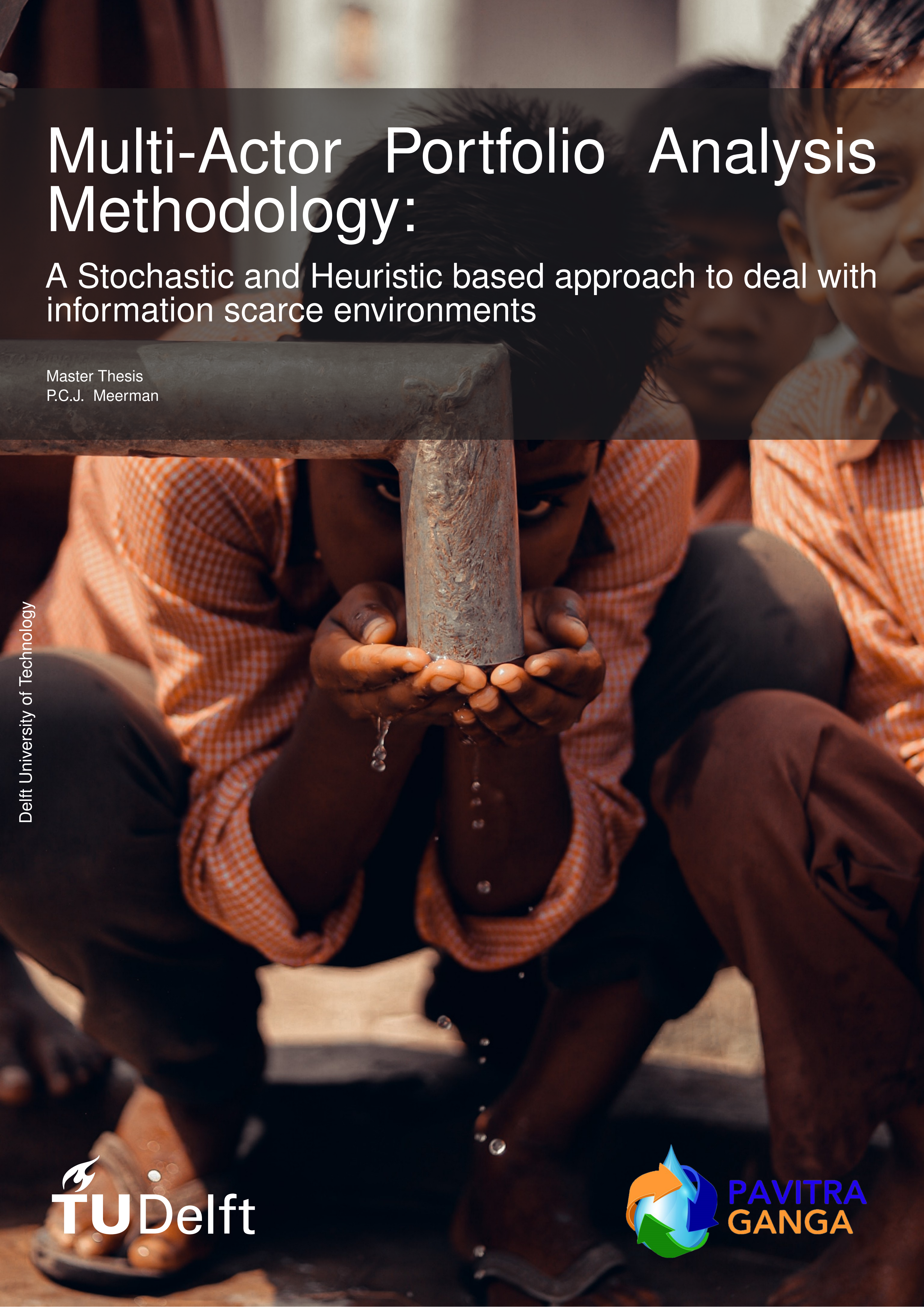


Multi-Actor Portfolio Analysis Methodology:

A Stochastic and Heuristic based approach to deal with information scarce environments

Master Thesis

P.C.J. Meerman

Delft University of Technology

TUDelft

PAVITRA GANGA

Multi-Actor Portfolio Analysis Methodology:

A Stochastic and Heuristic based approach to deal with information scarce environments

Master thesis submitted to Delft University of Technology
in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE
in **Engineering and Policy Analysis**
Faculty of Technology, Policy and Management

by

P.C.J. Meerman

To be defended in public on April 25th 2022

Student number:	4383591	
Project duration:	November 1, 2021 – April 25, 2022	
Graduation committee:	Dr. L. Scholten,	TU Delft, First Supervisor, Section Policy Analysis
	Dr. J. Rezaei,	TU Delft, Chairperson and Second Supervisor, Section Transport and Logistics
	Dr. T. Saharan,	TU Delft, External Supervisor, Pavitra Ganga

Cover Image: Link

TU Delft

Abstract

Previous academic work on Portfolio Decision Analysis (PDA) have pointed at the advantage of using PDA in multi-actor situations and environmental sectors, but have also pointed at the lack of research done in such case studies. Currently there exist situations where PDA would be beneficial but current methods are not able to analyse the systems. This is because most of the PDA and MCDA methods require certain data that is often absent in real life case studies, such as reliable information on alternatives performances including synergies and the availability of time and willingness for detailed preference elicitation. Multi-actor situations make preference elicitation an even more time consuming aspect. Therefore, this thesis develops a novel methodology making PDA analysis accessible, robust and fast in decision support to multi-actor systems lacking detailed synergy and preference information. First the portfolios are created in a linear-additivity framework covering the solution space (creating thus all feasible portfolios). The methodology uses different strains of thought and concepts within the MCDA community, such as a combination of stochastic and heuristic approaches to analyse the portfolios. The stochastic techniques include Monte Carlo sampling for uncertainty propagation of the alternative performances and stochastic multicriteria acceptability analysis (SMAA) to handle attribute weight uncertainties. The heuristics applied stand within the Fast and Frugal Heuristics tradition (FFH). The studied heuristics are the Take-The-Best Heuristic and a Lexicographic heuristic to incorporate synergy information. Application results in portfolio shortlists for the involved actors. In order to analyze consensus among the actors a Core Indices (CI) approach is applied. Subsequently a Sobol sensitivity analysis is conducted on the most promising portfolios. The integration of these methods result in an overarching unconventional approach capable of informing decision makers on most promising portfolios and most important attributes. An application of this methodology is shown through application to both simulated data and a real life case study in a multi-actor case-study in India, on wastewater treatment, reuse and resource recovery. The case study encompasses multiple stakeholders in which, due to the political nature of the situation, it is difficult to obtain explicit preference. The alternatives also likely have interactions and interdependencies, affecting the portfolio set performance. The application shows that the method is able to deal with the complexities of real life case studies. For both the simulated data set as for the case study consensus on portfolios, alternatives and corresponding attributes have identified for further decision-making. Regarding the analysis of the attribute weights uncertainty the SMAA-O approach is deemed better than the Take-The-Best approach if a complete ordinal ranking of the attributes is available. Furthermore, the approach of lexicographic heuristics to deal with the lack of detailed synergy information seems promising for situations where synergy information is lacking. For the general application of the method there are, however, some limitations due to assumed linear properties, these encompass: The assumed linear additivity of the alternatives in the portfolio creation, the monotonic behaviour of the marginal value functions and the hierarchical addition aggregation value function. These form also the points where the method can be expanded upon to integrate these assumptions and corresponding uncertainties.

In the future the method can be expanded upon in the open source software to make further use of the computing power able to perform stochastic analysis and to build upon the novel application of heuristics, quickly allowing PDA analysis in ever more environmental case studies.

Keywords: **Decision-making, Portfolio Decision Analysis (PDA), Stakeholder Analysis, Multi-Criteria Decision Analysis (MCDA), Heuristics**

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
CETP	Common effluent treatment plants
CPCB	Central Pollution Control Board
Cr	Chromium
Crore	10.000.000
CI	Consensus Index
DM	Decision-maker
EPA	Engineering and Policy Analysis
FFH	Fast and frugal heuristics
INGO	International non-governmental organization
Lakh	100.000
NGT	National Green Tribunal
MAVT	Multi-Attribute Value Theory
MCDA	Multi-Criteria Decision Analysis
MLD	Millions Liter a Day
PDA	Portfolio Decision Analysis
PETP	Primary Effluent Treatment Plant
PG	Pavitra Ganga
SMAA	Stochastic multicriteria acceptability analysis
STP	Sewage Treatment Plant
URMP	Urban River Management Plan
UP	Uttar Pradesh (Indian State)
WHO	World Health Organisation
WWT	Waste-Water Treatment
YODA	Your Own Decision Aid

Symbols

Symbol	Definition	Unit
BOD	Biochemical Oxygen Demand	$[g_{O_2}/g]$
COD	Chemical Oxygen Demand	$[g_{O_2}/g]$
TSS	Total Suspended Solids	[mg/L]
PE	Population Equivalent	[inhabitants]
ρ	Density	[kg/m^3]

List of Figures

List of Tables

1
Introduction

For even small decisions in life one often has to balance things: What is the fastest way to go to work today, what new smartphone contract do I need, what master thesis topic shall I study? Most of these decisions are made by weighing unconsciously or consciously different options and criteria. Similar decisions have to be made for larger questions, on road infrastructure, environmental protection, financial portfolios, wastewater treatment for example. Such decisions are often much more important and affect other people as well. Many problems are so complex that the human mind cannot fathom and weigh all aspects of options accurately. That is why several approaches have been developed to help, to assist others, the decision-makers (DM), with making such decisions. One such important umbrella of tools is Multicriteria Decision Making (MCDM) or Multicriteria Decision Analysis (MCDA). MCDA's strive to help by listing clearly all the options and criteria and explore those options on the different criteria a DM has. (Belton & Stewart, 2002). In this thesis these options are called *alternatives*, and the criteria are called *attributes*. There are also situations where there is not just one choice or decision that has to be made, but multiple at once; a so-called portfolio has to be chosen. For example the selection of infrastructure projects in a country or the range of products a store wants to sell; Assisting in such situations could than be called a Portfolio Decision Analysis (PDA).

There are some challenges, however, with the use of the standard MCDA/PDA approaches in real life. This thesis is going to explore, elaborate and address some of those challenges, such as the data requirements for an analysis and the uncertainties regarding the data. Firstly, in this chapter a brief overview is given of the MCDA and PDA concepts and approaches. Subsequently, the state-of-the-art literature is reviewed of these methods on the matter of multi-actor situations and alternative interactions. From this the research gaps are distilled and the main research and sub-questions are presented. Subsequently the Methods chapter outlines developed methodology. After this a real life multi-actor case study will be presented. The Results chapter shows the generated results from a set of simulated data and the case study. The Discussion chapter consequently reviews, the assumptions, results and limitations of the methodology. In the end the Conclusion chapter summarizes the findings on the research questions and the case study and finishes with point for further research.

1.1. Concepts

1.1.1. MCDA

Multi-Criteria Decision Analysis (MCDA) is a formalised and structured body of theory, and part of the developments in the Operation Research and Decision Analysis fields (Belton & Stewart, 2002). Especially, from the 1960's the methods became formalised and applied in many fields where multiple attributes had to be balanced in both private and public decisions (Baumol, 1961; Figueira et al., 2005; Keeney, 1982). For an overview of the development of MCDA I refer to Köksalan et al., 2013.

To deal with a certain problem, or achieve a certain goal, there often are multiple solutions possible, the alternatives. The first phase in which all these alternatives, attributes and problem-definitions are obtained is a different matter all-together and another body of methods (Problem Structuring Methods) and is outside the scope of this thesis (Marttunen et al., 2017; Rosenhead and Mingers, 2001). Subsequently, the matrix of outcomes of these alternatives is constructed as shown in Table 1.1. This matrix forms the core of the MCDA and Multi-Criteria PDA;

The alternatives are denoted as A_i for alternative i, and is judged on C_j attribute for attribute j. The outcomes, x_{ij}, for each alternative on each attribute is noted down. In the end one usually gets a table as 1.1 as is shown below. The attributes can have different levels of importance to the decision-makers. It now becomes necessary to construct a model to represent the DMs values and preferences; There are many methods to do so. Sometimes, for example, weights are given to the attributes and in other methods the attributes are ranked. In the core there is usually a way to represent the importance of the criteria and a way to compare the attributes. The attributes, alternatives, preferences and outcomes of the matrix are often elicited (determined) in close corporation with the decision-makers (DM) and experts.

Table 1.1: Matrix of alternative and attributes performances

	C_1	C_2	C_3	C_j
A_1	x_{11}	x_{12}	x_{13}	x_{1j}
A_2	x_{21}	x_{22}	x_{23}	x_{2j}
A_3	x_{31}	x_{32}	x_{33}	x_{3j}
A_i	x_{i1}	x_{i2}	x_{i3}	x_{ij}

The exact representation of the outcomes, x_{ij}, can be done in different ways: from simple numbers, to adjusted numbers based on the value function of a DM, to itself being a continuous function. You can make this representation as complex as you want, however the goal of the MCDA should be kept in mind; more complexity is not always better.

The different MCDA approaches can be categorized in 3 types based on the typology in *Multiple criteria decision analysis: an integrated approach*. (Belton and Stewart, 2002, pg.9):

- Value measurement models
- Outranking models
- Heuristics

Value measurement models

One of the ways the decision matrix can be filled are with the Multi-Attribute Value Theory (MAVT) approaches. These methods have an established tradition of use in the sector, therefore much literature can be found on it and certain (open) software applications already exist (Chacon-Hurtado and Scholten, 2021). The preference elicitation and inter-comparability/aggregation of attributes is done by the modelling of value-functions. These are obtained by extensive conversation, elicitation, with the decision-maker to obtain the 'marginal' value function for each *attribtute*. The outcomes of each alternative on the corresponding attribute give a certain 'value' for this attribute based on the value function. These can then be aggregated to obtain the overall value function.

A benefit of this method is the comparability over different attributes because the outcomes are expressed in values; weight attributes are easily incorporated in the outcome calculation. Furthermore, it is easy to add or remove certain attribute to matrix without having to do much recalculating. This is the MCDA approach used in this study.

Outranking models

Next to the umbrella of value-functions there are the Outranking methods, such as PROMETHEE and Electre. In contrast to MAVT approaches they do not create value functions for the attribute and do not aggregate these over the different attributes. Some attributes are seen as 'incomparable' and therefore this aggregation cannot be done say the proponents of these models (Roy, 1971). The alternatives are to be ranked for the separate attributes and this ranking is assumed to be independent from other attributes. The analyst using this method tries not to determine the strength of preference as such, but the relative preference between two alternatives on each attribute. Then the alternatives are ranked.

Heuristics
Heuristics are strategies or rules by which choices are made. Within the domain of MCDA it is sensible to make a differentiation in the application of heuristics for two different purposes. The first being in computer science and the second in behavioural psychology of choice (Durbach et al., 2020). In the first domain heuristics are applied to limit the amount of computation time required to solve a calculation. Here we refer to the second, where it refers to adaptive decision strategies to make choices, known as 'fast and frugal heuristics' (FFH) (Artinger et al., 2015).

Similar to the outranking models the analysis of the decision matrix is carried out on the partial information considered by the decision-makers; Alternatives are thus analysed for each attribute. The heuristic searches the problem space and will arrive at a solution when certain conditions are met. So, they are different from optimization strategies in that they often ignore information and do not try to optimize. Many such rules can be constructed and have been constructed to resemble how humans tend to make decisions or to see the performance of certain heuristics, even when humans tend not to use it. Its basis can be found in the work Simon and Newall and theories of bounded rationality (Simon, 1955). An example of such fast and frugal heuristic is the *Satisficing* heuristic. It starts searching the problem space until a satisfying solution is found and then it stops and selects that solution. This strategy does not try to maximize the solution and is used as an important strategy in decision-making (Simon, 1955). The first benefit of applying heuristics correctly is that they require relatively little time, knowledge and computation capacity. The second advantage of heuristics is that they are rather easy to understand and many people implicitly use them. Unaided decision-making is often done using certain heuristics, and can therefore be a starting point in aiding and communicating with the decision-maker in the decision-making problem (Katsikopoulos et al., 2018). However, the use of heuristics has been criticised by the research program of heuristics and biases of Tversky and Kahneman, in which heuristics, or System 1 thinking, is seen as a 'shortcut' that instills bias resulting in deviation from the 'optimal' outcome (Tversky & Kahneman, 1974). Nevertheless, the use of heuristics in certain cases has reemerged as satisficing strategies; The strength of heuristics, outperforming even complex optimization strategies, are found in dealing with 'ill-structured' problems or situations with high uncertainties as conceptualised by Knight as recent research points out (Artinger et al., 2015).

The choice of suitable MCDA method is not always straightforward and one has to think carefully on what type of data is available and what type of outcomes are needed to aid the decision-making process, see for more information to help with this process Roy and Słowiński, 2013. Key on the choice of MCDA methods is that one carefully ponders and reflects on the context of the situation and goals of the decision-making support to arrive at a good method. Perhaps do a MCDA on different MCDA methods to choose a satisfying one?

1.1.2. PDA
The outcomes of a MCDA gives a ranking of alternatives based on both the outcomes on the established attributes an the preferences of the decision-maker. Traditionally, the aim of a MCDA is to settle on one 'best' project and there is one decision-maker. However, this does not mean that in all situation the decision-maker is confined to making just one choice or that one choice is sufficient. There are situations where a set of actions/alternatives has to be chosen, this is the area of Portfolio Decision Analysis (Lahtinen et al., 2017). The ranking that follows from the MCDA can be used to select the best i amount of alternatives, see equation 1.1.

$$P = \{A_1, A_2, ..., A_i\} \tag{1.1}$$

The main differences making PDA more difficult than a single-choice MCDA are:

- When to stop adding alternatives to the portfolio set?
- Interactions and contingencies between alternatives that have to be incorporated in the performance of a portfolio when selected together.

When one wants to create a portfolio of the alternatives the information of these two points has to be available and taken into account. This 'problem' is conventionally modelled as a version of the 'Knapsack Problem', see Appendix A, section 1 (Pisinger, 1997). The PDA process will, consequently, look somewhat as presented in figure 1.1.

