+$	3	3
Peugeot 308 berline (p308)	18 500	0	3	4

First of all, we should specify the scale [2] for each criteria.

- Price $\Rightarrow$ $[g_{1*}, g_1^*] = [25\,000, 20\,000, 15\,000]$

- Comfort $\Rightarrow$ $[g_{2*}, g_2^*] = [0, +, ++, + + +]$

- Safety $\Rightarrow$ $[g_{3*}, g_3^*] = [1, 3, 5]$

According to this formula: $v(g(a)) = \sum_{i=1}^{n} v_i(g_i(a))$, the value of each alternative may be written:

- $v(g(ns)) = 0.4v_1(15\,000) + 0.6v_1(20\,000) + v_2(+++) + 0.5v_3(3) + 0.5v_3(5)$

- $v(g(c4)) = v_1(15\,000) + v_2(++) + 0.5v_3(1) + 0.5v_3(3) = v_1(15\,000) + v_2(++) + 0.5v_3(3)$

- $v(g(p208)) = v_1(25\,000) + v_2(+) + v_3(3) = v_2(+) + v_3(3)$

- $v(g(p308)) = 0.3v_1(15\,000) + 0.7v_1(20\,000) + v_2(0) + v_3(3) = 0.3v_1(15\,000) + 0.7v_1(20\,000) + v_3(3)$

We have that $v_1(25\,000) = v_2(0) = v_3(1) = 0$.
Since the marginal value $v_i(g_i)$ can be expressed in terms of variables w_{ij}: $v_i(g_i^j) = \sum_{t=1}^{j-1} w_{it}$, the value of each alternative can be written:

- $v(g(ns)) = w_{11} + 0.4w_{12} + w_{21} + w_{22} + w_{23} + w_{31} + 0.5w_{32}$

- $v(g(c4)) = w_{11} + w_{12} + w_{21} + w_{22} + 0.5w_{31}$

- $v(g(p208)) = w_{21} + w_{31}$

- $v(g(p308)) = w_{11} + 0.3w_{12} + w_{31}$

For each pair of consecutive alternatives, we express the difference between them:

- $\Delta(ns, c4) = -0.6w_{12} + w_{23} + 0.5w_{31} + 0.5w_{32} - \sigma_{ns}^+ + \sigma_{ns}^- + \sigma_{c4}^+ - \sigma_{c4}^-$

- $\Delta(c4, p208) = w_{11} + w_{12} + w_{22} - 0.5w_{31} - \sigma_{c4}^+ + \sigma_{c4}^- + \sigma_{p208}^+ - \sigma_{p208}^-$

- $\Delta(p208, p308) = w_{21} - w_{11} - 0.3w_{12} - \sigma_{p208}^+ + \sigma_{p208}^- + \sigma_{p308}^+ - \sigma_{p308}^-$

---

[2] the interval $[g_{i*}, g_i^*]$ is cut into equal intervals

Having $\delta = 0.05$, we can solve the following LP:

Objective:

$$Minimize \quad \sum_{a \in A} \sigma_a^+ + \sigma_a^- \tag{3.12}$$

Subject to:

$$\begin{cases} -0.6w_{12} + w_{23} + 0.5w_{31} + 0.5w_{32} - \sigma_{ns}^+ + \sigma_{ns}^- + \sigma_{c4}^+ - \sigma_{c4}^- \geq 0.05 \\ w_{11} + w_{12} + w_{22} - 0.5w_{31} - \sigma_{c4}^+ + \sigma_{c4}^- + \sigma_{p208}^+ - \sigma_{p208}^- \geq 0.05 \\ w_{21} - w_{11} - 0.3w_{12} - \sigma_{p208}^+ + \sigma_{p208}^- + \sigma_{p308}^+ - \sigma_{p308}^- \geq 0.05 \\ w_{11} + w_{12} + w_{21} + w_{22} + w_{23} + w_{31} + w_{32} = 1 \end{cases} \tag{3.13}$$

An optimal solution is $w_{12} = 0.34$, $w_{21} = 0.152$, $w_{32} = 0.51$ with $\sum_{a \in A} \sigma_a^+ + \sigma_a^- = 0$. The utilities found for each alternative are as follows:

- $v(g(ns)) = 0.543$

- $v(g(c4)) = 0.492$

- $v(g(p208)) = 0.152$

- $v(g(p308)) = 0.102$

Those utilities are consistent with the DM's preference ranking.

The UTA method build a value function based on the preferences of the DM and consists in solving a linear program (LP) to resolve a multi-criteria problem.
This method will elaborate a model of preferences which is as similar as possible to the DM's preferences.

3.2.5 Value Function

The value function of a problem will allow you to calculate the value of each alternative.
It is composed of several partial value functions, each one represents a criterion with the intervals that represent the coordinates of several points.
So the following partial value function is of the criterion that have the following scale: $[1000, 2000, 3000]$
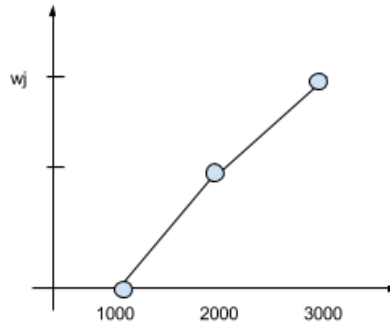

Figure 3.3 − UTA - Partial Value Function

So the sum w_j of all the partial value functions represented in the upper figure is equal to 1.

3.3 Linear Program Solver

One of the steps of the UTA algorithm is solving a Linear Program. To complete the UTA algorithm, I created an independent java application that has the objective of solving the LP by finding the optimal solution.

To achieve this goal, I had to use the google ortools (Optimization tools) library. The library is composed of two external jar files: *com.google.ortools.jar* and *protobuf.jar* and of one native class library *jniortools.dll*. You can find these libraries in the GitHub repository under src/libs.
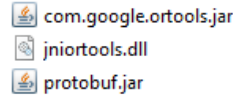

Figure 3.4 − Library used in the solver

Let's consider the following example:

Objective:

$$Maximize \quad 10x_1 + 6x_2 + 4x_3 \tag{3.14}$$

subject to:

$$\begin{cases} x_1 + x_2 + x_3 \leq 100 \\ 10x_1 + 4x_2 + 5x_3 \leq 600 \\ 2x_1 + 2x_2 + 6x_3 \leq 300 \end{cases} \tag{3.15}$$

To solve the LP, we need to create an instance of the solver:

```
MPSolver solver = new MPSolver("LinearProgrammingExample",
    MPSolver.OptimizationProblemType.valueOf("GLOP_LINEAR_PROGRAMMING"));
```

Figure 3.5 − Java code to define the solver

After that, we define the 3 variables x_1, x_2 et x_3 $\quad \in \quad [0; \infty]$:

```
MPVariable x1 = solver.makeNumVar(0.0, infinity, "x1");
MPVariable x2 = solver.makeNumVar(0.0, infinity, "x2");
MPVariable x3 = solver.makeNumVar(0.0, infinity, "x3");
```

Figure 3.6 − Java code to define the variables of the LP

We define the objective $Maximize \quad 10x_1 + 6x_2 + 4x_3$:

```
MPObjective objective = solver.objective();
objective.setCoefficient(x1, 10);
objective.setCoefficient(x2, 6);
objective.setCoefficient(x3, 4);
objective.setMaximization();
```

Figure 3.7 − Java code to define the objective of LP

We do the same for the constraints:

```
MPConstraint c0 = solver.makeConstraint(-infinity, 100.0);
c0.setCoefficient(x1, 1);
c0.setCoefficient(x2, 1);
c0.setCoefficient(x3, 1);

MPConstraint c1 = solver.makeConstraint(-infinity, 600.0);
c1.setCoefficient(x1, 10);
c1.setCoefficient(x2, 4);
c1.setCoefficient(x3, 5);

MPConstraint c2 = solver.makeConstraint(-infinity, 300.0);
c2.setCoefficient(x1, 2);
c2.setCoefficient(x2, 2);
c2.setCoefficient(x3, 6);
```

Figure 3.8 – Java code to define the constraints of the LP

After setting all the constraints and variables we can execute the solver:

```
final MPSolver.ResultStatus resultStatus = solver.solve();
```

Figure 3.9 – Java code for running the solver

After executing the solver, we display the optimal value of the objective and the value of the variables:

```
System.out.println("x1 = " + x1.solutionValue());
System.out.println("x2 = " + x2.solutionValue());
System.out.println("x3 = " + x3.solutionValue());
```

Figure 3.10 – Java code for displaying the result of the solver

Once you run the program, you will have the following the result:

$$Optimal\ objective\ value = 733.3333333333$$
$$x1 = 33.33333333336$$
$$x2 = 66.66666666666$$
$$x3 = 0.0$$

3.4 Implementation - UTA Calculator

We wanted to implement the UTASTAR method. In order to make it possible, we created a java program: uta-calculator. Basically, our goal was to generate a decision problem and run the UTAS-TAR algorithm. Another objective we were trying to achieve was to generate random alternatives and criteria, so we can generate a decision problem.

After implementing the software, we were trying to elaborate a simulation process that generated a random value function v^R and generated a value function v^T from UTASTAR algorithm. The following simulation and interpretation was made under the section *3.5 Research*.