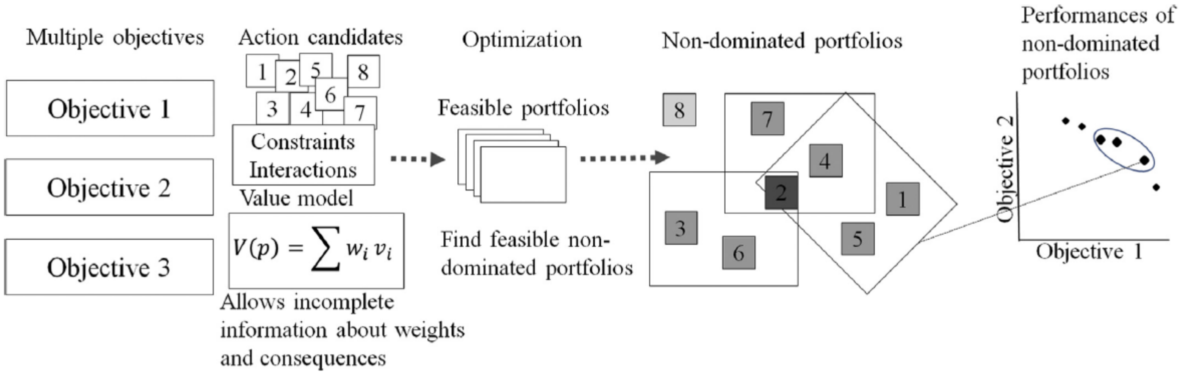


Figure 1.1: Multi-objective optimization approach: Identify the non-dominated portfolios (Lahtinen et al., 2017)

Another, more traditional way to create portfolios is by first generating portfolios through expert consultation and then do a MCDA on these portfolios, see figure 1.2. This method does not need complex models calculating interactions over all possible alternative combinations nor does need a formalised stopping rule. However, this approach also has its own problem of biases and lacking of cognitive abilities to fathom interactions, which are discussed later.

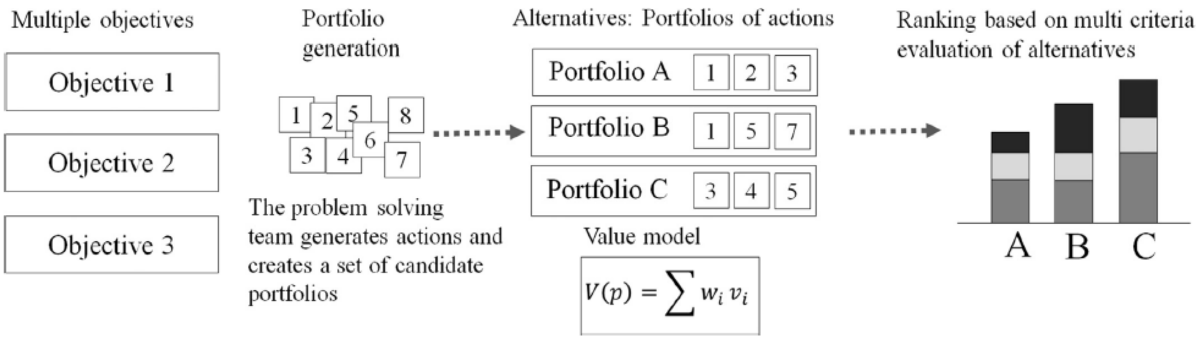


Figure 1.2: Traditional approach: Multi-criteria evaluation of portfolios. (Lahtinen et al., 2017)

In the beginning of the field of PDA much research has been done to the allocation of 'capital' and R&D budgets in companies (Lorie and Savage, 1955; Weingartner, 1966; Fox et al., 1984). Moreover, in the financial sector portfolio analysis is also used to create risk diversifying packages of stocks, obligations, derivatives etc. However, these contexts are not always multi-criteria, since they often optimize for a certain profit or Net Present Value (NPV) and therefore apply economic models, rather than Multi-Criteria Decision models to rank their alternatives and select their portfolios.

1.1.3. Uncertainty

When assessing the portfolio performances and the preferences of the decisionmakers several *uncertainties* have to be taken into account as well. Therefore, a brief overview of the different types of uncertainties found in the MAVT approach is in order. The typology as reviewed by Scholten et al. (2015) is followed. The sources of uncertainty are found in the categories:

1. Problem framing and structuring
2. Attribute prediction

3. Hierarchical aggregation function
4. Marginal (“single-attribute”) value functions
5. Aggregation parameters (“weights”)

The predominant uncertainties treated in this methodology are: Attribute Predictions and the Aggregation Parameters.

These uncertainties derive from the data collected in the process of determining the values, V, of the portfolios. These sources of uncertainty are expressed mathematically as $\psi(V(x,\xi),\phi)$ (Greco et al., 2016, Part V, Chapter 12). ψ is here the operator that incorporates two types of uncertainties, namely ξ and ϕ in the value (V). Here ξ denotes the uncertainty regarding the attribute predictions (2).

ϕ is the uncertainty regarding the preferences of the decisionmakers. This can be uncertainty on the *valuation* of the attribute impacts themselves (4), the uncertainty on the relative importance of the the different attributes (5) and whether or how there is compensation possible between attributes (3). For in-depth discussion on the corresponding assumptions and necessary properties regarding the types of uncertainty I refer to Chapter 5, Discussion, or to the paper of Scholten et al. (2015).

1.2. State of the Art PDA

In other sectors the Portfolio Decision Analysis approach has also been adapted recently to deal with multi-criteria problems that are also portfolio problems; such as environmental sector and water services, infrastructure selection (Lahtinen et al., 2017). The benefit of Multi-Criteria PDA selection is that it combines the explicit elicitation of preferences and structured process to arrive at a ranking of alternatives from the body of MCDA approached and the research to constructing portfolios from such ranking of alternatives from the PDA domain (Lahtinen et al., 2017, Chacon-Hurtado and Scholten, 2021, Durbach et al., 2020, Mild et al., 2015). The paper of Lahtinen reviews portfolio decision analysis approaches and software that can be applicable in environmental management. The paper finds that the use of PDA approaches to the environmental sector has its own advantages but also disadvantages and challenges. The quality of the traditional portfolio generation, as seen in Figure 1.2, depends on the experts ability to construct the portfolios. This is especially challenging when the number of possible portfolios is high. The other challenge is the existence of non-linearities of interactions across sets of alternatives.

"In practice, environmental portfolio problems are often addressed so that experts first generate a number of feasible portfolio alternatives, which are combinations of actions that satisfy the overall requirements. These alternatives are then compared by stakeholders using multi-criteria evaluation to identify the most preferred one. The quality of the resulting decision naturally depends on the experts’ ability to initially construct good portfolio alternatives. This task is particularly challenging when the number of action candidates is high and there are many conflicting objectives. There can also be non-linearities or interactions across the set of actions and their consequences. If this is the case, the overall performance of a combination of actions is not necessarily the sum of the action specific performances. Surprisingly, the extensive literature on environmental multi-criteria decision-making has so far given very little attention to the possibilities offered by portfolio modelling." (Lahtinen et al., 2017)

Here both the importance of studying the interaction with stakeholders/experts in creating alternative and/or portfolio sets and the importance of non-linearities arise. The first matter highlighted by unaided portfolio creation is that certain biases or path-dependencies emerge from champion alternatives, see chapter 7 of Salo et al., 2011. This can lead to overlooking some sets of alternatives that might be very different but also have high performance outcomes. PDA from the onset tries to mitigate this by systematically analysing the alternatives themselves and their impact on the final of Portfolios.

However most of these PDAs are still done in mono-actor situations, this is a property of the system that is often not upheld in environmental cases where multiple actors are involved with varying agency. In some situations incorporating the stakeholders into the problem structuring phase can enrich the alternative and object generation and also build a support base for the project. In this aspect some work has been undertaken to broaden the approach by Lahtinen to multi-actor situations.

1.2.1. PDA in Multi-Actor Situations

Some work has been done on this matter such as by the group of Aalto University, for example with the development of the YODA tool and, in the work presented by thesis of Alessia Matanò.

Vilkkumaa et al. developed a framework/concept of Core Indexes based on 'agendas'(Vilkkumaa et al., 2014). These are shared portfolios for different actors in the same system. In the case description in which they applied this idea for the creation of a national research agenda on wood products. Here this method allows for finding the agreed best performing topics. Similar implementation of this idea is found in YODA.

YODA

An interesting implementation of PDA theories is YODA (Your Own Decision Aid) (portfolio) tool from Luke (the Natural Resources Institute Finland) (Kurttila et al., 2020). YODA is developed for participatory decision processes with multi-criteria. Through a visual interface the preferences of the actors are elicited by identifying acceptable thresholds. The way the portfolios are combined explicitly, see the figure 1.3 (linear) additivity property has been assumed. This is due to the nature of the alternatives, spatial independent peat production locations. Their own analysis shows that the way the YODA tool now operates it does not provide efficient portfolios, albeit feasible ones. Furthermore, small acceptance threshold changes can exclude many projects, thus the method is rather sensitive and strict. However, the core idea of YODA is appealing and the interaction with the stakeholders creates better understanding among the stakeholders as well.

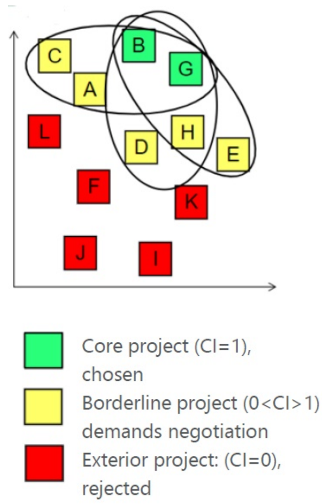


Figure 1.3: Visualisation of Portfolio categories (YODA presentation)

After analysing the alternatives on multiple-criteria the alternatives are either a "core" object, a "Borderline" project or a Exterior project. The core projects would all be chosen by the actors and do not need to be negotiated. The exterior projects would not be chosen by any of the actors and therefore also do not have to be negotiated, see Figure 1.3. These categories can also be seen in “Selecting infrastructure maintenance projects with Robust Portfolio Modeling” (Mild et al., 2015).

The problem with Portfolio implementation in many real case scenarios is the assumption of much known data, preferences and impacts. Furthermore, nothing is noted on uncertainty of outcome impacts, the time it takes to elicitate information and the alternatives/topics in the portfolios are deemed independent. Only in a very narrow amount of case problems this approach can be applied. A similar approach is taken by Alessia Matanò.

Master Thesis of Alessia Matanò
In the thesis of Matanò, she used a PDA model in a multiple stakeholder case for water service portfolio selection (Matanò, 2019). Her addition to the current body of literature is :

"The analysis in this thesis contributes to the current literature by assessing the applicability of the SDM-approach and PDA model to a new topic: the spatial allocation of newly arrived refugees in host countries." (Matanò, 2019, pg. 3)

More specifically, how the refugee allocation process could be supported by PDA. She does give multiple recommendations found at pg.76 (Matanò, 2019):

- Sensitivity of the results to data and value model needs to be tested through a systematic methodology.
- Stakeholder selection and engagement can be improved by adopting a more systematic approach. The engagement of key actors in the decision process can be enhanced by the organization of workshops;
- Value model could be improved in this research by adopting systematic preference elicitation method
- In the Ugandan case study, logical interdependencies among alternatives were not explored.

Acceptability
An important matter for application of Decision Making Support systems is whether the DM accepts the methods and feels confidence in the outcomes. A certain method can create beautiful data graphs and formulate all kinds of unconventional value functions and be justified in doing so. However, if the DM does not understand the method this hurts the decision-making process, since the DM loses confidence (Buchanan, 1994). Which research suggests hurts the quality of decision-making. (Aloysius et al., 2006). There is a tendency to fall back on experience and heuristics in such situations, because they are familiar (Lusk and Kersnick, 1979). A starting point is to carefully balance complexity, accuracy and acceptability when choosing a support approach. These depend on the available information/data, the goal of the support process and the actors involved. One way, when an approach is chosen is to help the decision-maker understand the outcomes of portfolio analysis by choosing a help-full visualisation. Several approaches have been developed to do this; such as *heatmaps*, *modified sunburst diagram* and *interactive parallel coordinate plots*. Providing such graphical insights might mitigate some of the difficulties a DM might have interpreting the outcomes (Salo et al., 2011, Chapter 9). This acceptability is especially of importance when dealing with a multi-actor situations. Both the method and the communication of the method has to fit with the actors involved to achieve the aforementioned confidence.

1.2.2. PDA and interaction optimization

Often it is assumed that alternatives do not have impact on each other when creating a portfolio, as in the peat production case study of YODA (Kurttila et al., 2020) and in the thesis of (Matanò, 2019). In this case a version of the knapsack problem solving algorithms can and is often used to create portfolios and this simplifies the alternative selection, see Appendix A; section 1 (Martello et al., 2000; Jaszkiewicz, 2002).

Sometimes, however, this independence cannot be assumed. Therefore methods have also been developed that do take into account the interactions between alternatives. One method of doing so is adapting the linear-additive portfolio value model with interactions, see Appendix A.2 (Weingartner, 1966; Stummer and Heidenberger, 2003). The benefits or synergies can be taken into account in the selection process in multiple ways, such as dummy variables, which allows for continuing the basic linear approach of optimization (Liesiö et al., 2008). This would mean that an artificial alternative is created which does not really exist, but is has the x_{ij} properties that correspond with the synergy values. This artificial alternative is than added to the set when the corresponding alternatives are part of the set. This approach allows for the continuation of linear approaches.

Another approach is the use of a gamma table (Kantu, 2021). This is similar to the dummy variables, however instead of explicitly adding this information as part of the sets the information is only added after the portfolios are created. Both approaches quantify these interactions and add the values quantitatively to the portfolios to incorporate the non-linearities/synergies. The benefit of the linear-additive portfolio value model with interactions is that it exactly calculates the optimal point as long as the data is complete and accurate.

The main constraint, however, is the availability of the data on alternative performance and the coding of all constraints and dependency combinations if there are many. It is rather laborious work to create all the possible interaction sets/dummy variables. Furthermore, for case studies this data is not always available quantitatively. However, this unavailability of data does not mean it can be ignored. To deal with this problem let us look at the work of Durbach.

Durbach

In certain portfolios the benefits and costs of alternatives are not necessarily additive, as mentioned before; Exact solutions require that all project interactions are assessed. This is not always possible or worthwhile. Here heuristics can play a role for preliminary screening to reduce the amount of alternatives to be assessed or to create sets of portfolio for further examination as in figure 1.2 (Durbach et al., 2020; Kantu, 2021).

Durbach draws a distinction between the heuristics used to improve the computational speed of the optimization algorithms and the heuristics used to asses the alternatives themselves in the tradition of 'fast en frugal heuristics'(Gigerenzer & Todd, 1999) or psychological heuristics (Durbach et al., 2020). Durbach tests several heuristics, some with and some without positive interactions. In the end a set of best heuristics is presented. A PhD student under mentorship of Durbach also tested these heuristics to select portfolio of research papers also incorporating the negative interactions (Kantu, 2021). So they developed a set of heuristics that created portfolios close to the 'optimal' linear-additive portfolio value model with interactions, using only a fraction of the information and computational time in the case of the heuristic 'Unit value with Synergy'. Among the developed and tested heuristics are:

1. **Highest Value**: Adds projects in descending order of their values.
2. **Lowest Cost**: Adds projects in ascending order of their costs.
3. **Unit Value**: Adds projects in descending order of their value-to-cost ratios. Values are based on individual project values only.
4. **Unit value with Synergy**: Identifies all projects that are involved in at least one positive interaction. Adds projects from this set using the Unit Value heuristic i.e. in descending order of their value-to-cost ratios, with values based on individual project values only. Once this set has been exhausted, adds projects from outside the set, again using Unit Value.
5. **Added Value**: This heuristic adds the project whose selection would lead to the largest increase in overall portfolio value per unit cost. The incremental benefit includes the individual value of the project, as well as the value of all interaction subsets that would be completed if the project were to be added.

Some heuristics need the same information as the linear additive functions with interactions and some do not. The difference with the linear additive function method is that due to the heuristic not the entire 'problem' space is calculated but only in the greedy sense of the next step. One potential danger of this method that the analyst has to be aware of is that this method can create path-dependency. However, the application of these heuristics allow incorporating interactions where information and time are limited. Furthermore, the nature of the heuristics as a method of selection is intuitive which might help the support and communication process in a multi-actor case study, as already mentioned in section 1.2.1: Acceptability.

The paper “Fast and frugal heuristics for portfolio decisions with positive project interactions” concludes: "Our simulation results showed that two heuristics, Added Value and Unit Value with Synergy provided outcomes that were competitive with theoretically optimal models under a fairly wide range of environmental conditions." (Durbach et al., 2020).

For further research it is proposed that this method can be used as a drop-in replacement for more information-intensive optimization methods, appropriate for applications where time or other constraints make it impossible to assess the information required by optimization methods.

The problem with the approach of Durbach is that it uses a greedy algorithm, which can create path dependencies and does not explore the entire solution space. Also it is not tailored to multi-actor situations per se. Uncertainties of the impact of alternatives in a portfolio in itself were not mentioned. In that sense, information on synergies was the focus and the other information was assumed to be present. The idea of applying such heuristics to improve the decision quality without increasing the cost of doing the decision analysis can found by both Gigenzer and predominantly Keisler who applied it to the Decision making field in the form of MCDA/PDA. Keisler, 2004; Keisler, 2011. Let us try to synthesise the gaps from these two promising research strands in Portfolio Decision making; at one hand mathematically dense analysis of dominance or efficiency in stakeholder analysis (mainly Finnish Universities) and at the other hand the research strand of application of heuristics. (Durbach and Keisler in the tradition of FFH)

1.2.3. Literature Gaps

From studying the literature there seems to be an opportunity in the research to Multi-Criteria Portfolio Decision Analysis. On the application of portfolio decision analysis it is argued that certain decisions challenges would benefit from the structuring as portfolio problems. The guiding principles of PDA can help to mitigate biases and prevent overlooking beneficial alternatives in the portfolio creation by stakeholders. It also forces to make aspects explicit and provides information for better decision-making. Because of these advantages this approach has now been applied in a couple of situations, mainly infrastructure project selection (Mild et al., 2015; Liesiö et al., 2007), the domain of water services allocation (Matanò, 2019) and environmental management (Kurttila et al., 2020; Lahtinen et al., 2017; Chacon-Hurtado and Scholten, 2021). However, little research is done in the application of such methods to multi-actor situations and to the communication and support of these, the studies of Matanò, Vilkkumaa et al., 2014 and Kurttila et al., being notable exceptions. However these portfolios are of a particular independent nature (no synergies) with rather ’certain’ data.