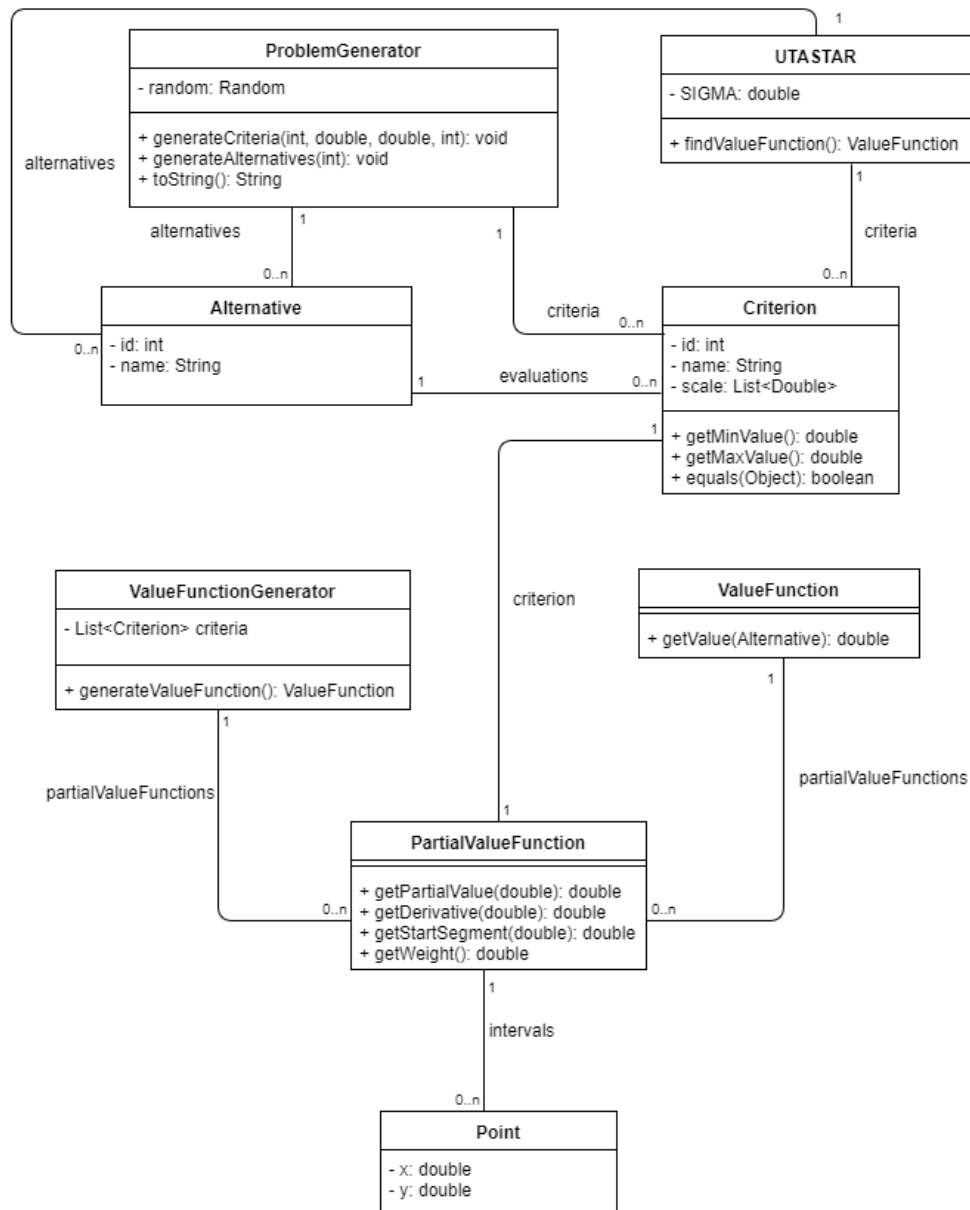### 3.4.1 Architecture

Class Diagram

Figure 3.11 − Class Diagram

Other than the external libraries used in the resolution of a LP, we used the library commons-math from Apache which helped us by calculating the average and standard deviation of the results obtained during the simulation.

Maven

Since we wanted to create an open source project, following the standards conventions was a must which led us to the use of Maven.

We figured out that we need to configure our project in Maven. Maven has been created to make a standard way to build a project and a way to share JAR files through many projects. Figuring out dependencies for small projects is not hard, but once the dependencies expand it will get out of hand. Another plus of using Maven is that it provides a quality project information.

I personally prefer using the IDE Eclipse Luna to code in Java and I am currently using the version Luna 4.4.2 . Using Maven won't be a problem for people who don't use Eclipse, they can use NetBeans or IDEA or any other IDE that supports Maven.

Configuration of Maven:

- groupId: io.github.oliviercailloux
- artifactId: uta-calculator

3.4.2 Functionalities

The following list represents the functionalities of the program I worked on:

- Program your own decision problem
- Generate a random decision problem
- Solve the decision problem by using UTASTAR algorithm
- Elaborate a ValueFunction from a decision problem
- Generate a ValueFunction
- Publishing the application as an open source

3.4.3 Utils

To make some functions more generic, we created a package called utils, where we can find two java classes:

- NumbersGenerator
- ScaleGenerator

NumbersGenerator

The goal of this util class is to generate decimal numbers that have a certain sum. For example, creating three numbers that have the sum of 1.
To make this possible, the class NumbersGenerator follows the method generate that takes the following as parameters:

1. counter (int)
2. targetSum (double)

The parameter counter represents the counter of numbers we want to generate, and the parameter targetSum represents the sum we are targeting.
You can specify the Random value in case you want to have the same result when doing testing.
This method will return a List of number of type double.

ScaleGenerator

The goal of this util class is to generate scale for a criterion, where the interval $[minValue, maxValue]$ is cut into equal intervals. For example, If you want the scale of a criterion that have 10 as minimum value, 20 as maximum value and with four cuts, the function should return the following list: $[10.0, 13.333, 16.667, 20.0]$.
To make this possible, the class ScaleGenerator follows the method generate that takes the following as parameters:

1. minValue (double)

2. maxValue (double)

3. cuts (int)

The parameter minValue represents the minimum value of a criterion, the parameter maxValue represents the maximum value of a criterion and the parameter cuts represents the numbers of cuts into which you want to cut the interval.
This method will return a List of numbers of type double.

3.4.4 Inputs and outputs of the UTASTAR

In order to generate a problem, you have two possibilities:

- Generate a random problem
- Creating your own problem

To **generate a random problem**, you have to specify the following inputs when running the *MainRandomProblem* java class.:

- criterion information
    - numbers of criteria
    - minimum and maximum value of a criterion
    - number of cuts of the scale of a criterion
- numbers of alternatives

By having the following inputs, the software will create the alternatives and criteria and run the UTASTAR algorithm.
To **create your own problem** all you have to do is create a list of criteria and a list of alternatives in the code java, and run the algorithm by giving them parameters. The example of **Buying New Car** is available in the *MainBuyingNewCar* java class.

As output, both possibilities will display the results of the UTASTAR algorithm, value function and the value of each alternative.

3.4.5 Future Improvements

This program has been done during my three months internship. It didn't take three months to make this program, but in order to make such a program I had to do a research and learn about the UTA method. Basically, we had to compromise on some functionalities in order to make the implementation easier. Since the program will be available as an open source on GitHub, it has the potential to grow with some improvements. The following list is some of the improvements that could be made:

1. in the current version of the program, the criterion evaluation has a minimum value and a maximum value, so the evaluation of an action is made in the range $[minValue, maxValue]$ with minValue being the least preferred criterion and maxValue the most preferred. But what if we want to represent a criterion that has maxValue as the least preferred criterion and minValue as the most preferred, like the criterion price, in the current version, this won't be possible.

2. in our program, a criterion has a minimum value and a maximum value, but what if the criterion is not a quantitative item, not evaluated in numbers. For example, let's say we want to evaluate the comfort, we can have the following values: 0, +, ++, +++ or not comfortable, basic, comfortable, very comfortable. So as an improvement we can expect to be able to design all of the criteria possible.