Another gap is on the point of interactions. Current methods have been developed for very similar alternatives, but Portfolio decision context do exist where alternatives are dissimilar. Furthermore, the methods, up to now, tend to assume the availability of information on these. The application of FFH is still very undiscovered terrain and there seems to be potential for PDA analysis in such situations where time is scarce and information limited. Heuristics can be developed and the communication of these to be communicated to the relevant actors. These two research strands can combat the realities of many situations in the field where conventional approaches can not be upheld. Research to deal with such challenges can, therefore, attribute to the support of many real-life case studies. Case studies that are Multi-Actor, that lack extensive preference information, that lack time and accurate alternative information need a robust method that is understandable. Such a situation asks for a different approach due to the different preferences and insights in the situation and above all, agency, than when only one DM is concerned. Different actors can intervene in different matters, combining these alternatives available to the different actors in a combined portfolio would be novel and useful to support decision-making. It is therefore useful for many real-life case studies in which the uncertainties of attributes measurements have to be taken into account.

Taking into account the aforementioned questions and considerations the main research question of this thesis proposal is:

1.3. Research Question

"How to create and explore portfolio sets in multi-actor and synergy situations with a lack of time and access to detailed preference information"

The practical nature leads to a natural dissection of the main research questions into several sub-questions.

1.3.1. Sub-Questions

From the main research question and the identified approach a set of sub-questions are delineated:

1. What are the challenges and suitable ways to create the feasible portfolios?
2. How to deal with uncertainties in the portfolios and the preference modelling?
3. How to incorporate interaction information, so-called synergies, where quantitative data is lacking?
4. How to select good-performing portfolios in the wake of the above mentioned uncertainties in multi-actor situations?

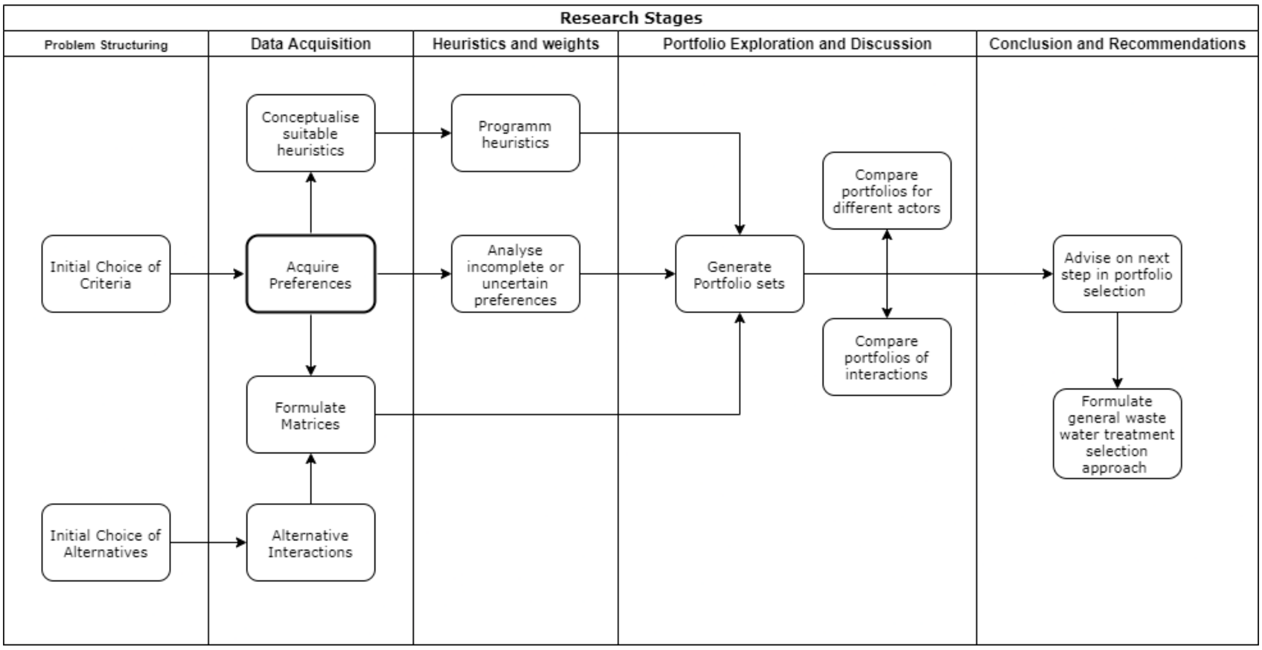


Figure 1.4: Research Flow Diagram

1.3.2. Goal

The before-mentioned steps in the flow diagram, see Figure 1.4, leads to the over-aching goal of this thesis. The goal of this thesis is the development of a methodology to support the decision-makers in a situations where there are multiple actors, where the performance of alternatives are uncertain, where multiple alternatives have to be selected and interactions are expected between these alternatives. Furthermore only limited preference elicitation is possible. The methodology is specially developed for such situations and can provide a shortlist of alternatives/portfolios that are acceptable to the actors and provide information on the importance of the attributes.

1.4. Relevance with EPA

The research question and the following thesis builds upon and further develop current methodologies to support decision analysis. One direction in which the methodology is developed further is especially to deal with a multi-actor situation. Which is one of the requirements of the EPA thesis. Furthermore the case study analysed delves into the intersection of engineering (WWT etc) and society and policy. I develop and use methods and techniques for problem analysis and exploration, which are used systematically. The challenge of long-term availability of potable water is also a Social Development Goal (SDG 6) and hence part of a Grand Challenge. Hence, there is a strong link with the EPA Master and it fulfills the requirements.

2
Materials & Methods

2.1. Materials

The material used in this thesis is a Lenovo laptop, operating on a 64 bit operating system of Windows 10. The system contains 16 GB RAM, Processor: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s).

For the analysis code has been written in Python3 (Van Rossum, 2020), run on Jupyter Notebook for generation of graphs and for documentation purposes (Kluyver et al., 2016). Analysis without documentation in one go can be repeatedly done in another interpreter if wished so. Files are stored in both .py and ipynb. To speed up certain aspects of the analysis, code and data are vectorised and calculations are carried out using the numpy library (Harris et al., 2020). To run parts of the analysis not all the datasets have to be recreated and therefore the code is written in a Classes format. For the handling of dataformats the panda library is also used extensively (McKinney, 2010). The input data is formatted in Excel and consequently loaded in using methods from the same panda library. For the sensitivity analysis the SaLib library is used (Herman & Usher, 2017). The entire code is both available on GitHub and in Appendix K. For graphical visualisations functions have been used from the matplotlib and seaborn libraries (Hunter, 2007; Waskom, 2021). For fast iterative processes to create combinations the itertools library is used (Van Rossum, 2020). The thesis has been written using the Overleaf interpretation of the LateX typesetting system.

2.2. Methods

The Methods outlined in this section are the results of careful considerations and are part of the findings of the study. Here the data and methods used to carry out the PDA are outlined to handle situations as described in the research question. The Result chapter shows the outcomes of applying the noted methods and in the Discussion chapter the explanation, justification and limitations of the formed method are elaborated upon. A broad overview of the method is shown in Figure 2.1, and subsequently the individual steps are explained below.

2.2.1. Problem Structuring

As with most MCDA and PDA problems the first phase is the problem structuring phase. This has been done by dr. Scholten and dr. Saharan who had access to the actors in Kanpur for the case study. From them via interviews and co-creation workshops the problem has been identified and a cognitive map has been created. From this cognitive map the objectives were constructed and from this objective hierarchy the attributes and alternatives were developed. The subsequently discovered alternatives and attributes are the frame of the decision matrix. As this is not part of the discussed methodology it is assumed the alternatives and attributes are discovered and specified.

Similarly the way the preference elicitation is done also falls outside the scope of the thesis. Consequently the data regarding the alternative performance on the attributes have been collected.

The method is developed in such a way that it can handle information stored in the format of decision matrices. The method has been developed to be able to handle mxn decision matrices of different sizes (Table 1.1). The case study is for illustrative purposes.

In the Excel sheet the Alternatives are stored in the first two columns (alternative number and alternative name) in the sheets "Sheet2" and "Deviation". The Attributes are noted down in the first

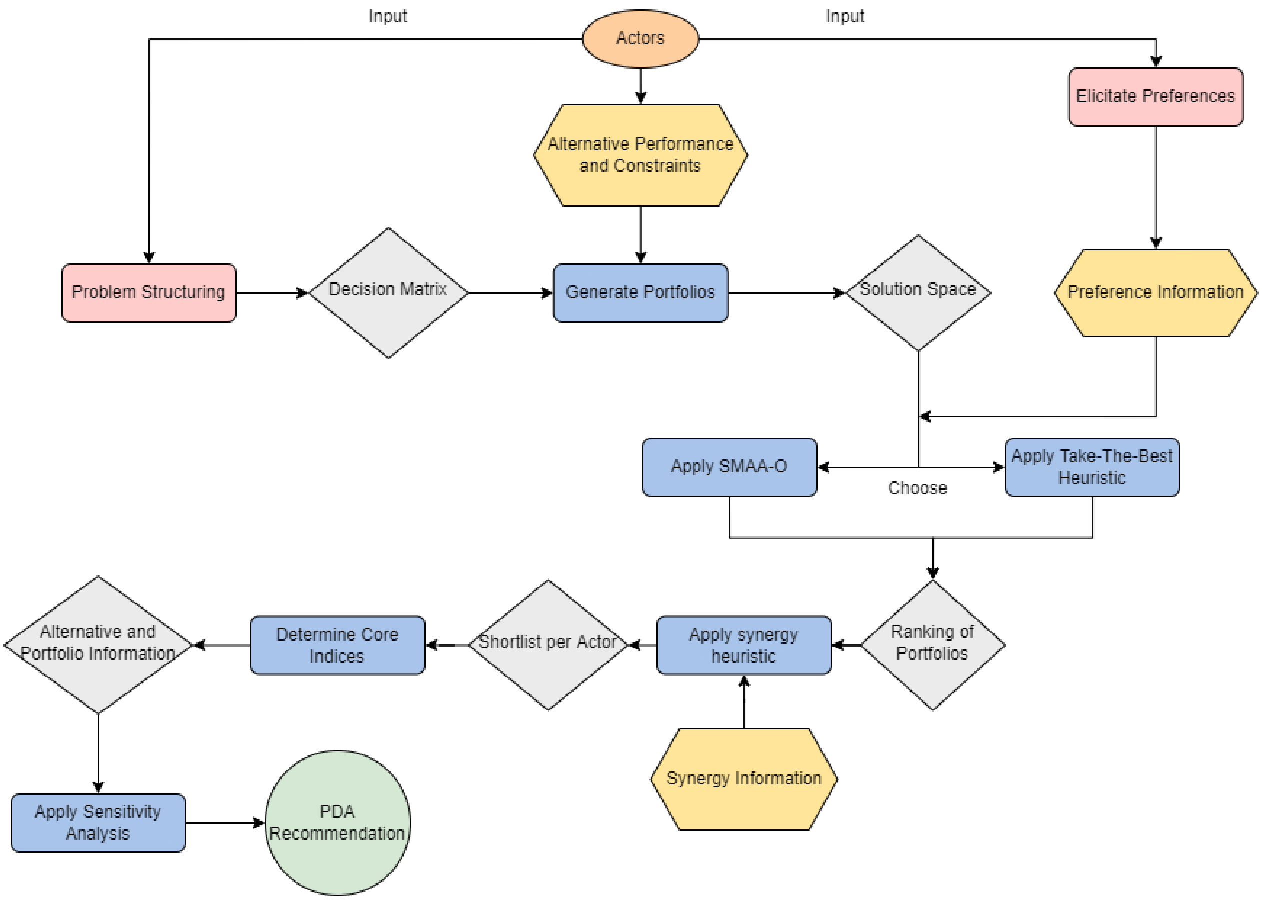


Figure 2.1: Overview of the methodology: Collected Data (Yellow), Application of Methods outside the scope of the thesis (Red), Application of Method/Technique part of this Thesis (Blue), Intermediate Results (Grey), Actors involved (Orange), Final result (Green)

row of the Excel sheets "Sheet2", "Deviation", "Criteria" and "Units". Sheet2 resembles the Decision Matrix as shown in Table 1.1.

2.2.2. Multi-Attribute-Value Theory

As already described in the introduction there are many methods to rank a set of alternatives. In this thesis the multi-attribute-value theory (MAVT) is chosen as the basis of the portfolio analysis. In the Introduction section this approach has been explained. An important assumption is made as to the aggregated value function; the linear-additivity of the marginal functions. There are, however, some properties to which the attributes and data have to adhere to assume the linear-additivity (Belton and Stewart, 2002; Eisenführ et al., 2010 (Chapter 6)). See Appendix B for an overview.

$$V_{P_i} = \sum_{i=1}^{m} w_i v_i(x_i) \tag{2.1}$$

This approach leads to a certain form of preference data to be collected, outlined below in the subsection Data Collection.

2.2.3. Data Collection

Information for the case study has been collected and is defined for three different 'uses': 1) Alternative performance impacts, 2) Interactions and constraints regarding portfolios and, 3) Preference Information. This set of data has been obtained in different ways; via literature study, on the ground empirical study by other research groups in the PG project and via expert consultations for the case study, see Table G.2. For the simulated data that is used, see Appendix F.

I) Performance Data
The alternatives impacts on the attributes to fill in the x_{ij} of the decision-matrix are obtained. The method uses cardinal data with a type of uncertainty distribution and corresponding data to indicate the uncertainty of the impact (Belton & Stewart, 2002).

Since portfolios deal with a change to some base value the *value impacts must be noted as the change to the current situation and not the new situation itself*; otherwise these values cannot be aggregated. These matters are part of the problem structuring phase, but since it matters for the analysis this has to be taken into account in the formulation of the attributes before the analysis.

Global or local value scales over the attributes have been determined, limiting their scale between the best and worst conceivable performance outcome (Belton & Stewart, 2002). This ensures a scale that is independent from the initial set so that new alternatives can be added later without value rescaling.

II) Interaction Information
Secondly, what are **interactions between the alternatives**? The interactions between alternatives can be categorised in exclusivity, dependency and synergy information.

Exclusivity Certain alternatives can exclude each other for a plethora of reasons and this has to be known before portfolio generation. It ensures that due to logical constraints alternatives that cannot be combined will not be combined. In the Code this information is processed within the Class "$Data\_Acquisition()$ calling the method $add\_mutually\_exclusive(alt1, alt2)$ from the $Explicit\_Space()$ Class.

Dependency Similarly to the Exclusivity information, an alternative (sub) can sometimes only be included when another certain alternative (parent) is selected in the Portfolio (this does not have to be the other way around, otherwise one should combine them in one alternative.) In the Code this information is processed within the Class $Data\_Acquisition()$ calling the method $add\_subsequent(parent, sub)$ from the $Explicit\_Space()$ Class.

Synergy The interaction information that was gathered is not a logical constraints but is information on non-linearities: **what are the alternative sets/combinations that create synergy** in portfolios and are they either positive or negative (as outlined in Appendix A.2 section A.2 (Kantu, 2021)? This synergy data has been collected in the form of pairs of alternatives for application of the heuristics, such as *"Unit value with Synergy"* (Durbach et al., 2020). The interaction information is either a simple +, - or 0. The positive interaction pairs were noted in the Excel tab "Interactions".

III) Preference Information
Attribute Weights The preference information on the attributes for the actors has been elicited/simulated, however the elicitation technique to obtain this information falls outside the scope of the thesis. An ordinal ranking of the attributes has been simulated and gathered to be used in the aggregated value function. In the Excel the attributes ranks were noted in the "Criteria" sheet, second row.

Value Functions The second sort of preference data within the MAVT approach are the value functions. The cardinal performance data are to be translated to values for the stakeholders via value functions on the attributes. In case studies there might be some different value functions for the different attributes for the different actors; however for the sake of manageability it is wise to agree on one value function (shape) per attribute as it is a first rough analysis of the portfolios.

These value functions that are created for each attribute are the so-called "marginal" or "partial" value functions (Belton & Stewart, 2002).

Subsequently an increase of the performance reflects corresponding marginal increase in the perceived value for the stakeholders. **A very important assumption in the analysis for this thesis is the linearity of the value functions.** This is done due to time constraints and elicitation difficulty in the case study. Furthermore, there is much literature on the construction of value functions to which this thesis does not aim to contribute. Attributes are noted as either benefit or cost functions, see 2.2 in the Excel sheet "Criteria", row 3.

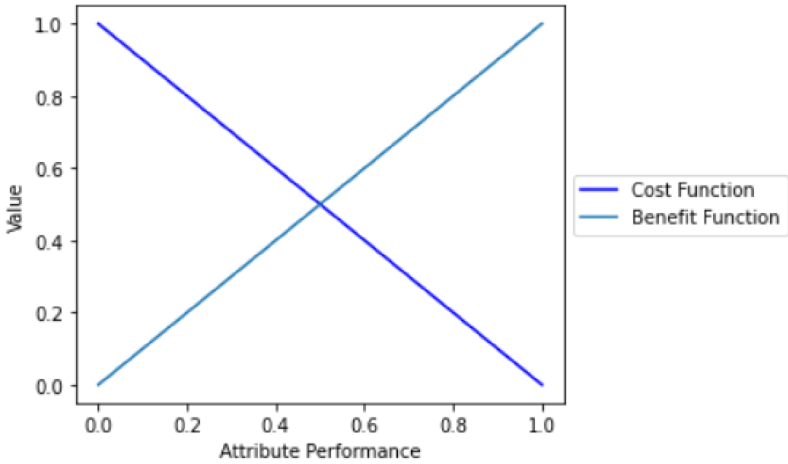


Figure 2.2: Shape of value function; either defined as cost or as benefit function

2.2.4. Applied Methods

The above outlined data gathered for the study has been analysed using a body of methods which are outlined here. The methods or techniques used are run in Python code and are shown in the Appendix K. The different Classes referred to in this section are indicated as the place where the relevant part of the analysis have been carried out in the code.

2.2.5. Portfolio Creation

To analyse the possible portfolios these first have been created from the set of alternatives.

All the feasible portfolios are generated: covering the entire solution space, see the chapter 5, Discussion, for the considerations. This approach *combines all possible alternative combinations* so that no good performing portfolios are excluded in the analysis. How to generate all combinations efficiently? Firstly, there is a difference between combinations and permutations. It is assumed that the order in which the alternatives are put into the portfolio does not matter (e.g $P = \{1, 2, 3\}$ equals $P = \{2, 1, 3\}$). All these combinations together are called the "solution space". The solution space is not straightforward to generate. The logic is that from a certain set, let us say 1 to 5, every number can be chosen once. However, it is not a given that the set has to be exhausted. This means, we can also select just one number from the set. But we can also select four numbers in the set. This is the *maximum alternative amount* and can be a constraint to cap the amount of alternatives.

For a set of 9 alternatives $\{1, 2, .., 9\}$ if the maximum alternative amount is 5 this means that the maximal amount a combination/portfolio can have is 5 alternatives selected from those 9 alternatives. P = $\{1, 2, 6, 7, 9\}$ would be one such portfolio. But it gets more complicated. Assume there is no minimum amount of alternatives that have to be selected. So even if the maximum amount of a portfolio set encompasses 5 alternatives there is no reason why it could not compose of only 3. The other two 'spaces' are empty. These options/portfolios are also part of the solution space. This complicates generation of the combinations. The algorithm iterates through these combinations by first creating all combinations where 1 alternative is allowed, subsequently it generates all portfolios where 2 alternatives are allowed and so on until the maximum amount of alternatives are included. Here the analyst can choose whether to cap the amount of alternatives in a portfolio. When a logical constraint is encountered this combination is not included. This data (II Interaction information) is incorporated before the generation of the solutions space. The algorithm checks whether an exclusive or dependency occurs in the combination before adding the to the generated set of feasible portfolios. See Appendix K where the Class $Explicit\_Space()$ is the above described code. In the $Data\_Acquisition()$ Class this Class is called upon to generate the portfolios for the data given as input.

2.2.6. Portfolio values

From the collected alternatives per portfolio the assumption is made that the values can be linearly aggregated within each attribute. The impact values of the individual alternatives in Portfolio P are summed over attribute, i, for m attributes to get the 'base impacts',x_{bi}, for that particular portfolio, P in the following way (Golabi et al., 1981; Vilkkumaa et al., 2014;):

$$x_{bi}(P_i) = \sum_{i=1}^{m} A_i \; for \; A_i \in P_i \tag{2.2}$$

The current code checks in the Excel sheet "Units", whether the attribute is expressed in percentages and then calculate the additivity for such attribute as in eq. 5.1, since here linearity is not upheld per definition of the attribute, see the Discussion for an elaboration. This operation was carried out in the $Data\_Acquisition()$ Class using the run() method that calls upon other methods to select and aggregate the data from the individual alternatives in the Excel sheet and give a Panda DataFrame output ($Portfolio\_Data$) of all the portfolios containing information on the underlying alternatives.

2.2.7. Uncertainty on Portfolio Performances

In order to propagate the uncertainties of the performance of the alternative, ξ_i, Monte Carlo Sampling has been used. The Monte Carlo simulation *Inverse Transformation Method*" (ITM) draws samples (n= $1 * 10^5$) from the established uncertainty distributions of each alternatives. These samples were drawn from each alternative distribution in the portfolio set and summed for each n. This led to the Portfolio Performances, which have been expressed via probability density functions (PDF), see Figure 4.7. See Appendix K, where the Class $Uncertainty\_Alternatives()$ is called upon by the class $Uncertainty\_Portfolios()$ to generate the uncertainty distributions of the portfolios by propagating the uncertainties of the alternatives. Normal, Log normal or Uniform distributions have been used as input to describe the expected impact of most alternatives. For Portfolio, j, this leads for all attributes to a vector $\boldsymbol{\xi_j}$ containing the Monte Carlo distribution functions, ξ_{ij} for each attribute i:

$$\boldsymbol{\xi_j} = [\xi_{1i}, \xi_{2i}, ..., \xi_{ij}] \tag{2.3}$$

For each of these ξ_i there is a corresponding partial value function to calculate the corresponding values on attribute i:

$$\Xi_i(\boldsymbol{\xi_i}) \tag{2.4}$$

Important in the definition of the value function are the scales on which the values are determined. These are either local or global scales. For a global set, pre-defined, scales are used. When a global scale is identified and the individual values of the alternative exceeded the maximum or minimum boundaries as defined by the global scales the outcomes were capped.

For a local set the scales were obtained from the ξ_i distributions over all j portfolios. The highest and lowest attribute performance values sampled in the sampled set of $(n \cdot j)$ matrix, $\boldsymbol{\xi_i}$, were used (n being the amount of Monte Carlo Samples and j the amount of portfolios). Hence, for a benefit value function the corresponding partial value distribution on attribute i was determined via:

$$\Xi_{ij}(\xi_{ij}) = \frac{\xi_{ij}}{(x_{max} - x_{min})} - \frac{x_{min}}{(x_{max} - x_{min})} \; with \; x_{max} = min(\boldsymbol{\xi_i}) \;\; and \; x_{min} = max(\boldsymbol{\xi_i}) \tag{2.5}$$

Integrating these partial value distributions gives the expected value, E[X], which has been used to rank to portfolios.

2.2.8. Uncertainty on Preferences

The "internal" uncertainty, ϕ, stems from the preference elicitation process. The uncertainty of the shape of the partial value functions is not delved into due to the assumption made on the shape of the functions.

The other major source of internal uncertainty is the accuracy of the attribute weights obtained, $w(\phi)$. As ordinal data is used as input, this is a major source of this uncertainty. The incorporation of the two uncertainties ξ and ϕ in the aggregated value function result in the altered aggregated value function, for portfolio j:

$$V_j = \boldsymbol{w}(\phi) \cdot \Xi_j(\boldsymbol{\xi_j}) \tag{2.6}$$

The vector containing the aggregated values of all portfolios is denoted as $\boldsymbol{V} = \{V_1, V_2, ..., V_j\}$ How to rank the outcomes of vector $\boldsymbol{V}$?

One of the approaches found that did deal with the ordinal ranking uncertainties is the SMAA-O approach (here the O stands for ordinal) (Tervonen & Lahdelma, 2007). This therefore is opted as a suitable way to deal with the uncertainty. However, as it is a rather complex method it is compared to a simpler Take-The-Best-Heuristic approach. This research studies the use of these two approaches as a method to deal with limited attribute preference information. These two methods have thus been applied in this thesis to deal with uncertainty in the attribute preferences weighting of the actors.

Stochastic multicriteria acceptability analysis (SMAA-O) The SMAA-O approach has been used (Lahdelma et al., 2003; Tervonen and Lahdelma, 2007) to process the attribute preference information in situations of complete ordinal ranking of attribute preferences, see Appendix C for explanation of the method. The method produces acceptability indices with a percentage score on the ranks the portfolio achieves with the ordinal ranking of the particular actor. In the Code this analysis is carried out by the $Values()$ Class by the $ranks\_2()$ method sampling 100000 weight vectors. The $acceptability\_indeces()$ method formatted the outcomes in a Panda's DataFrame.

Take-The-Best-Heuristic The other approach used are heuristics to deal with incomplete ordinal ranking of the attribute preferences. The use of heuristics has already been discussed in the introduction. Applied is a set of heuristics identified based on the work of Simon and Gigerenzer, the so-called take-the-best heuristics (Gigerenzer & Goldstein, 1996).

The steps that go into the take-the-best heuristic are (Gigerenzer & Goldstein, 1996):

1. **Search rule**: Look through cues and evaluate which one will allow for fast but accurate decisions.
2. **Stopping rule**: Stop the search when you encounter the first cue that will differentiate between alternatives.
3. **Decision rule**: Make a decision based on which scores better according to the chosen cue.

From a set of portfolios, for which the partial value functions are determined the aggregation step is forgone in contrast to the SMAA approach. The set of information on which one than selects the portfolios are in the format as presented in Table, 4.1. The search rule used contained one cue: What is the most preferred attribute. The heuristic searched through the set of values on that particular attribute. The result has been a ranking on that attribute and corresponding selection of the top performing portfolios on that attribute. In the Code the method $heuristic()$ from the Class $Values()$ applied this step.

2.2.9. Synergy Heuristic

After dealing with the preference information a ranking of portfolios for each actor has been achieved. These portfolios have a certain expected value, E[X], due to the uncertainties identified above. A heuristic has been applied to these portfolio sets. The heuristic searched through the portfolios to check whether synergy pairs were present in the portfolio sets. Only portfolios containing a pair of alternatives were kept in the list. This achieved a sub list of the original list of the best performing portfolios that do also have synergy interactions.

The assumption on synergy here is important. Superadditivity is assumed, meaning that the 'real' value of a set of alternatives is more than the *sum* of these alternative values. Similarly, negative values, that have not been processed, would be seen as lower than the addition of the two alternatives (subadditivity). See Discussion chapter for more in depth considerations. In the Code the Class $Interaction()$ handles this heuristic process.

2.2.10. Multi Actor Analysis

Since case studies are often multi-actor systems the analysis has been for multiple actors. First and foremost, the external uncertainties and performances of the alternatives on the system, including their dependencies and interactions are assumed to be equal for the actors. The preferences of the actors are however assumed to be different. This information set S includes all information and uncertainties of portfolio performance and weights for the corresponding actor: S = $\{V(\xi, w(\phi))\}$. The set of Portfolios for actors is denoted as $P_N(S)$ for information set S.

To analyse the outcomes for the portfolios first it is determined whether there was overlap in portfolios.

Portfolio Overlap If a certain (set) of portfolios reemerged across all actors than this set is agreed upon as part of the final shortlist of the actors.

Consensus Indices Subsequently, the Consensus Indices (CI) were determined for the top ten performing portfolios in the shortlist emerging from synergy heuristic selection of the SMAA-O analysis for all actors. An implementation similar to Vilkkumaa et al., 2014 and YODA has been used. This CI is studied to see which alternatives were found in the portfolios and where consensus was found. Thence the Core Index has thus been established (Liesiö et al., 2007).

$$CI(x^j, S) = \frac{\{\mid p \in P_N(S) \mid x^j \in \ p \mid\}}{\mid P_N(S) \mid} \tag{2.7}$$

To increase the visual presentation the table also has be presented as a heatmap to indicate consensus over attributes. For each actor the ordinal ranking in the Excel file, sheet "Criteria" has to updated, saved, and the 'Value' Class module has to be rerun. This gave the portfolio shortlists and consequently also the Core Indices of the actor.

2.2.11. Sensitivity Analysis

Another important matter for continuing elicitation has been to discover which attribute contribute most to the value outcome of the aggregation function. This allowed for more precise focus of elicitation efforts. Therefore a sensitivity analysis has been carried out. The outcome of the values were dependent on the composition of the portfolio themselves, hence a sensitivity analysis was carried out over these portfolios.

A sensitivity analysis allows for prioritisation of the most important variables (factor-prioritisation) and which can be safely ignore (factor-fixation).

How do the weights over the marginal value functions affect the outcome. A Sobol analysis is undertaken (Sobol, 2001) to determine this. For a short explanation of the Sobol implementation see Appendix D.

2.2.12. Final Recommendation

The found impacts have been combined to give feedback on the actors. This feedback contains the shortlist of portfolios most preferred by each actor, the consensus over these portfolios and underlying alternatives and attributes that have impacted the outcomes most. The values of the outcomes of analysis are shown with explicit uncertainties on the attributes. This information allows the actors to ignore much of the possible solution space in favour of a select set of portfolios for more focused decision making.

3
Case Study

3.1. Pavitra Ganga project

As part of the Horizon2020 programme the E.U. in a collaboration with the Department of Biotechnology of India has set up the Pavitra Ganga (PG) project to study the challenges regarding wastewater treatment in India in light of the SDG6 goal (Ensure availability and sustainable management of water and sanitation for all). The goal of the project is to study two case studies, Kanpur and Delhi, which have been selected as exemplary of the challenges waste water treatment in India. The findings of the projects aim to support the Indian institutions and municipalities implementing novel technologies, suitable for the Indian conditions and develop water governance best practices (See PG site). Furthermore, the innovative technologies developed might result in new business cases and start-ups.

3.1.1. Kanpur

Kanpur a large city situated in the Indian state of Uttar Pradesh (UP) and lies on the banks of the Ganges river . This stretch of the river is deemed most polluted of the entire length. The pollution here is so high due to a combination the industrial and domestic waste water(Santy et al., 2020; Khatoon et al., 2013).

The waste-water treatment challenges in Kanpur are threefold. First Kanpur underwent heavy urbanisation and currently has a population of 3.1 million inhabitants. This has put a strain on the existing infrastructure, both drinking water and sewage treatment. Domestic potable water demand is estimated at 600 million liters per day (MLD), but supply is limited to only 385 MLD and only half of the sewage is treated (Bassi et al., 2019). Currently, the government is expanding the sewage system and the STPs in existence currently can handle the wastewater input, moreover they are performing under capacity. The problem here is the transportation to the STP's.

The second challenge is the discharge water of the tannery industry. Kanpur has a large industrial sector mainly consisting of roughly 400 tanneries, which are the economic core of the Jajmau sector and to a part the city of Kanpur. The economic impact of the tannery industry is very significant. However, these tanneries are considered gross industrial polluters and have therefore, officially, to oblige to certain standards. These tanneries are obliged to have a treatment plant within their premises, so-called Primary Effluent Treatment Plants (PETP). However, these plants often do not work properly as indicated by actors in the field and indirectly seen from UPCB measurements (Central Pollution Control Board, 2021). As a result much of the effluent water still is heavily polluted. The effluent of the PETP would then have to be mixed with domestic wastewater to make it more treatable in the common effluent treatment plants (CETP). Currently, discharge of the tanneries varies and is much above the CETPs capacity of 9 MLD industrial waste water, both quantitative and qualitative (Central Pollution Control Board, 2021). This makes treatment by the CETP's difficult, since the CETP has not been developed for such high tannery effluent flows. Due to power shortages, cost and compliance issues, treatment plants also do not work at capacity, especially the PETPs. The problems in Jajmau are particular the heavy metal components of Chromium (Cr) and salts in the wastewater effluent as result of the leather production (Central Pollution Control Board, 2021). Further elaboration on the tannery process is found below. Other metals, probably from other industries are also measured in the water. A further complicating aspect of the Jajmau sector is the intertwining of the domestic and industrial settings. Much wastewater from, often informal, production operation within private households is discharged directly into the sewer, lacking any treatment. The wastewater from this industry might also end up in the regular' STP (Sewage Treatment Plants) ill-devised to handle industrial discharge water.

The third challenge is the use of this insufficiently treated wastewater for irrigation purposes by farmers downstream. The effluent from the CETP is discharged in an irrigation channel that provides much of the farmlands downstream. This practice leads Chrome and Saline contaminated soil leading to health risks. (Dotaniya et al., 2017). Simultaneously, due to the high presence of sulphates and phosphates in this effluent the wastewater has also fertilizer properties as indicated by the actors. There is thus, to put it like this, a hate-love relationship among the farmers towards this water. Further elaboration on the wastewater-treatment systems is present further down in this section as well.

The problem structuring resulted in a objective hierarchy. From this hierarchy 11 alternatives concerning wastewater treatment technology were investigated and scored on 11 attributes for 4 stakeholders. These attributes have to be direct, operational, understandable, comprehensive and unambiguous. For a complete elaboration on the choice of objectives I refer to the paper: “Selecting Attributes to Measure the Achievement of Objectives” (Keeney & Gregory, 2005). This is an aspiring ideal. The method and code developed in this thesis is flexible enough that implementing a different selection of attributes and alternatives is straightforward. In this way the deliverable of this thesis can be used for more comprehensive further research, as long as the relevant data is available. Next to these attributes and alternatives information regarding synergies have been collected and for 4 of the 5 actors preference information has been collected, see Saharan et al., In preparation. From these alternatives portfolios are created and the outcomes are presented in the Results section.

4
Results

The structure of this chapter follows the structure of the research questions. Following the research approach discussed in the Introduction chapter the findings for portfolio analysis are shown using the methods outlined in the Materials and Methods Section. It deals first with the input of the portfolios, than treats the uncertainties and subsequently the sensitivities. This yields the shortlist of portfolios, alternatives and attributes that matter most. The goal of the analysis must thus be kept in mind; to obtain quickly a shortlist of portfolios and information on the important attribute information. The goal is not to arrive at a top performing portfolio. This method is just one way to treat the data, albeit in a rather unconventional way using both stochastic methods and heuristic methods. For a discussion on limitations and considerations see chapter 5 Discussion that follows this chapter.

4.1. Portfolio Creation

The outcomes of the Portfolio Creation approach is satisfying and renders fast, see 4.1.

The code for this thesis, presented in Appendix K, can handle the portfolio amounts without any problem up to a set of 14 alternatives within an hour when run via Juypter Notebook.