3. the current version doesn't support a graphical interface, a graphical interface can be handy to make the software more user friendly.

3.5   Research

One of the most important aspect of this internship is the research. We started by searching on similar studies made on UTA which helped us to understand more the UTA method. After the implementation of the *uta-calculator* we made a simulation that compared two value functions:

1. v^R a random value function

2. v^T a value function evaluated by the UTA algorithm implemented

In this section, we will talk about the studies found on UTA and then we will talk about the simulation made and the interpretation elaborated.

3.5.1   Comparative analysis of UTA multicriteria methods

The paper Comparative analysis of UTA multicriteria methods, written by Michel Beuthe and Giuseppe Scannella, is mainly about the variants of UTA method. As discussed in this paper a method of multi-criteria analysis is chosen depending on the circumstances of the decision making. UTA makes the estimation of a nonlinear additive function possible, which is obtained using a linear program. The only information required from the decision maker is the global preferences between projects.
After the execution of the algorithm, it is possible to evaluate the preference stated by the decision maker.
Another point stated in the paper is that the initial value of δ must not be given too high. In the basic model, it was noted that the values given to δ were to some extent arbitrary.

Simulations

The simulation is applied to the case of 353 road projects in Belgian Network during the period 1985-2010. The Center of Road Research (CRR) realized a multi criteria analysis that used 29 criteria regrouped in six main themes:

- safety on the present road

- projects socioeconomic aspects

- impact on environment

- current and future traffic

- problems of planning and urbanism

- wear state of the current road.

The 29 criteria have been sorted with a scale going from 1 to 5.

Variants of UTA

1. UTA

2. UTASTAR

3. UTA2

4. UTAMKEN

5. UTAMP1

6. UTAMP2

7. UTAMIME

8. UTASTARMIME

9. UTA2MIME

Conclusions made

When there is no error in the value function estimation, $\sum_{a \in A} \sigma_a^+ + \sigma_a^- = 0$, the basic UTA method provides the most practical and efficient method of estimation.
Even when there is interdependence between criteria, the UTA approach provides good results.
In the case where the value function estimation is positive, $\sum_{a \in A} \sigma_a^+ + \sigma_a^- > 00$, the UTASTAR appears more reliable.
The simulations results indicate that small value of δ lead to better results in case of a positive value function estimation. But in case of no error in the value function estimation, larger value of δ can provide better results. The use of UTAMP1 or UTAMP2 may then be used to find the practical upper bound of the values given.

3.5.2 Disadvantages of the UTA method

1. We can always question the decision marker about his preferences
   The UTA method need the alternatives and action list given with their preferences. The preferences should be given by the DM, so if he made a mistake or a wrong estimation of the preference the linear program can have a wrong constraint which can affect the final results.

2. Solution may not be the only one as in any LP we can have multiple solutions
   A step of the UTA is solving a linear program and as we all know linear program can have different solutions each time we run the LP.

3.5.3 Experimentation about the variants of UTA

An experimentation was made by J. SISKOS and/c D. YANNACOPOULOUS and was written in the paper: *Amélioration de la méthode UTA par introduction d'une double fonction d'erreur* which is the *Cahier du LAMSADE n49*. The goal of this experimentation was to show the advantages offered by the newer version of UTA compared to the older version. Ten actions have been evaluated randomly over six criteria with a scale of $[1, 4]$ where four is the most preferred value.

In order to facilitate the comparison, two factors have been used:

1. Iterations necessary to obtain the optimal value

2. Objective of the LP: Minimization of $\sum_{a \in A} \sigma_a^+ + \sigma_a^-$

The first factor was better by 48% than the older version. The second factor was better by 36% than the older version. Comparison table:

Factors	UTA	UTASTAR	Result
Iterations	21.5	11.1	drop of 48%
Objective of LP	0.34	0.53	drop of 36%

3.5.4   Simulation and Comparison uta-calcultor

The following represents the main objective of my research: simulation of the implemented software uta-calculator.

Simulation

We generated a value function called v^R with a certain number of criteria, then we ordered a list of $10,000$ alternatives by the value function generated.
Then we took a sub list of the alternatives ordered and we called it *exampleAlternatives* (the number of examples will vary during the simulation). And for the rest of the alternatives we placed them in another list called *alternativesToCompare*.
We generated the value function v^T by running the UTASTAR algorithm on the list *exampleAlternatives*.
Then we took *alternativesToCompare* and ordered them according to the two value functions v^R and v^T and we compared the difference of ranking between the two list corresponding to each value function.
We repeated the upper process 500 times and then we calculated the mean of the Kendall rank correlation coefficient of the difference of ranking. The Kendall's tau coefficient of the difference of ranking is a statistic method used to measure the ordinal association between two list, the tau correlation returns a value between -1 and 1. The Kendall will be high when observations have a similar, or identical when the correlation is equal to 1. The Kendall will be low when observations have a dissimilar rank, or fully different when the correlation is equal to -1.
This following simulation was repeated by varying the number of criteria between $1, 2, 3, 4, 5$ and by varying the number of alternatives as example (given to UTASTAR which generate v^T) by $10, 25, 50, 75, 100$.
The following results were obtained after the simulation:

1 criterion	
Examples of alternatives to generate v^T	Mean of the Kendall rank correlation coefficient
10	1.0
25	1.0
50	1.0
75	1.0
100	1.0

Figure 3.12 − Table - Simulation results of Kendall rank correlation coefficient with 1 criterion

2 criterion	
Examples of alternatives to generate v^T	Mean of the Kendall rank correlation coefficient
10	0.530
25	0.531
50	0.546
75	0.537
100	0.551

Figure 3.13 − Table - Simulation results of Kendall rank correlation coefficient with 2 criterion

3 criterion	
Examples of alternatives to generate v^T	Mean of the Kendall rank correlation coefficient
10	0.409
25	0.421
50	0.473
75	0.452
100	0.474

Figure 3.14 − Table - Simulation results of Kendall rank correlation coefficient with 3 criterion

4 criterion	
Examples of alternatives to generate v^T	Mean of the Kendall rank correlation coefficient
10	0.296
25	0.337
50	0.362
75	0.403
100	0.405

Figure 3.15 − Table - Simulation results of Kendall rank correlation coefficient with 4 criterion

5 criterion	
Examples of alternatives to generate v^T	Mean of the Kendall rank correlation coefficient
10	0.220
25	0.262
50	0.293
75	0.308
100	0.330

Figure 3.16 − Table - Simulation results of Kendall rank correlation coefficient with 5 criterion

Interpretation

- The following graph represents the simulation results where we evaluated $10,000$ alternatives on 1 criterion by varying the number of alternatives as example by $[10, 25, 50, 75, 100]$
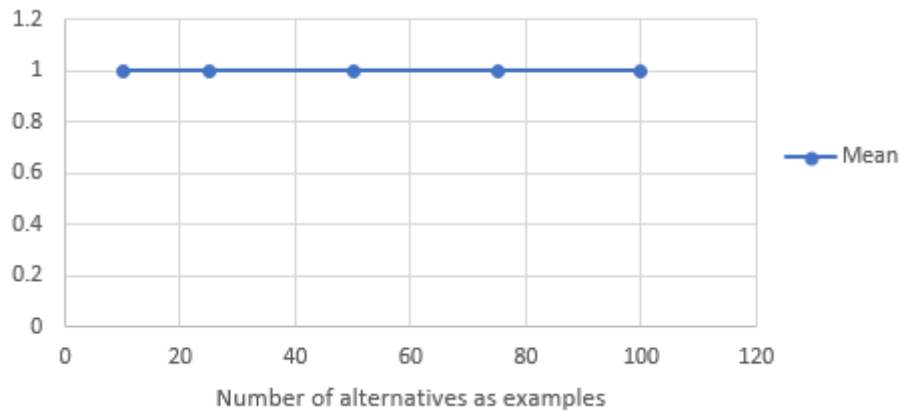

Figure 3.17 − Graph - Simulation results with 1 criterion

In the case where we have 1 criterion, the two value functions generated two list that have

identical ranking.

- The following graph represents the simulation results where we evaluated $10,000$ alternatives by taking 200 alternatives as examples that generate v^T and by varying the number of criteria $[1, 2, 3, 4, 5]$
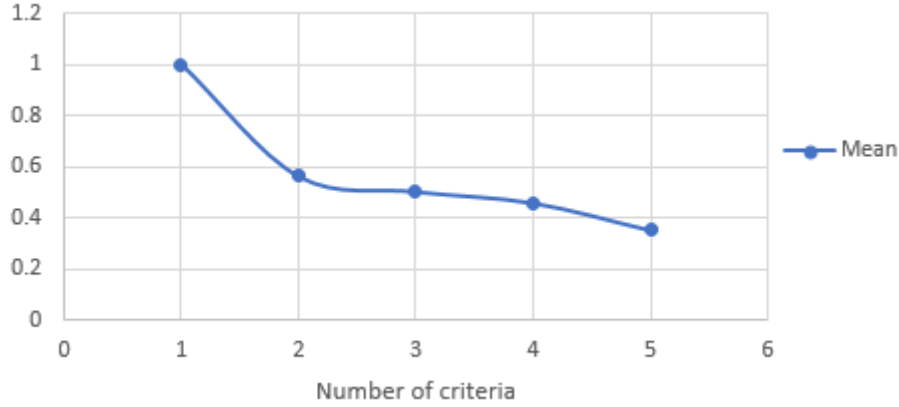

Figure 3.18 – Graph - Simulation results by varying the number of criteria

The more we added a criterion the more the differences grew between the two lists, which explain the decrease of the Kendall rank correlation coefficient.

- The following graph represents the simulation results where we evaluated $10,000$ alternatives on 2 criteria by varying the number of alternatives as example that generate v^T by $[10, 25, 50, 75, 100, 200, 500, 1000]$
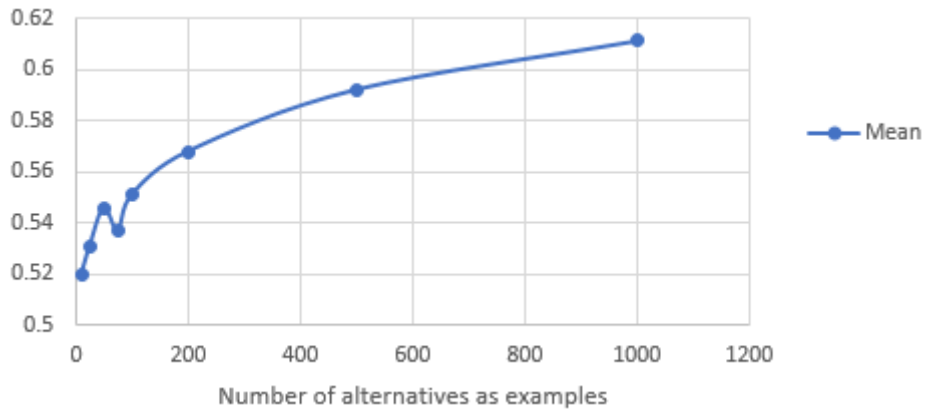

Figure 3.19 – Graph - Simulation results by varying the number of alternatives as example with 2 criterion

The more the number of alternatives as example increased, the more the value function v^T rank the list as similar as possible as the one generated by v^R. Which explain why more the numbers of example increase the more Kendall tau get close to 1.