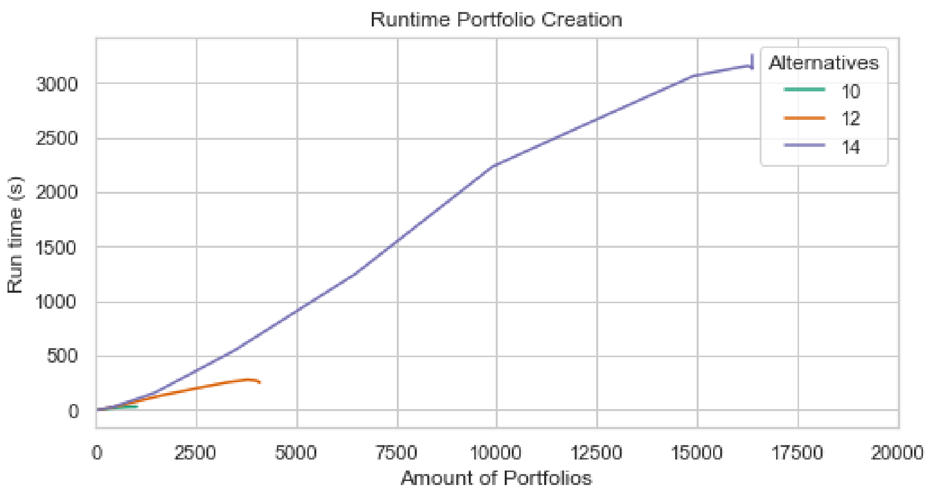


Figure 4.1: The run time in seconds of the piece of code to generate and collect the data of the portfolios for different alternative sets; The longest run times are for the maximum number of combinations and are 54 minutes for 14 Alternatives (16383 Portfolios); 4 minutes and 12 seconds for 12 Alternatives (4095 Portfolios); 30 seconds for 10 Alternatives (1023 Portfolios).

4.1.1. Case Study Portfolios

The Case Study Decision matrix, including the exclusivity constraints results in the creation of 673 portfolios. The time to create this solution space and collect the data for each portfolio was 371.58 seconds.

4.2. Portfolio Performance Uncertainty

See Figure 4.2 for simulated data in a normal distribution.

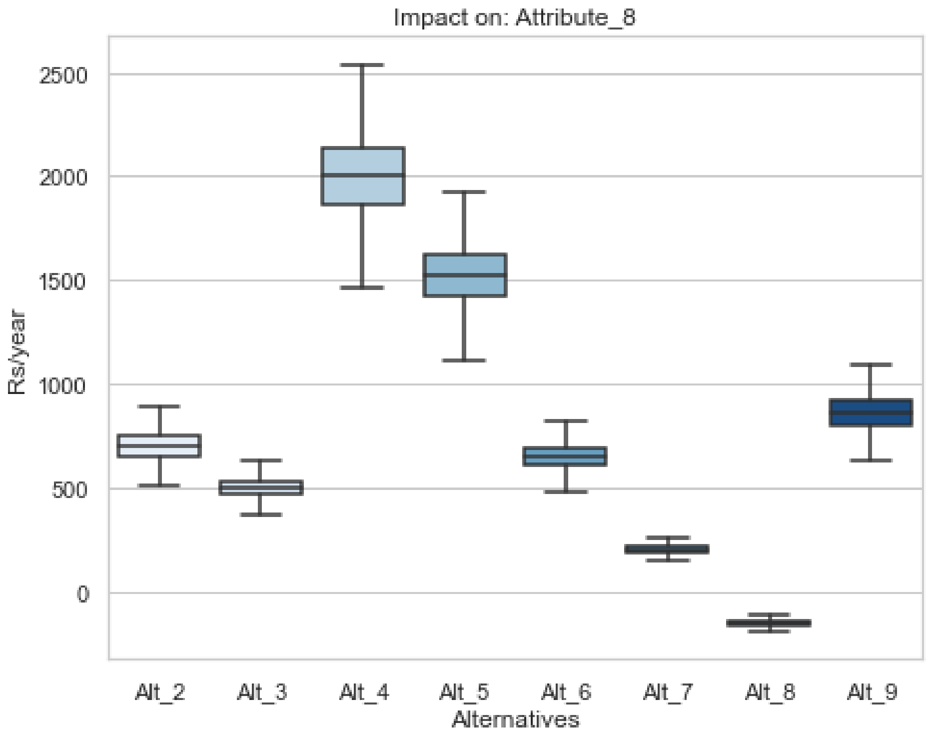


Figure 4.2: A boxplot representation of the impact of individual alternatives on Attribute 8.

Specific distribution data can be handled for each alternative if the information is available. For Alternative 2 different distributions are simulated. For a distribution of alternative 2, expressed in a normal distribution, see equation 4.1, from which $1 * 10^5$ random samples are drawn. This accordingly generates a distribution presented in boxplot for Alternative 2 (Figure). Similarly data can be formatted/generated in log-normal and uniform distributions, for alternative 2 examples of such boxplots are generated.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma}^2)} \tag{4.1}$$

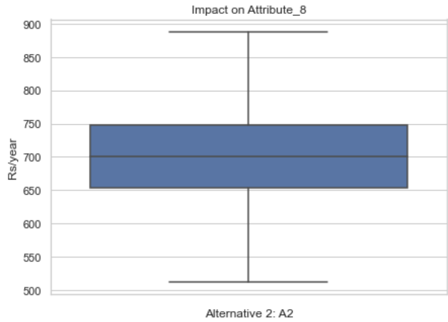


Figure 4.3: A boxplot representation of the impact of alternative 2 in a normal distribution: $\mu = 700$, $\sigma = 70$

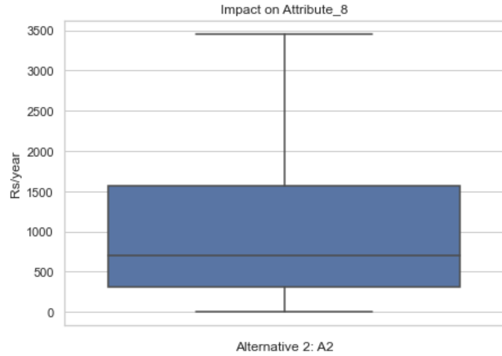


Figure 4.4: A boxplot representation of the impact of alternative 2 in a log-normal distribution: $\mu = 700$, $\sigma = 1.2$

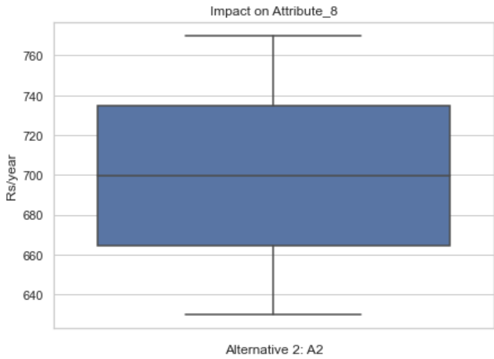


Figure 4.5: A boxplot representation of the impact of alternative 2 in a uniform distribution: $low(a) = 630$, $high(b) = 770$

When the probability density functions (PDF) of Alternatives 2, 5 and 7 are plotted one sees immediately the joint distribution function result for Portfolio 46: $f_{46}(\xi_2, \xi_5, \xi_7)$ in Figure 4.7.

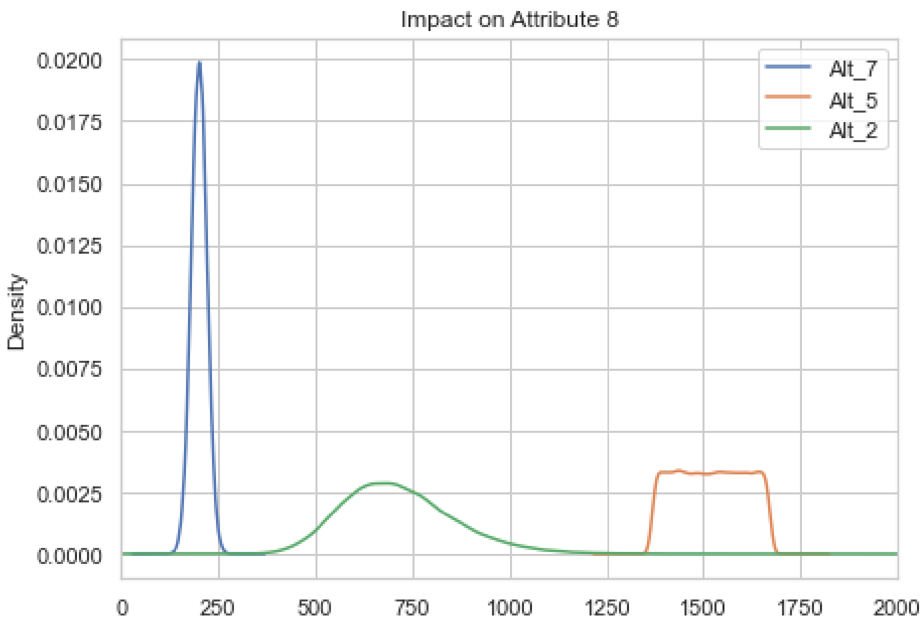


Figure 4.6: PDFs of 3 alternatives. Alternative 5 is a uniform distribution ($low(a) = 1368$, $high(b) = 1672$), whilst alternatives 2 is log-normally distributed ($\mu = 700$ and $\sigma = 0.2$) and alternative 7 is a normal distribution ($\mu = 200$ and $\sigma = 20$).

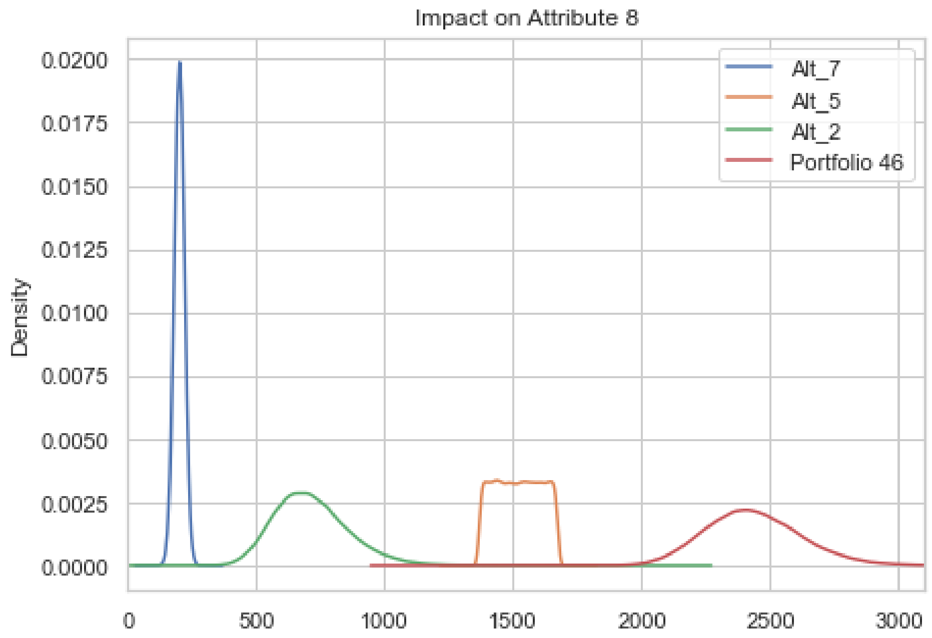


Figure 4.7: PDFs of 3 alternatives combined obtains the PDF of Portfolio 46 = {2,5,7}.

Table 4.1: The expected marginal values for 8 portfolios on 11 attributes obtained from the simulated data set

	Attribute 1	Attribute 2	Attribute 3	Attribute 4	Attribute 5	Attribute 6	Attribute 7	Attribute 8	Attribute 9	Attribute 10	Attribute 11
0	0.000000	0.000000	0.000915	0.090080	0.086332	0.793499	0.77220	0.972530	0.000000	0.000000	0.612744
1	0.027161	0.071212	0.000914	0.321432	0.077601	0.720662	0.86332	0.883005	0.000000	0.000000	0.558655
2	0.036214	0.094949	0.085409	0.269693	0.068776	0.793499	0.88610	0.908584	0.016050	0.000000	0.612744
3	0.271608	0.178029	0.127656	0.294201	0.104074	0.574987	1.00000	0.716745	0.641983	0.009416	0.342301
4	0.316876	0.284846	0.085409	0.089971	0.086332	0.574987	0.97722	0.778134	0.040124	0.235394	0.558655
5	0.036214	0.056969	0.093858	0.117201	0.086332	0.749796	0.79498	0.889400	0.000000	0.000000	0.558655
6	0.000000	0.000000	0.000914	0.089971	0.086332	0.939326	1.00000	0.946951	0.000000	0.000000	0.612744
7	0.045268	0.077146	0.148780	0.089971	0.086332	0.793499	0.77220	0.991694	0.008025	0.282472	0.612744

4.3. Preference Uncertainty

Now the results on how to address the preference uncertainties.

Weight Space analysis of complete ordinal rankings

When the SMAA-O analysis is run for the ordinal ranking of actor X and Actor Y this obtains the following outcomes.

Table 4.2: Portfolios in rank 1 for Actor X

Strategic Portfolio	Acceptibility Index	rank
76	0.23793	rank 1
114	0.48046	rank 1
118	0.06885	rank 1
137	0.21276	rank 1

Table 4.3: Portfolios in rank 2 for Actor X

Strategic Portfolio	Acceptibility Index	rank
35	0.20841	rank 2
72	0.28258	rank 2
76	0.02815	rank 2
114	0.19846	rank 2
118	0.04382	rank 2
137	0.23858	rank 2

From the rankings the top performing portfolios are obtained for the top 10 for both actors:

Table 4.4: Ranked portfolios for Actor Y

Strategic Portfolio	Acceptibility Index	rank
118	1.00000	rank 1
132	1.00000	rank 2
116	0.69710	rank 3
133	0.30290	rank 3
116	0.30290	rank 4
127	0.51867	rank 4
133	0.17843	rank 4

Table 4.5: Top performing portfolios for Actor X from SMAA-O analysis

	Strategic_Portfolio
0	76
1	114
2	118
3	137
4	35
5	72
6	60
7	113
8	115
9	132

Table 4.6: Top performing portfolios for Actor Y from SMAA-O analysis

	Strategic_Portfolio
0	118
1	132
2	116
3	133
4	127
5	117
6	99
7	128
8	131
9	77

However, what if the information on the attribute weights is not complete? if only the most important two or three attributes are identified? Then the set is incomplete and thus information is even more scarce. A SMAA-O approach could technically be taken, letting loose all the lower rank restrictions; increasing the possible sub-space from which the weight vector can be sampled.

The other way to deal with this lacking information is to use frugal heuristics.

Case Study SMAA-O analysis

Table 4.7: SMAA-O outcomes: NGO Farmers

Strategic Portfolio	Acceptibility Index	rank
672	0.9244	rank 1
673	0.0756	rank 1
645	0.3610	rank 2
672	0.0756	rank 2
673	0.5634	rank 2
645	0.5634	rank 3
652	0.2519	rank 3
657	0.0756	rank 3
673	0.1091	rank 3
560	0.0552	rank 4
645	0.0756	rank 4
652	0.3594	rank 4
657	0.3131	rank 4
673	0.1967	rank 4

Table 4.8: SMAA-O outcomes: NGO Tanners

Strategic Portfolio	Acceptibility Index	rank
618	0.1119	rank 1
667	0.7952	rank 1
669	0.0929	rank 1
618	0.7771	rank 2
631	0.0747	rank 2
667	0.1301	rank 2
669	0.0181	rank 2
489	0.1119	rank 3
618	0.0181	rank 3
625	0.4002	rank 3
631	0.0182	rank 3
667	0.0747	rank 3
668	0.2981	rank 3
669	0.0788	rank 3

Table 4.9: SMAA-O outcomes: Research Institute

	Strategic Portfolio	Acceptibility Index	rank
0	629	0.22214	rank 1
1	668	0.77786	rank 1
0	619	0.08025	rank 2
1	629	0.69761	rank 2
2	668	0.22214	rank 2
0	501	0.22214	rank 3
1	619	0.69761	rank 3
2	629	0.08025	rank 3
0	501	0.77786	rank 4
1	619	0.22214	rank 4

Table 4.10: SMAA-O outcomes: Governmental body

Strategic Portfolio	Acceptibility Index	rank
666	0.5925	rank 1
668	0.2835	rank 1
669	0.1240	rank 1
614	0.3045	rank 2
666	0.4075	rank 2
668	0.1969	rank 2
669	0.0911	rank 2
614	0.2432	rank 3
619	0.2835	rank 3
631	0.1240	rank 3
668	0.1727	rank 3
669	0.1766	rank 3

Take-the-best Heuristics

For the ordinal preference rankings of Actor X and Y this yield for their most preferred attributes.

Table 4.11: Ranking of Portfolios on the most important attribute for actor X, which is Attribute 8

Rank	Portfolios
1	8
2	1
3	33
4	7
5	21
6	3
7	31
8	60
9	16
10	6

Table 4.12: Ranking of Portfolios on the most important attribute for actor Y, which is Attribute 1

Rank	Portfolios
1	132
2	127
3	118
4	116
5	108
6	136
7	99
8	133
9	89
10	97

4.3.1. Case Study Take-The-Best

Table 4.13: Ranking of Portfolios on the most important attribute for actor NGO Tanners, which is Time irrigation water demand is met

Rank	Portfolios
1	564
2	575
3	651
4	570
5	649
6	554
7	644
8	577
9	559
10	618

Table 4.14: Ranking of Portfolios on the most important attribute for actor NGO Farmers, which is Attribute Time irrigation water meets reuse standards

Rank	Portfolios
1	655
2	571
3	672
4	645
5	407
6	560
7	652
8	578
9	664
10	613

Table 4.15: Ranking of Portfolios on the most important attribute for actor Research Institute, which is Annualized O&M cost percentage average household

Rank	Portfolios
1	53
2	10
3	61
4	181
5	182
6	30
7	62
8	122
9	57
10	165

Table 4.16: Ranking of Portfolios on the most important attribute for actor Governmental Body, which is Attribute Time without electricity interruptions

Rank	Portfolios
1	673
2	606
3	353
4	579
5	347
6	585
7	586
8	605
9	316
10	634

4.4. Interactions

For the attributes it is determined whether there are combinations in which there is an added benefit. (As already described in a previous subsection 2.2.3 Data.) The results provided here are after the heuristic is applied onto the 10 top performing portfolio SMAA outcomes of Actor X (Table 4.5) and Actor Y (Table 4.5).

Table 4.17: Top performing portfolios for Actor X from SMAA-O analysis with synergies incorporated

Portfolio with Synergies
118
113
132
137

Table 4.18: Top performing portfolios for Actor Y from SMAA-O analysis with synergies incorporated

Portfolio with Synergies
77
116
117
118
127
128
131
132

Every Portfolio has now a distribution of performances on each attribute. These are translated into values using the partial value function. These are than aggregated using the weights. The portfolios are now ranked from best to worst. The interaction value information are added for a top performing set; now a small set of robust, good performing portfolios containing synergy remains.