- The following graph represents the simulation results where we evaluated $10,000$ alternatives on 5 criteria by varying the number of alternatives as example by $[10, 25, 50, 75, 100]$
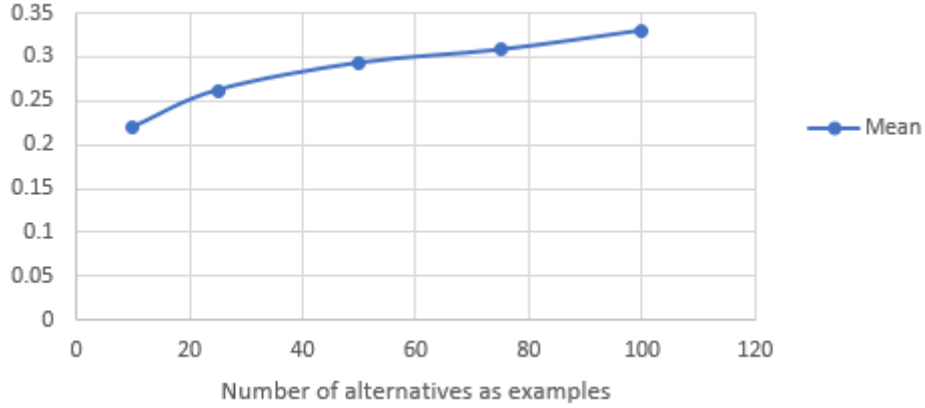

Figure 3.20 – Graph - Simulation results by varying the number of alternatives as example

The more the *exampleList* augmented the more the difference between the two list minimized and the Kendall rank correlation coefficient increase this is due to the fact that the v^T will have more alternatives to generate a more similar value function of v^R.

3.6 Conclusion on the projects

During the internship, a lot of tasks has been accomplished. First, I have been able to learn about the platform MyDraft, to make a tutorial and to realize a small application. Secondly, I was able to learn about a new decision method, UTA, where I made a summary about it, and created my own problem, Buying New Car. Before implementing the UTASTAR method, a research was made. The goal of the research was to find similar studies made on UTA method. And finally, I was able to implement the UTASTAR method by the software *uta-calculator*. After implementing the software I was able to make a simulation and to elaborate an interpretation of the results obtained.

During the implementation, we used some external libraries that were given by Mr. Cailloux. The libraries provided by my supervisor were handy and very lightweight. We published the work done and the source code on GitHub, I believe that this was a great choice since we were aiming to provide an open source software and there is no better way than GitHub. The code was done in Java which helped us to find some external libraries.

Chapter 4

Conclusion

During my three months internship, the main goal was to implement an open source software about the UTA method which is an ordinal regression technique, proposed by E. Jacquete-Lagreze and J. Siskos, that adjusts a system of additive utility function. Under the supervision and the help of Mr. Olivier Cailloux, I created this software and published it to GitHub under the MIT license. The implementation of the software was followed by a research made on the simulation of the software.

4.1   Personal benefits

During my internship, I have learned a lot, put in practice many of the subjects learned during my academic years, so here the list of my new acquired skills:

- Learning to write documents using LaTeX which helped with the appearance of documents to focus more on the content.

- Learning a new Decision method, the UTA method

- Re-doing my project of Portfolio Management was helpful for me since next year I will be completing my master's degree in Information System for finance

- Learning how to make research of literature on a certain subject

- Making projects using Maven

- Publishing a software on GitHub as open source

- Practicing my English skill

4.2   Difficulties encountered

I didn't have many difficulties, the only main problem I faced was the fact that I was working alone. Working alone and not surrounded by colleagues can be a little set back, since you cannot interact with them and ask questions on the spot. I had to wait for the next meeting or send an email in case I had any inquiries or problems.

The internship has been very rewarding for me on several levels, being able to practice several subjects taken during my academic years like the course *Decision*, *Marché Financier* and *Advanced Java*. Due to this internship, I learned how to be autonomous and how to learn to manage my own time to finish the work.
To sum-up, these three months internship were a great experience on a professional level and a personal level.

Bibliography

[1] GRECO S., EHRGOTT M., RUI FIGUEIRA J., 2004, *Multiple Criteria Decision Analysis*

[2] SISKOS J., 1979, *Cahier du LAMSADE n24 : Les programmes UTA*

[3] SISKOS J., YANNACOPOULOS D., 1982, *Cahier du LAMSADE n49 : Amélioration de la méthode UTA par introduction d'une double fonction d'erreurs*

[4] HAMMAMI A., 2003, *Thèse : Modélisation technico-économique d'une chaîne logistique dans une entreprise reseau*

[5] LUO H., SUN Z., 2014, *A study on Stock Ranking and Selection Strategy Based on UTA Method under the Condition of Inconsistence*

[6] ZOPOUNIDIS C., DOUMPOS M., 1997, *A multicriteria decision aid methodology for the assessment of country risk*

[7] EHSANIFAR M., ESHLAGHI A., KERAMATI M., NAZEMI J., 2013, *The Development of UTASTAR Method in Fuzzy Environment for Supplier Selection*

[8] SISKOS J., 1983, *Application de la méthode UTA à un problème de sélection de points de vente mettant en jeu des critères multiples*

List of Figures

Appendices

Annexe A

Compte Rendus avec M. Cailloux

A.1   CR1 - 15 juin 2017

15 juin 2017 / 15 :00 / Université Paris Dauphine

<u>Participants</u>

Olivier Cailloux, Elie Daher

<u>Déroulement de la réunion</u>

- Explication du plan du stage
- J'ai présenté les tâches effectuées
- J'avais quelques questions sur la méthode UTA et plus précisément sur un exemple "A Numerical Example" du chapitre 9 : "UTA Method" du livre : "Multiple Criteria Decision Analysis"
- Présentation des tâches à effectuer

<u>Les tâches à faire</u>

- Créer un repository sur GitHub
- Ecrire un résumé sur la méthode UTA en utilisant LaTeX
- En parallèle, effectuer des recherches sur des littératures existantes

<u>Pièces jointes</u>

- First iteration : document présenté par M. Cailloux qui contient le plan de la première itération

La réunion a été levée à 15 :30

La prochaine réunion est prévue pour le mercredi 21 juin 2017 à 14 :00

A.2 CR2 - 21 juin 2017

21 juin 2017 / 14 :00 / Université Paris Dauphine

Participants

Olivier Cailloux, Elie Daher

Déroulement de la réunion

- Présentation du travail effectué
- Résolution du projet java sur le programme linéaire

Les tâches à faire

- Rendre le document plus détaillé et développé
- Rendre le document plus compréhensible pour les lecteurs qui n'ont pas un context décisionnel
- Fixer le programme linear programming sur Java

La réunion a été levée à 14 :40

La prochaine réunion est prévue pour le mercredi 28 juin 2017 à 11 :00

A.3 CR3 - 28 juin 2017

28 juin 2017 / 11 :00 / Université Paris Dauphine

<u>Participants</u>

Olivier Cailloux, Elie Daher

<u>Déroulement de la réunion</u>

- Présentation du travail effectué

- Citer les remarques et les modifications à réaliser sur le document summary-uta et sur le repository de GitHub (enlever le folder qui contient les *.class)

- Présentation du travail à effectuer pour la semaine prochaine

<u>Les tâches à faire</u>

- Corriger les remarques sur le document

- Développer v^R et v^T

- Effectuer une recherche sur les littératures similaires déjà réalisées

La réunion a été levée à 11 :30

La prochaine réunion est prévue pour le mercredi 5 juillet 2017 à 10 :00

A.4 CR4 - 18 juillet 2017

18 juillet 2017 / 10 :00 / Université Paris Dauphine

<u>Participants</u>

Olivier Cailloux, Elie Daher

<u>Déroulement de la réunion</u>

- Présentation du travail effectué
- Explication de la fusion entre le projet Alternative-Criteria et UTA
- Apprendre comment télécharger des littératures à partir du site de la BU de Dauphine

<u>Les tâches à faire</u>

- Elaborer un document qui regroupe tout le travail effectué
- S'approfondir dans la recherche des littératures
- Fixer la méthode GenerateNumbers

La réunion a été levée à 10 :45

La prochaine réunion est prévue pour le mardi 25 juillet 2017 à 10 :00

A.5 CR5 - 25 juillet 2017

25 juillet 2017 / 10 :00 / Université Paris Dauphine

Participants

Olivier Cailloux, Elie Daher

Déroulement de la réunion

- Présentation du travail effectué
- Explication des modifications à faire dans le programme UTA
- Explication des modification à effectuer dans le document report

Les tâches à faire

- Ajouter les annexes des littérature dans le document
- Définir l'objectif de la méthode UTA dans le document
- Etablir les modifications sur le programme UTA
- Elaborer des résumés pour les articles trouvés

La réunion a été levée à 11 :15

La prochaine réunion est prévue pour le lundi 31 juillet 2017 à 10 :00

A.6   CR6 - 31 juillet 2017

31 juillet 2017 / 10 :00 / Café Lumière - Hôtel Scribe

Participants

Olivier Cailloux, Elie Daher

Déroulement de la réunion

- Présentation du travail effectué
- Refaire l'architecture du programme
- Présenter les modifications à faire

Les tâches à faire

- Elaborer des petits résumés pour les articles trouvés
- Compléter le rapport final
- Effectuer les modifications sur le programme

La réunion a été levée à 11 :15

A.7 CR7 - 16 août 2017

16 août 2017 / 10 :00 / Café Mocate - Opera

<u>Participants</u>

Olivier Cailloux, Elie Daher

<u>Déroulement de la réunion</u>

- Présentation du travail effectué
- Remarques sur le rapport
- Explication des étapes à faire dans la partie simulation

<u>Les tâches à faire</u>

- Commencer la simulation du programme uta-calculator
- Finaliser le rapport

La réunion a été levée à 11 :00

La prochaine réunion est prévue pour le mardi 22 août 2017 à 10 :00

A.8 CR8 - 22 août 2017

22 août 2017 / 10 :00 / Université Paris Dauphine

<u>Participants</u>

Olivier Cailloux, Elie Daher

<u>Déroulement de la réunion</u>

- Présentation du travail effectué
- Discussion sur la partie simulation

<u>Les tâches à faire</u>

- Commencer à préparer pour la présentation - soutenance du 4 septembre
- Compléter la partie simulation

La réunion a été levée à 11 :00

La prochaine réunion est prévue pour le jeudi 28 août 2017 à 10 :00

Annexe B

Example of UTASTAR - Analyzing the choice of transportation

A DM wants to analyse the choice of transportation. The DM is interstered in the following criteria

1. price

2. time (min)

3. comfort (possibility to have a seat)

The evaluation of the previous criteria is presented in the following table :

Means of transportation	Price	Time	Comfort	Ranking of the DM
RER	3	10	$+$	1
METRO (1)	4	20	$++$	2
METRO (2)	2	20	0	2
BUS	6	40	0	3
TAXI	30	30	$+++$	4

DM's preferences : $RER \succ Metro1 \approx Metro2 \succ Bus \succ Taxi$

First of all, we should specify the scale [1] for each criteria.