Shortlists Case Study

From Table 4.20 to Table 4.22 the shortlists of the actors in the case study are presented. The only portfolio that is found in all shortlists is portfolio 668. Portfolio 668 contains Alternatives: 3, 4, 6, 7, 8, 10 and 11.

Table 4.19: Top performing portfolios for actor NGO Tanners from SMAA-O analysis with synergies incorporated

	Portfolios with Synergy
0	489
1	618
2	619
3	625
4	631
5	634
6	644
7	667
8	668
9	669

Table 4.20: Top performing portfolios for actor Farmers NGO from SMAA-O analysis with synergies incorporated

	Portfolios with Synergy
0	560
1	645
2	652
3	655
4	657
5	660
6	664
7	668
8	672
9	673

Table 4.21: Top performing portfolios for actor Research Institute from SMAA-O analysis with synergies incorporated

	Portfolios with Synergy
0	306
1	490
2	501
3	508
4	541
5	619
6	626
7	629
8	636
9	668

Table 4.22: Top performing portfolios for actor Governmental Body from SMAA-O analysis with synergies incorporated

	Portfolios with Synergy
0	614
1	617
2	619
3	626
4	631
5	634
6	666
7	667
8	668
9	669

4.5. Multi-Actor Portfolio Selection

For the results of the simulated data set, see Tables 4.5 and 4.6 for the SMAA outcomes and Tables 4.17 and 4.18 after the incorporation of the synergy information. In both sets only Portfolios 132 and 118 are shared.

For alternative x^j in portfolio p, the alternative occurrences are counted.

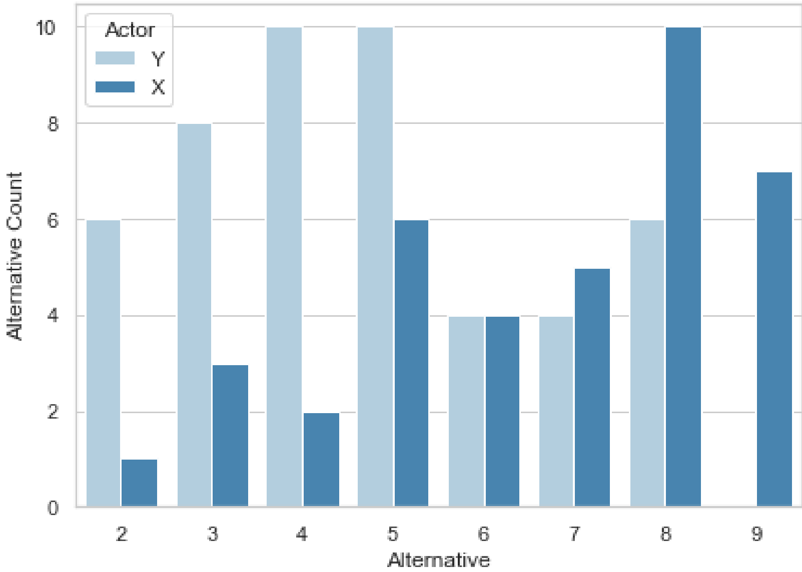


Figure 4.8: Counting the occurrence of projects x^j in the top 10 best performing portfolios for actor X and Y.

For the simulated data this yield the core indices shown in Table 4.23. For overview the "Combined CI" is the multiplication of CI of actor Y and X.

Core Alternative	Actor Y	Actor X	Combined CI
4	1.0	0.2	0.20
5	1.0	0.6	0.60
3	0.8	0.3	0.24
2	0.6	0.1	0.06
8	0.6	1.0	0.60
6	0.4	0.4	0.16
7	0.4	0.5	0.20
9	0.0	0.7	0.00

Table 4.23: Example of Core Index Values for Actor X and Y for the top 10 performing portfolios

From the simulated data used the recommendations delivers a shortlist of portfolios for each actor, Tables 4.17 and 4.18. A starting point of alignment would be portfolios 118 and 132.

Looking at the alternatives in the top performing sets, P_N, alternatives 5 and 8 score a relatively high CI of (1, .6) and (0.6, 1) respectively.

Table 4.25: Composition of Portfolio Set 118

Alternative	Portfolio_set
2	118
3	118
4	118
5	118
8	118

Table 4.26: Composition of Portfolio Set 132

Alternative	Portfolio_set
3	132
4	132
5	132
6	132
8	132

Table 4.24: Heatmap of Core Index values for actor Y and X

The alternatives 5 and 8 can also be found in the top performing portfolios 132 and 118, which is perhaps not surprising. This reinforces the recommendation to look into these portfolios. For example portfolios 118 and 132:

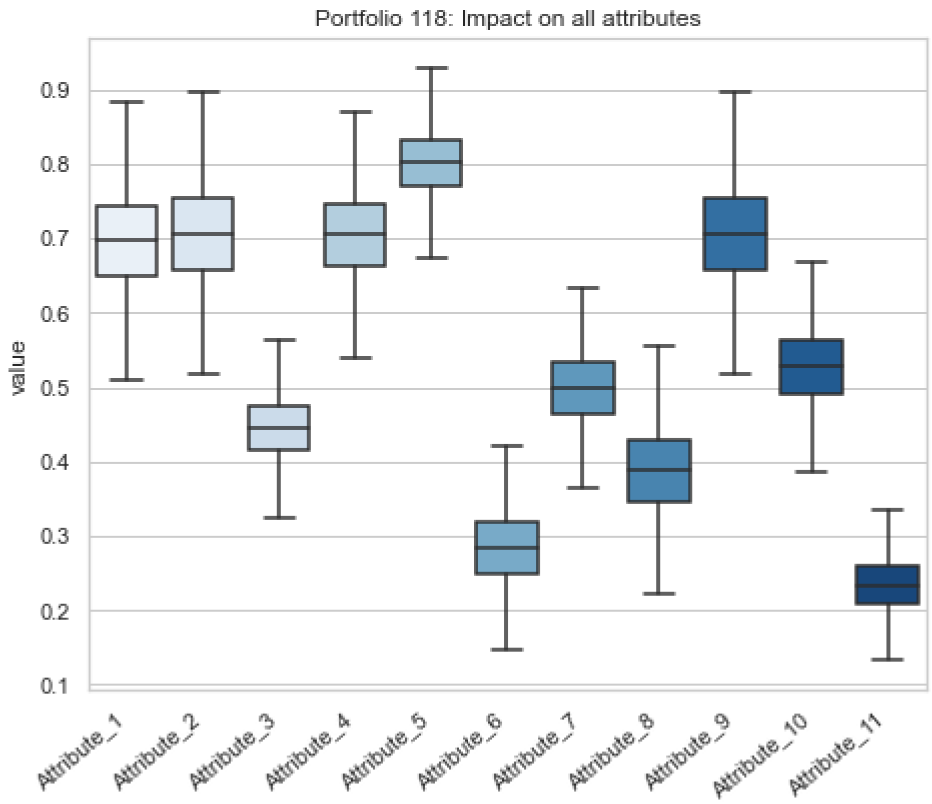


Figure 4.9: Value Distributions of Portfolio 118 across the attributes

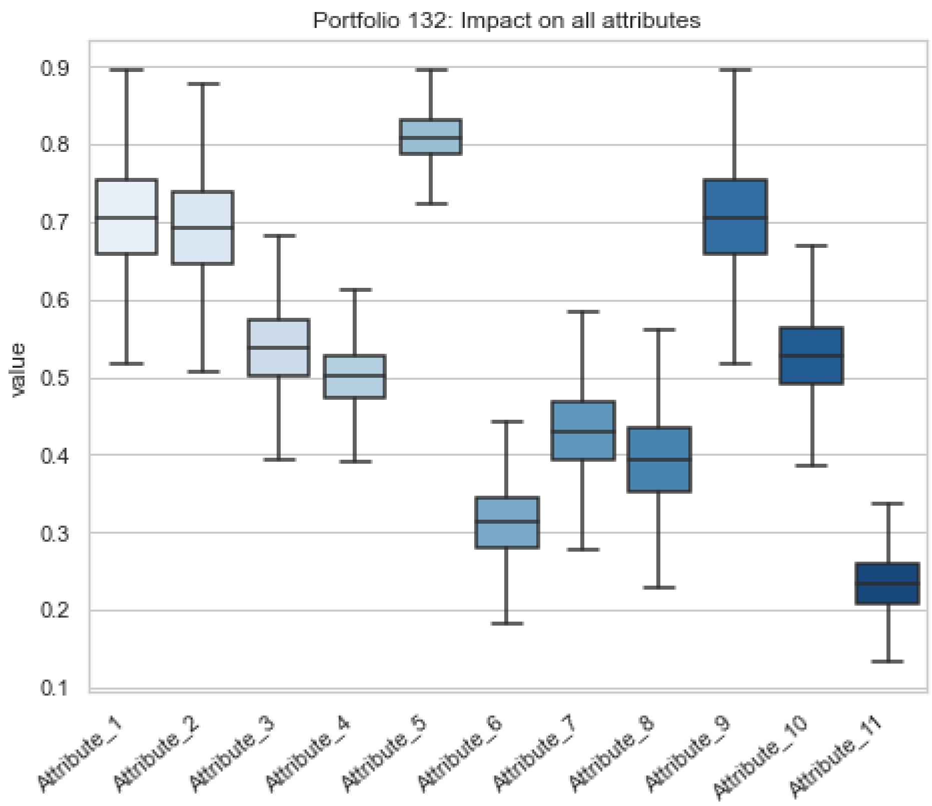


Figure 4.10: Value Distributions of Portfolio 132 across the attributes

4.5.1. Case Study Core Indices

As can be seen in the heatmap resulting from the core index analysis of the actors for the top 10 best performing portfolios there is some overlap, see Table 4.27. Especially alternative 7 (Semi-centralized Self-forming dynamic MBR) is found to have high CI for all the actors. Alternatives 6 (Separate conveyance system for treated non-potable WW) and 3 (Vegetable tanning) are also alternatives that are found in the top performing portfolios of all the actors. Alternative 12 (Relocation tannery sector) is not so much preferred.

Table 4.27: Heatmap of Core Index values for the Actors of the Case Study

4.6. Sensitivity Analysis

For the most promising portfolio 118 the Sobol indices are are shown in Figure 4.11.

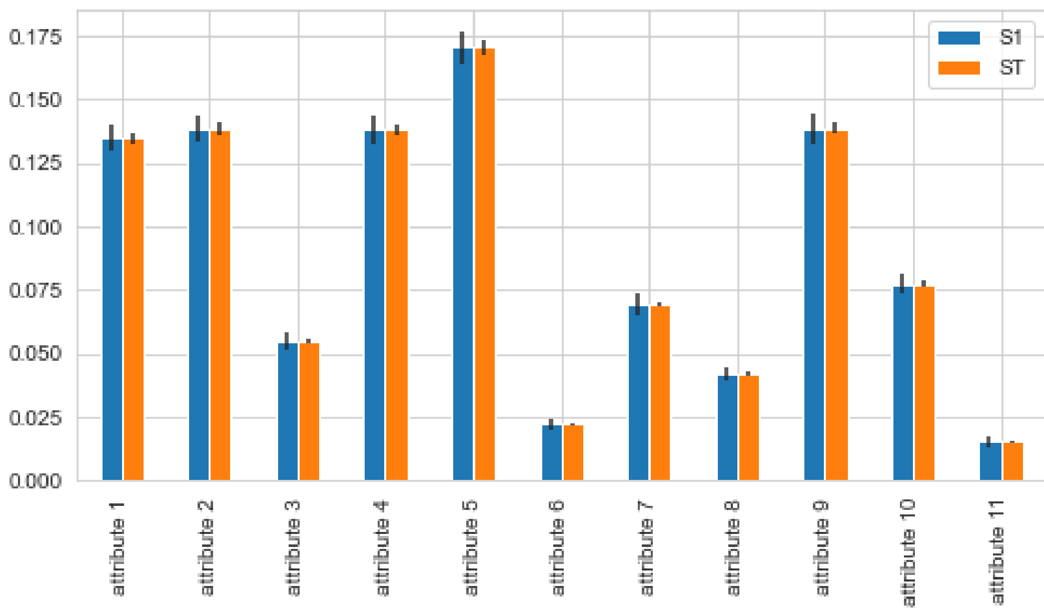


Figure 4.11: Impact indices of the attributes on the valuation of portfolio 118

It can be seen that the attributes that matter most differ per portfolio. Attribute 5 is most important in these two portfolios and Attribute 11 least important and also attributes 3, 6 and 8 are less important. Elicitation efforts if deemed necessary should be on attributes 1, 2, 4, 5 and 9.

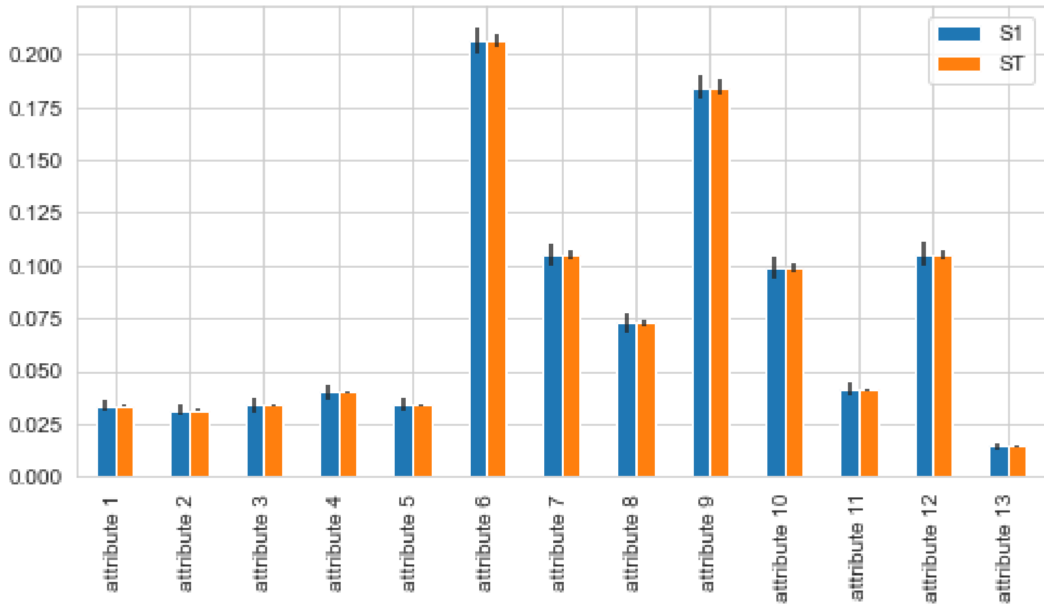


Figure 4.12: Impact indices of the attributes on the valuation of portfolio 668

Regarding the outcomes of the sensitivity analysis on portfolio 668 from the case study, see Figure 4.12, attributes 6 (Profits from hides processing), 9 (Annualized Investment cost percentage average household) and 12 (Time without electricity interruptions) have the most impact. Further elicitation efforts should probably be done on these attributes.

5 Discussion

Some of the choices made and result outcomes deserve some discussion. The structure follows the sections from the Results chapter.

5.1. Portfolio Creation

First the findings on the question what are the challenges and suitable ways to create the feasible portfolios are discussed. To create portfolios from a set of predefined alternatives, including logical constraints, there are roughly three approaches:

1. Naturalistic Generation
2. Using a Greedy Algorithm
3. Covering Entire Solution Space

The last approach to generate all possible portfolios is chosen. This however, has its own limitations. The main difficulty of creating the entire solution space is that it can become too computationally expensive. The main factor is the amount of alternatives from which the portfolios are generated, the other important factor are the constraints that limit the size of the solution space and thirdly the amount of samples in the SMAA-O analysis step. For lower amounts of alternatives the generation process is very fast. However, increasingly more alternatives will take increasingly more time, see Figure 4.1 Whether the runtime is acceptable depends on how many alternatives are available, what the logical constraints are and how often one has to adapt the input data. It is not rare to find case studies containing 20 to 30 alternatives and with the current set-up it is probably not reasonably applicable to situations with more than 15 alternatives on similar hardware systems.

There might be some ways to improve the generation speed of the portfolios by improving the software. There are quite a lot of for loops in the code Class for "Data Acquisition" which slows down the collection of the portfolio data. However, vectorisation does have its own data limitations in the data-size of arrays the RAM can process simultaneously. This mainly depends on the hardware and is not an limiting factor of the method per se.

If a case is encountered in which the entire solution space cannot be covered, one can also look into suitable naturalistic or greedy algorithms to create the portfolios. However, the use of these have their own drawbacks, which were the main choices to opt for the current approach.

5.1.1. Other Creation Strategies

The first approach to create portfolios is using a greedy algorithm, Durbach et al. (2020) used this approach in their paper. Using a heuristic selection procedure the algorithm selects the first alternative in the portfolio. Then a second and so forth, until a certain stopping rule is met. This is often a budget constraint, but it can be anything. For high amounts of alternatives this can be a good approach. The large drawback of this method is the possibility of path-dependencies when only part of the solution space is generated in which a greedy algorithm starts with certain good performing alternatives.

Furthermore, the difficulty of using greedy algorithms in a multi-actor situation is the need to generate the solution set. Be aware that the solution set is a subset of the solution space, which per design is not entirely covered. The drawback is thus when the portfolio analysis is done the sets have to be generated multiple times. Furthermore, to compare the portfolios across actors it might be that the relevant portfolio is not even generated for that particular actor due the initial path selection of the greedy algorithm. Furthermore portfolios that operate well might be missed due to path-dependencies. Such greedy algorithms work often via linear-additive portfolios or solving the 'knapsack' problem. For an elaborate explanation of linear-additive portfolios, see Appendix A.