- Price $\rightarrow$ $[30, 16, 2]$

- Time $\rightarrow$ $[40, 30, 20, 10]$

- Comfort $\rightarrow$ $[0, +, ++, +++]$

According to this formula : $v(g(a)) = \sum_{i=1}^{n} v_i(g_i(a))$, the value of each alternative may be written :

- $v[g(RER)] = 0.07v_1(16) + 0.93v_1(2) + v_2(10) + v_3(+)$

- $v[g(METRO1)] = 0.14v_1(16) + 0.86v_1(2) + v_2(20) + v_3(++)$

- $v[g(METRO2)] = v_1(2) + v_2(20) + v_3(0) = v_1(2) + v_2(20)$

- $v[g(BUS)] = 0.29v_1(16) + 0.71v_1(2) + v_2(40) + v_3(0) = 0.29v_1(16) + 0.71v_1(2)$

- $v[g(TAXI)] = v_1(30) + v_2(30) + v_3(+++) = v_2(30) + v_3(+++)$

We have that $v_1(30) = v_2(40) = v_3(0) = 0$.

Since the marginal value $v_i(g_i)$ can be expressed in terms of variables w_{ij} : $v_i(g_i^j) = \sum_{t=1}^{j-1} w_{it}$, the value of each alternative can be written :

- $v[g(RER)] = w_{11} + 0.93w_{12} + w_{21} + w_{22} + w_{23} + w_{31}$

- $v[g(METRO1)] = w_{11} + 0.86w_{12} + w_{21} + w_{22} + w_{31} + w_{32}$

- $v[g(METRO2)] = w_{11} + w_{12} + w_{21} + w_{22}$

- $v[g(BUS)] = w_{11} + 0.71w_{12}$

- $v[g(TAXI)] = w_{21} + w_{31} + w_{32} + w_{33}$

For each pair of consecutive alternatives, we express the difference between them :

- $\Delta(RER, METRO1) = 0.07w_{12} + w_{23} - w_{32} \geq \delta$

- $\Delta(METRO1, METRO2) = -0.14w_{12} + w_{31} + w_{32} = 0$

- $\Delta(METRO2, BUS) = 0.29w_{12} + w_{21} + w_{22} \geq \delta$

- $\Delta(BUS, TAXI) = w_{11} + 0.71w_{12} - w_{21} - w_{31} - w_{32} - w_{33} \geq \delta$

Having $\delta = 0.05$, we can solve the following LP :

Objective :

$$Minimize \quad \sum_{a \in A} \sigma_a^+ + \sigma_a^- \tag{B.1}$$

Subject to :

$$
\begin{cases}
0.07w_{12} + w_{23} - w_{32} - \sigma_{RER}^+ + \sigma_{RER}^- + \sigma_{METRO1}^+ - \sigma_{METRO1}^- \geq \delta \\
-0.14w_{12} + w_{31} + w_{32} - \sigma_{METRO1}^+ + \sigma_{METRO1}^- + \sigma_{METRO2}^+ - \sigma_{METRO2}^- = 0 \\
0.29w_{12} + w_{21} + w_{22} - \sigma_{METRO2}^+ + \sigma_{METRO2}^- + \sigma_{BUS}^+ - \sigma_{BUS}^- \geq \delta \\
w_{11} + 0.71w_{12} - w_{21} - w_{31} - w_{32} - w_{33} - \sigma_{BUS}^+ + \sigma_{BUS}^- + \sigma_{TAXI}^+ - \sigma_{TAXI}^- \geq \delta \\
w_{11} + w_{12} + w_{21} + w_{22} + w_{23} + w_{31} + w_{32} + w_{33} = 1
\end{cases} \tag{B.2}
$$

So by using the com.google.ortools library, we can solve the Linear Program above with $\sigma = 0.05$. This Linear Program solution is coded in Java class ChoiceTransportation.

By executing the class ChoiceTransportation, you will have the following result :

An optimal solution is $w_{11} = 0.5$, $w_{22} = 0.05$, $w_{23} = 0.05$, $w_{33} = 0.4$ with $\sum_{a \in A} \sigma_a^+ + \sigma_a^- = 0$. The utilities found for each alternative are as follows :

- $v(g(RER)) = 0.6$

---

[1] the interval $[g_{i*}, g_i^*]$ is cut into equal intervals

- $v(g(METRO1)) = 0.55$

- $v(g(METRO2)) = 0.55$

- $v(g(BUS)) = 0.5$

- $v(g(TAXI)) = 0.4$

Those utilities are consistent with the DM's preference ranking.

Annexe C

Literature found during research

C.1 Literature used in the documentation

- Multiple Criteria Decision Analysis - Salvatore Greco, Matthias Ehrgott, Jose Rui Figueira, 2004

- Cahier du LAMSADE n24 : Les programmes UTA - J. Siskos, Octobre 1979

- Cahier du LAMSADE n49 : Amélioration de la méthode UTA par introduction d'une double fonction d'erreurs - J.Siskos, D. Yannacopoulos, octobre 1983

- Thèse : Modélisation technico-économique d'une chaîne logistique dans une entreprise reseau - Abdelkader Hammami, 2003

- Assessing a set of additive utility functions for multicriteria decision-making, the UTA method

- Assessing non-additive utility for multicriteria decision aid

- Comparative analysis of UTA multicriteria methods

- Preference disaggregation : 20 years of MCDA experience

C.2 Useful literature

- On the expressiveness of the additive value function and the Choquet integral models

- ACUTA : A novel method for eliciting additive value functions on the basis of holistic preference statements

- Stewart - Robustness of Additive Value Function Methods in MCDM (1996)

- La Conception et l'implementation d'un outil d'aide a la decision multicriteres integrant les concepts de la gestion des connaissances et du cycle de vie : application de la chaine d'approvisionnement humanitaire

- A study on Stock Ranking and Selection Strategy Based on UTA Method under the Condition of Inconsistence - Hong-chen Luo, Zhao-xu Sun, aout 2014

- The Development of UTASTAR Method in Fuzzy Environment for Supplier Selection

- A multicriteria decision aid methodology for the assessment of country risk

- Application de la méthode UTA à un problème de sélection de points de vente mettant en jeu des critères multiples