Similarly the naturalistic creation of the portfolios might lead to excluding good performing portfolios. This approach encompasses workshops of experts of DM's that themselves compose a limited amount of portfolios from the set of available alternatives. So, there is no formal structure to generate the portfolios in this instance. Here the cause is not so much the path-dependency but the plethora of biases that might influence the portfolio creation. For example the existence of championing projects/alternatives. For an overview of biases see chapter 7 of *Portfolio decision analysis: Improved methods for resource allocation* (Salo et al., 2011).

Interestingly, much research applies the greedy-algorithm as a way to handle larger alternative sets (Durbach et al., 2020; Stummer and Heidenberger, 2003; Golabi et al., 1981). Furthermore, research indicates the importance of applying developed methods utilizing such creation methods (Liesiö et al., 2008). The developed method here deviates from this (anno 2022) because the improvements made to computing power and improved (access) to software allow the entire solution space to be generated for many situations. So in order to prevent either behavioural biases or path-dependencies in the search strategy it is opted for the creation of the entire solution space. Using this strategy one should be aware to cover the mutual exclusivities and dependencies as part of the alternative specification.

5.1.2. Constraints as Decision Variables

Another benefit is that the constraint of the greedy algorithm can be incorporated as an attribute/decision variable in the value function. The benefit of incorporating costs as a value function/attribute in the portfolio selection is the greater detail of information that can be handled. (Liesiö et al., 2008; Phillips and Bana e Costa, 2007) The incorporation as a separate attribute allows for the 'added value' that cheaper is better within the set of all affordable portfolios. Information that is not taken into account with the external constraint rule, see Figure 5.1.

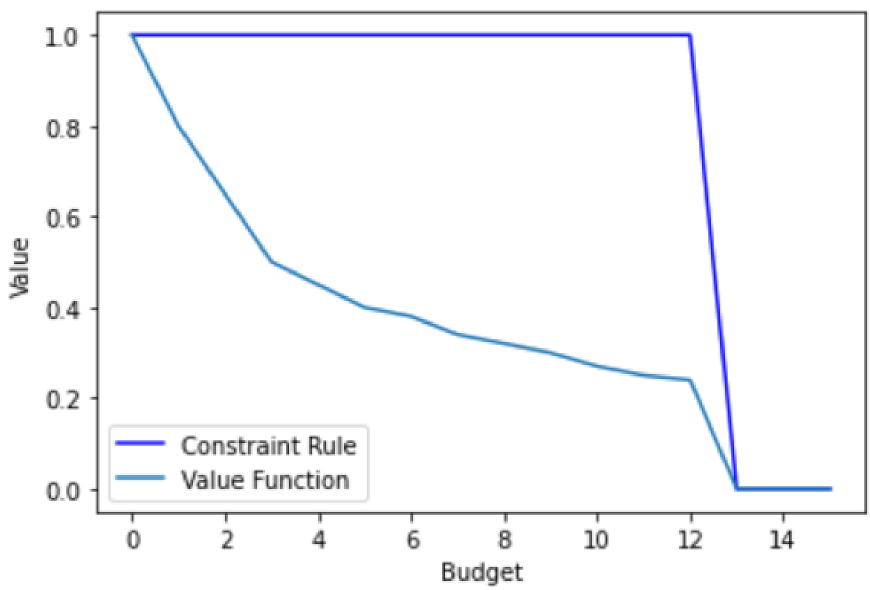


Figure 5.1: Value multiplier example to portfolios for different budget-levels comparing the two approaches; for graphical purposes 0 is lower limit, but can also be infinitely negative.

Currently when looking at the result, this constraint as decision analysis is not really applied to the fullest extend as possible. Currently local scales are applied instead of global. This leads to the valuation of the most expensive portfolio a 0. This lack of value on this 'attribute' can be compensated by another attribute. However, if careful elicitation on the resources were available, this information would yield global scales with possible large negative values. Then the attribute would behave more as a constraint. As this is currently not the case the top performing portfolios are often portfolios with many alternatives. The cumulative added value of multiple attributes outweighs the one or two cost attributes. This possibly does not reflect the real valuation of the cost attribute. A suggestion is to ask for a maximum expenditure (or other cost attributes with hard constraints) to set a boundary level within the value function after which the value becomes infinitely negative.

5.1.3. Linear Additivity of Alternative performances

On the matter of portfolio performances on the individual attributes it has been assumed that the linear aggregation of the alternative performances reflects the performances of the portfolios on that attribute (initially, before synergies are taken into account). However, there is the matter of non-linearities, that needs to be discussed as part of this portfolio creation step. In the current PDA field interactions and the additivity function are treated in roughly two different ways (Liesiö et al., 2021). Either the linearity assumption, as in this thesis, is upheld, or non-linear aggregation functions are used. The first approach uses dummy variables to trigger a superadditive operation and the other incorporates synergies via non-linear aggregation functions. For example of multiplication via a gamma table see (Kantu, 2021).

Regarding this aggregation much research assumes the alternative (synergy) performances data to be readily available. The focus lies on the processing of this data, which is also highlighted by Lieslo et al. (2021). As already mentioned in the Introduction chapter, this can be problematic. Since the information that studies use as input is often not available, this begs the question on the usefulness of the method in practice. Therefore, the approach in this thesis is taken to use lexicographic data, which is qualitative in nature. This, is expected, to be much easier to obtain. Qualitative information on the synergies does not provide sufficient information to develop non-linear value functions. Perhaps this is a point for further consideration or research? Thus, at one hand the approach of this thesis belongs to the group that takes the superadditive approach whilst at the other hand it is vastly different due to the qualitative data the method uses. In contrast, for example, to research that uses qualitative synergy data from interviews with experts (Chowdhury & Quaddus, 2015). See the section 5.6 for the discussion on the incorporation of this qualitative synergy information.

Next to synergy information there is the challenge of certain types of performance data that cannot be aggregated linearly. These are attributes expressed in percentages, for example. For example, a certain alternative reduces pollution with 20% and another reduces the pollution to the base case with 10%. Linear additive aggregation would state that the total reduction would be 30%. However, this is not the case: If pollution from the start is 100, then a first 20% reduction means a level of 80. Then a further reduction of 10% would lead to a level of 72. Similarly, the aggregated 30% would lead to a reduction to a level of 70. A clear difference. The assumption of linearity does not hold in the particular case. The assumption that the order of the alternatives do not matter *does* hold however. Then how to deal with this? The first option would be to convert all individual percentages of the alternatives to absolute numbers. This would probably mean to change the attribute. However, this might not always be possible since information might only be accessible or estimated through percentages. The benefit of using percentages is that a quantification of current absolute numbers does not have to be available. The other solution to this problem is to calculate them properly. This means that another equation (eq. 5.1) has to be used to aggregate ratios or percentages. Thus, the linear additivity aggregation is not upheld. For m alternatives in portfolio, i, expressed in percentages, the value for that particular portfolio on a percentage attribute becomes:

$$x_{bi}(P_i) = \sum_{j=1}^{m} A_j + \prod_{j=1}^{m} A_j \; for \; A \in P_i \tag{5.1}$$

Therefore the code checks specifically for percentages attributes and aggregate them using this function. Similarly, there might be proxy attributes for which the linear property does not hold also.

This depends on the situation and is something to reflect on before creating the portfolios. I suspect that for most natural attributes the linear aggregations assumption does hold indeed. At least for the Case Study attributes, see (Saharan et al., In preparation). The attributes are assumed to be able to be aggregated linearly (other than those attributes expressed in percentages). Any non-linearities that arise are expected to emerge/caused from the synergies between alternatives interactions and not from the attribute metric/unit used themselves inherently.

Another aspect that arises on the attribute performances are the scales. It might be needed to specify maximum boundaries; are the outcomes allowed to exceed the [0, 100] range? Can there be negative values or larger than 100% values? If so, then the base values have to be capped. Again, this depends on the situation and needs to be reflected upon already in the Problem Structuring and Elicitation.

From these considerations regarding aggregation the portfolios are created. Let us now discuss the uncertainties of these values.

5.2. Performance Uncertainty

As already mentioned in the introduction there are several types of uncertainties. The uncertainty regarding the structuring phase falls outside the scope of this thesis. Therefore first the matter of performances of the alternatives and portfolios on the attributes is treated.

Part of the uncertainty of the portfolio performance stems from the uncertainty on the acceptability of the linear addition of the individual alternatives, see the previous subsection the treatment of this aspect as part of the portfolio creation process. Next to this matter is the question how to model and address the alternative performances that compose the portfolios.

Regarding the alternative performance often ranges are given in the form of +/- range, noting a uniform distribution. However, there is possibly much detailed information behind the possible ranges. Alternative performances are treated as distributions that do include this information.

The propagation of the uncertainties to portfolios done via Monte Carlo simulation is already an established approach (Papadopoulos and Yeung, 2001; Scholten et al., 2015; Gatian and Mavris, 2015; Lahdelma et al., 2003). The propagation of the uncertainty distributions (within the discussed linear framework) is capable of dealing with the different distributions as can be seen in Figure 4.7. The amount of Monte Carlo samples ($n = 1 * 10^5$) is more than sufficiently for a sufficient approximation. As in Monte Carlo simulation the required number of iterations is inversely proportional to the square of the desired accuracy; Interestingly, MC does not depend on the dimensionality of the problem significantly (Fishman, 1996). Therefore, Monte Carlo simulation can be used to obtain a distribution of portfolio performances with moderate effort.

From this the uncertainty distributions and corresponding E[X] are accepted as such. The shape of the uncertainty can vary much whilst the E[X] can remain fixed. So the information of the uncertainty range in itself is not quantifiable taken into account in the calculation of performance! For incorporation of (uncertain) risk/utility preferences in situations where this information is available I refer to the many possible approaches to deal with this uncertainty such as by Haag et al. 2019 into Expected Expected Utility (EEU) or Scholten et al. 2015 into Expected Utility (EU).

This might mean that the portfolios generated as the 'best', can have a very large uncertainty range. For example, it might be that there is a portfolio with a lower expected value, E[X], but with a much higher certainty (a smaller uncertainty range). The outcomes now does leave this question of risk attitudes numerically in the open, however that does not mean these are ignored. The probability density functions and current uncertainties are explicitly shown to the actors in boxplot or graphical visualisation, see for example Figures 4.9 and 4.10. The explicit visualisation allows actors, with sufficient understanding, to integrate this explicit uncertainty in their decision-making when multiple suitable portfolios emerge. Generating this information on the uncertainty distribution over the attributes can help together with the sensitivity analysis to identify attributes that might need more focus to reduce the uncertainty of the portfolio on that particular attribute.

The current method thus deals with uncertainty explicitly and flexible. Its only drawback is that it also relies on information on the distributions.

5.3. Preference Uncertainty

Next to the attribute predictions are the preference uncertainties. As mentioned in the introduction these encompass the marginal value functions, the hierarchical aggregation function and the aggregation parameters.

In this study the only preference information that is deemed available is an ordinal attribute ranking obtained from the actors. Therefore the focus is on the aggregation parameters. Other research has been done on the other types of (preference)uncertainties and how to deal with those, for example on the hierarchical aggregation function. As already stated in the methods chapter, a hierarchical linear aggregation function has been assumed. There is thus the possibility that other forms of aggregation would reflect the preferences of the decisionmakers better. I refer, for example, to the research of Scholten et al. 2015, see section 2.4.2, for an uncertainty analysis into the extend of upholding linear value additivity properties in MCDA when this information is unavailable.

Similarly, for the valuation of the attribute performances via the partial value functions linearity is assumed. This assumption is due to it being a reasonable approximation for monotonic behaviour in situations where elicitation is not possible. This behaviour is accepted due to the nature of the attributes; If there are certain attributes, such as optimal temperature, for which monotonic behaviour is probably non upheld, effort should be given to elicit a detailed value function. One could test the outcomes by researching the impact of altering value functions on the same dataset, see for example Chacon-Hurtado and Scholten (2021). This would be point for further research.

The preference uncertainty treated and analysed is the uncertainty on the attribute weights. This focus on ordinal weights and very little information is in contrast to much other research that study uncertain preference information. Much current research focuses on the treatment of uncertainties when a cardinal set of attribute weights have been obtained.

Let us first discuss the outcomes of the SMAA-O approach and then the results of the Take-The-Best-Heuristics.

5.4. SMAA-O

The outcomes of a SMAA-O analysis are shown in Tables 4.2, 4.3 and 4.4 for the simulated dataset. For SMAA-O there are two index values that are particular interesting. If a value is $b_i^r = 0$ then a portfolio will never achieve that particular rank. In contrast if $b_i^r = 1$ than a portfolio will always achieve that particular rank no matter the weights, see Table 4.4. This is particular interesting for the best ranks. Furthermore, the Acceptability Index within each rank sums to 1 and the acceptability index for one portfolio over the ranks also sums up to 1. These can be seen as the percentages of a weight subspace that is dominated by the portfolio in that rank. From these ranks, one can select the best performing portfolios either by selecting the x ranks one wants to study or a x amount of portfolios that perform best. Here portfolios 76, 114, 118, 137 are the dominating portfolios within the weight space constrained by the ordinal ranking of actor X. Portfolio 114 can be said to be the best with a 48.046% accuracy; which is the most robust outcome.

Similarly for the Case Study the results are shown in Tables 4.8 to Table 4.10. The largest drawback of the SMAA approach is its complexity and interpretation of the the results. As already mentioned in the subsection 5.1.2, many portfolios with high numbers are present in the top performing portfolios. These are the portfolios that do contain more alternatives than those with lower numbers. (Due to the way the combinations are made in algorithm). Without any constraints the last created portfolio would contain all alternatives. Also the different preferences over the weight lead to much diversitity in the top performing portfolios, albeit mostly portfolios with many alternatives only portfolio 668 overlaps when studying the outcomes containing the synergy pairs!

5.5. Take-The-Best-Heuristic

The original satisficing and other heuristics have in common that they are often implicitly used by human decision-making. In real life case studies it is thus expected that initial behaviour of the decision makers would follow lines of identified heuristics. As the work of Gigenzer states, this does not have to be a problem (Gigerenzer & Goldstein, 1996). Moreover, as confidence and communication is important in decision support, the heuristics can form paths to communicate more easily than complex approaches. The arguments in support of using heuristics is the specialisation of these approaches to deal with limited information and the intuitive understanding of the participants in the decision process. A major disadvantage is that the outcomes are likely not optimal.

The Take-The-Best-Heuristic ranks the portfolios via the value outcomes on the chosen attribute X. Whilst there might be little difference between top performing portfolios on that particular attribute X there might be on the other attributes. Let us look at the outcomes for actor X for both the SMAA-O approach and the Take-The-Best-Heuristic, see Tables 4.11 and 4.2. As can be seen, there is **barely overlap** when ranking by Attribute 8, a cost attribute, where a lower amount is better. Here only portfolio 60 is found in both top 10 rankings. On the other hand Actor Y has Attribute 1 (benefit attribute) as its most preferred attribute and now there is **significant overlap** between the portfolios of Table 4.4 and 4.12. 6 of the found portfolios in the heuristic is also found in the outcomes of the SMAA-O approach. Attribute 1 is the *second* ranked attribute for Actor X, sorting on this would lead naturally to the same outcome for actor Y.

This behaviour is also seen in the results of the case study. The actor of CLRI has noted a cost attribute as the most important attribute. Consequently, the outcomes of the Take-The-Best Heuristic approach lead to portfolios containing few alternatives, see Table 4.15. There is no overlap in the outcomes of this approach and the SMAA-O approach.

So what leads to this difference in overlap between SMAA-O and heuristics? SMAA-O takes into account both cost and benefit attributes. The large amount of benefit attributes probably dominate the aggregated value function. The costs are offset with the benefits. If a take-the-best heuristic selects using a benefit attribute there is expected to be more overlap in that case because it 'follows' the aggregated value function behaviour.

If the aggregated value function, however, would be dominated by cost attributes it is expected that than there would be more overlap when choosing a cost attribute.

Whether this matters depends on the exact attribute weight preferences and its intervals. The SMAA approach can say with a certain probability the robustness of the portfolio looking at all possible intervals. However, there is an option, which can be very small, that the preference of the particular actor is so strong that the attribute weight of its number one attribute approximates 1 (reducing the other weights to negligence). Than effectively the aggregated value function becomes the corresponding Take-The-Best-Heuristic. So one could say that the Take-The-Best heuristic is one extreme sample from the Weightspace, **W'**.

Therefore it cannot be said that the SMAA-O approach can be a benchmark to test the outcomes of the heuristics approach. However, in the absence of more accurate cardinal weights the SMAA-O approach is probably better due to the robust information is delivers and the information on its chance. This then raises the question whether this Take-The-Best heuristic, containing 1 cue, can be improved upon.

5.5.1. Other Heuristics

The large difference in outcomes suggests that the incorporating of at least multiple attributes in the heuristics approach could be a good approach, especially if cost attributes and benefit attributes are mixed. For example, the stopping rule, which varies per heuristic, can be set aspirational: "When a certain level is reached include". So, if an aspirational heuristic can be agreed upon than an aspirational *satisficing* heuristic might be suitable. The problem with multi-actor situation is that this is often not possible and this increases the needed elicitation efforts not assumed present in the study. One could also surpass the actors and use 'objective' levels as aspirations goals to the extend that it would be a normative exercise based on a set of rules/laws or ideas of what is right. An example of such objective levels are minimum pollution levels as stated by the WHO, the reduction of human casualties in traffic etc. But still these are often balanced in many real life situations. Thus this circumventing of actors to set aspiration levels should not be done for a single attribute in the system. A "no-matter-the-cost" situation is seldom acceptable and attributes reformulated using Willingness to Pay aspects. Thus it could than be done for multiple or all of the identified objectives. As to prevent a complete deterioration on objectives for which no aspiration level is stated; expenditure on health care at one hand and health care quality on the other. However, before one knows it the decision supporter is covering the entire attribute space with aspiration levels returning to the classical rational decision making methods in the form of goal programming.

"The linkage between satisficing and goal programming is clear. All goal programming models contain a set of goal values to be reached. Meeting these goals as closely as possible is the main aim of the goal programme. Thus satisficing can be rightly thought of as the prime underlying philosophy of goal programming" (Jones, Tamiz, et al., 2010; Chapter 1, Section 1.2, page 7)

Furthermore, in the current MAVT framework such aspiration levels could already be incorporated in the value functions for each actor with much more detail, see subsection 5.1.2. Hence, the Take-The-Best-Heuristics should only be applied here in very information scarce environments. One conclusion drawn here is thus that the Take-The-Best heuristic in the studied multi-criteria situation is not preferred. In contrast the SMAA-O approach is capable of generating robust portfolios fast, albeit a somewhat complex approach.

5.6. Incorporating Synergies

From the portfolio lists obtained from the SMAA-O or Take-The-Best Heuristic approach an extra analysis step has been carried out. Here the synergy values are taken into account using qualitative data and a heuristic approach.

Thus integrating the non-linearities into the linear additivity function of the portfolio creation process. This is the most novel step of the methodology so it deserves some explanation and discussion regarding the epistemological properties of the collected synergy data. From the experts information has been elicited on 1) the presence of the synergy between an alternative pair and 2) whether it is a positive synergy or a 'negative' synergy.

Whilst this approach uses the idea of the heuristic **Unit value with Synergy** as developed by Durbach et al. (2020), its application is different (Durbach et al., 2020); It is not part of the selection procedure to *create* the portfolios, but it is an approach added in the end as a way to refine the final top performing portfolios.

Due to the lexicographical nature of the data first two possible situations/interpretations have to be discussed regarding the base values, x_b, and the estimated *benefit* synergy values, x_s^+:

1. Synergy values are higher than the base values, $x_s^+ > x_b$: the highest performing portfolio set, containing this synergy combination should be taken into the best performing portfolio shortlist. The actors or experts can than review these portfolios and determine whether this x_s^+ is so large that it might make an otherwise average portfolio better than the best performing portfolios if only one looks at the x_b.
2. $x_s^+ < x_b$: the selection of best performing portfolios can be fine tuned by selecting from already best ranking portfolios only portfolios with synergies. (This is similar to highest Values with Synergy from Durbach et al.(2020)). In this second situation it is also assumed that the best performing portfolios have similar x_b outcomes in which the existence of a synergy can decide in favour of very similar Portfolios. The difficulty is to judge how much the difference between x_b can be to be 'compensated' by x_s^+; or phrased differently, how similar can they be? There is no definite answer on this matter.

In the methodology chapter the approach implicitly assumes the second situation as the portfolios that are to be checked for 'added' synergies are the top performing based on outcomes of the linear additive function. However, would the first situation be a more accurate interpretation of the obtained qualitative data than this poses somewhat of a limitation. In the current method an average portfolio might not emerge amongst the top performing portfolios, whilst 'having' this synergy.

First, such a situation as described in situation one is quite remarkable and in discussion with experts on the matter this information can arise next to the simple: "Synergy? To be or not to be". In such situations one can choose to combine the alternatives into 1 alternative or explicitly give feedback to the decision analyst. No such situation has been encountered however in the case study. It is also a possible point for further research into synergy elicitation. For now the information that is obtained from experts is whether a synergy exists between pairs. However there is ample room for more in-depth questions here. However, similar biases might emerge in this process/effort to obtain more detailed information as would in the traditional form of portfolio creation by experts.

The amount the shortlists are reduced depend on the amount of synergy pairs discovered that can be present in the top performing portfolios. For the simulated data these are {2, 3}, {2,4} and {5,6}, see Appendix F, Table F.5. The application of the heuristic step is thus successfully incorporated.

However there are some situations for which further decision steps might be decided upon. For example, when almost all combinations create synergies. The system is then very complex and the heuristic approach diminishes in usefulness especially when information remains scarce. One approach would be to try to quantify the synergies on a cardinal scale and incorporate these as highlighted in Appendix A.2. The other approach is try to elicit this more detailed information; rank the synergy pairs ordinal or get some synergy strength indication using swift like approaches. Furthermore the attributes on which the synergies act might also be considered in combination with the attribute preferences. This differentiation of synergy values allows for further decision rules to select preferable sub-sets. Again, this would be a matter of further research into synergy elicitation.

5.6.1. Cost Heuristics

Due to the time constraint the incorporation of cost synergies or "negative" non-linearities has not been applied in the current methodology. However, implementation would be very similar as to the heuristic approach of the benefit heuristic.

Regarding the above outlined two situations to consider for the benefit heuristic only the second one than has to be considered when incorporating cost heuristics; For an average performing portfolio can not be made better by an identified cost synergy interaction.

5.7. Consensus

The heatmaps are a useful way to present the findings. The current outcomes show clearly the alternatives preferred implicitly by the actors. Similarly, when overlap is found this can help the decision process. The current approach allows for quick assessment of the different weights since all the information of set S is unchanged.

Table 4.23 for example, shows that different actors have different core alternatives in their set of top performing portfolios. If for a certain alternative the Combined CI = 1, this would be seen by all involved actors as core project that should be definitely reviewed. To achieve consensus two approaches to the generated portfolio data are proposed. The most straightforward of these is the alignment/overlap of similar portfolios. However, it is expected that with increasingly more actors a satisfying overlap will be less likely achieved. Here the Core Indexes offer a more flexible approach. The portfolios within the $P_N(S)$ sets are decomposed, looking which alternatives make up that portfolio and the alternatives are than analysed.

If an alternative is found in all portfolios in the set $P_N(S)$, than CI = 1. This is what strictly speaking is a *core project*. The other extreme would be a CI = 0, in which a project is not found in any of the top performing portfolios. Most of the alternatives will fall between these two and give additional information whether a certain alternative should be looked into more closely. The alternatives can be than delved into more deeply possibly to reduce external and internal uncertainties is further elicitation efforts.

5.8. Sensitivity Analysis

It should be noted that the second order interactions are absent, which is in correspondence with the hierarchical aggregation function assumed. Hence the Sobol Total indices correspondent to the first-order.

Sensitivity Analysis can be categorized broadly in **local** and **global** sensitivity. Local sensitivity varies one variable within the vicinity of the inputs found, often ten percent. Furthermore often only one input variable whilst keeping the others fixed, so called One-factor-at-a-time (OAT) sensitivity (Saltelli et al., 2008).

To determine sensitivity it is prudent to carry out a global sensitivity analysis (Saltelli et al., 2019). The current examples provide linear properties, both for aggregation of alternatives and of attributes in combination with linear monotonic value functions. However the method drafted should be able to deal with multiplicative functions as well, also non-monotonic value functions should be able to be used. With the current linear-properties of the equations studied no second-order are expected as can also be seen in the same outputs of the S1 and ST indices. This might lead to the suggestion that Sobol analysis would "overdo it". However with the current ease of implementation in Python using the SALib library in combination with the computing power the problems of the past (Generating sufficiently large sample sets) seem to be largely resolved (Herman & Usher, 2017). Thus the implementation does not impede a disadvantage. In contrast if in the future a non-linear aggregated value function (multiplicative etc) is used the negligence of second-order interactions between attributes cannot be justified.

5.9. Final Recommendation

The final recommendation includes the top performing portfolios for each actor, the overlap between portfolios and alternatives in the form of Core Indexes as a starting point for consensus building. One can also compare the difference between the sets when interaction information is included. It also contains the explicit value representations of the portfolio, such as presented in Figures 4.9 and 4.10. Lastly it provides for the these portfolio the insight of the impact of the Sobol analysis. This gives the Actors in the system valuable insight for structured discussion without having to consider all attributes and all possible 1000+ portfolios. It now is a manageable robust amount without the need for extensive modelling of system interactions. Now let us apply the methodology to a real life case study.

6 Conclusion

6.1. Summary

From the studied literature and challenges encountered in real-life case study the answers have been sought to the main research question: "How to create and explore portfolio sets in multi-actor and synergy situations with a lack of time and access to detailed preference information?" This research question was studied in 4 parts via the sub-questions.

1. **What are the challenges and suitable ways to create the feasible portfolios?**

   The main challenge in portfolio creation are the occurrences of biases in creating portfolio sets. To deal with this all possible portfolios using iterative algorithms are generated. The main drawback is computational expense but the current code is able to generate portfolios from a set of 14 alternatives, without any constraints, resulting in 16383 Portfolios in a run-time of 54 minutes running on a 16 GB RAM, Processor: Intel(R) Core(TM) i7-9750H CPU at 2.60GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s). The amount of to-be generated portfolios is reduced by adding logical constraints decreasing the amount of feasible portfolios possibly mitigating some of the computational expenses. The generation of the solution space is not limited by any amount of alternative combinations other than running time.

2. **How to deal with uncertainties in the portfolios and the preference modelling?**

   Uncertainties are categorized in external and internal uncertainties. External uncertainties refer to the expected performances of the alternatives on the system. Internal uncertainties refer to the uncertainty in the preference information obtained from interaction with the actors.

   The external uncertainties of the alternative performances are expressed as probability density functions. The methodology allows for a multitude of distributions adding a layer of detailed information if the information is available. One can also opt for uniform ranges with a large uncertainty range if the information is very limited. This thus allows for much flexibility. Via Monte Carlo sampling the external uncertainties are propagated to the portfolios. The external uncertainties are kept explicit and are visually presented either as probability density functions or via boxplot representation.

   The internal uncertainty efforts are focused on analysing attribute weights preferences. With limited information available complete or incomplete ordinal rankings of the attribute weights impacts on portfolio performance are analysed. This is done via a stochastic multicriteria acceptability analysis (SMAA) and using a Take-The-Best-Heuristics. The outcomes are compared and it is found that the SMAA outcomes are more robust method albeit much more complex. than the outcomes of the Take-The-Best-Heuristic. The outcomes are more robust since the approach takes into account all attributes and generates probability outcomes on the change of that particular outcome. Whilst the Take-The-Best-Heuristics only looks at the most important attribute. The complexity of the SMAA-O method predominantly occurs in the mathematical process of the result in contrast to the heuristic method.

   After the generation of the portfolios a Sobol sensitivity analysis has been carried out to determine which attributes contributes most to the overall values of the selected portfolios. This information helps focusing elicitation efforts in a later stadium on the attributes preferences of those particular sensitive attributes. Due to the additive-nature of the value aggregation function assumed, there have not been found any second-order interactions. The choice for this application is both the ease of implementation in the current python libraries and in light of possible alterations to the methodology of applying non-linear-additive hierarchical value functions. Simultaneously, it is recognised that it can indeed be considered ’overshooting’, nonetheless the outcomes are generated.

3. **How to incorporate interaction information, so-called synergies, where quantitative data is lacking?**

   Sometimes time is scarce and it is uncertain if it would be possible to obtain detailed synergy information, therefore a novel method has been explored to generate useful outcomes nonetheless. Through expert consultation it is often easier to obtain the existence of interactions lexicographically rather than numerically. Using this lexicographic information one knows whether there is a synergy or not. This information is used to filter the already top performing sets of portfolios from the SMAA or Take-The-Best analysis. This ensures the incorporation of synergies in a very rough manner suitable for the generation of shortlists.

4. **How to select good-performing portfolios in the wake of the above mentioned uncertainties in multi-actor situations?**

   For each actor a portfolio shortlist is generated. When there is overlap between the shortlists of portfolios these can be chosen for public discussion. For both the simulated dataset and the case study portfolios were found agreed upon by all actors. However when no consensus is found in the complete portfolios the portfolios in the shortlists are decomposed. This means that the alternatives within the portfolios are studied. The core indices of the alternatives are determined and here a starting point for consensus is found. These highlighted alternatives can also be studied for further specification, adaptation and research to reduce uncertainties on their performance. This helps in determining which alternatives need to be modeled more precisely in order to reduce their performance uncertainties, if this (iterative) data-collection step is desired. Furthermore, a portfolio with high consensus can possibly be formed from the alternatives with a high core index as a starting portfolio framework. The uncertainties of the portfolios are kept explicit as to allow for the actors to internalise the uncertainties in their further decision making.

6.1.1. Main Research Question

Combining these findings the research has found and developed a robust and fast way for executing a PDA. This allows for focusing attention to the portfolios, alternatives and attributes that matter. The flexibility of the method allows for adding and removing additional alternatives and attributes as long as limited information is available: Cardinal data on the alternative performances, cardinal data to capture the alternative performance uncertainties, and an ordinal ranking of the weight preferences. Whereas other research has mainly focused on incorporating available numerical data on synergies and cardinal attribute preference weights using a plethora of methods this study acknowledged the difficulty of obtaining the data itself. The research therefore strived to explore and develop ways to deal with this lack of data using heuristics.

Where the matter of (individual) alternative performance uncertainties can be partially accounted for by incorporating the uncertainty distributions. Synergy data, however, is another matter. The lack of information here posed the question whether it would be possible to analyse anything at all. The first reaction might be to stall this matter, ignoring it or contrariwise trying to model everything extensively. However, this would be time consuming and often expensive. It is novel and daring in the sense that the methodology uses crude quantitative data instead of qualitative data. Furthermore, the blending of mathematically dense operations and heuristic incorporation of quantitative data might be considered unorthodox. The information obtained from the analysis for the decision makers must not be seen as an end point but as a starting point. If the actors are knowledgeable and confident enough they maybe do not need extensive modelling information from a third party (the analyst). And if they do need the extensive knowledge, they now know where to start. This prevents wasting time and resources on analysing alternatives/projects that for nobody performances sufficiently. So there is much potential to the use of synergy heuristics here. Interestingly to deal with the preference uncertainty on the attribute weights I found the analytical SMAA method more convincing than the Take-The-Best Heuristic.

The code to run the analysis can be found on GitHub and in the corresponding Appendix and is free to use. The hope is that portfolio analysis is used as a framework for conducting structured decision support in similar case studies. Above all else, the method shows that PDA is possible, even in complex, messy and time limited situations, such as the case study described.

6.1.2. Limitations

This thesis has explored the use of PDA in a preference information scarce situation. This made necessary some of the assumptions described in the methods and discussion chapter. However, therefore, the current findings apply only to a subset of the problem structures. In effect this means that when properties of linearity are not upheld, considerations are needed whether to adapt the current method if one want to apply it. The two most important aspects where this occurs are firstly in the linear-additivity property of the *generation of the portfolios*. Secondly it occurs at the aggregation of the partial value functions to achieve a value distribution per portfolio used for ranking; the hierarchical aggregation function. In the discussion chapter I refer to research that explores these properties in depth. In addition, the method applies linear marginal value functions. In situations where non-monotonic valuation of attributes is expected the linear approximation is not upheld, the value functions on those attributes have to be considered more in-depth.

6.2. Further Research

For further research the methodology still has some room for exploration and improvement as is listed below:

- Improve the code in order to generate Portfolio faster: There is room for factorization to replace the many for loops.
- Exploration of the marginal value functions in portfolio ranking by exploring the value function areas for different levels of concavity or convexity. This would result into a sort of "Value Space/Distribution", similar to the SMAA-0's weight space.
- Explore the impacts of different lexicographic heuristics, the application of lexicographic heuristics in PDA is quite unexplored.
- Synergy elicitation is limited and exploration to more detailed ways to incorporate synergy information can be conducted.
- Develop approaches to incorporate negative interactions via heuristics.
- The study of compensatory aspects for example by exploring how different hierarchical value aggregation functions impact shortlist selection.
- Incorporate future scenarios analysis into the alternative performances increasing the robustness of the outcomes.

References

Ahmed, S., & Ismail, S. (2018). Water pollution and its sources, effects & management: A case study of Delhi. *Shahid Ahmed and Saba Ismail (2018)'Water Pollution and its Sources, Effects & Management: A Case Study of Delhi', International Journal of Current Advanced Research*, *7*(2), 10436–10442. https://ssrn.com/abstract=3145289

Aloysius, J. A., Davis, F. D., Wilson, D. D., Ross Taylor, A., & Kottemann, J. E. (2006). User acceptance of multi-criteria decision support systems: The impact of preference elicitation techniques. *European Journal of Operational Research*, *169*(1), 273–285. https://doi.org/10.1016/j.ejor.2004.05.031

Artinger, F., Petersen, M., Gigerenzer, G., & Weibler, J. (2015). Heuristics as adaptive decision strategies in management. *Journal of Organizational Behavior*, *36*(S1), S33–S52. https://doi.org/10.1002/job.1950

Azzini, I., Mara, T. A., & Rosati, R. (2021). Comparison of two sets of monte carlo estimators of sobol'indices. *Environmental Modelling & Software*, *144*, 105167. https://doi.org/10.1016/j.envsoft.2021.105167

Bassi, N., Suresh Babu, S., & Kumar, M. D. (2019). Chapter 9 - Water accounting for Kanpur metropolitan area in Ganges river basin. In M. D. Kumar, V. R. Reddy, & A. James (Eds.), *From catchment management to managing river basins* (pp. 245–274). Elsevier. https://doi.org/10.1016/B978-0-12-814851-8.00009-4

Baumol, W. J. (1961). Management models and industrial applications of linear programming, volume i, by Abraham Charnes and William W. Cooper. https://doi.org/10.1002/nav.3800090109

Belton, V., & Stewart, T. (2002). *Multiple criteria decision analysis: An integrated approach*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4615-1495-4

Buchanan, J. T. (1994). An Experimental Evaluation of Interactive MCDM Methods and the Decision Making Process. *Journal of the Operational Research Society*, *45*(9), 1050–1059. https://doi.org/10.1057/jors.1994.170

Central Pollution Control Board. (2021). *Report by CPCB in o.a. no. 985 of 2019*. National Green Tribunal. https://greentribunal.gov.in/report-cpcb-oa-no-985-2019-re-water-pollution-tanneries-jajmau-kanpur-uttar-pradesh-986-2019-titled

Chacon-Hurtado, J., & Scholten, L. (2021). Decisi-o-rama: An open-source Python library for multi-attribute value/utility decision analysis. *Environmental Modelling & Software*, *135*, 104890. https://doi.org/10.1016/j.envsoft.2020.104890

Chakraborty, S., & Yeh, C.-H. (2009). A simulation comparison of normalization procedures for TOPSIS. *2009 International Conference on Computers Industrial Engineering*, 1815–1820. https://doi.org/10.1109/ICCIE.2009.5223811

Chaplin, S. E. (2011). Indian cities, sanitation and the state: The politics of the failure to provide. *Environment and Urbanization*, *23*(1), 57–70. https://doi.org/10.1177/0956247810396277

Charnetski, J. R., & Soland, R. M. (1978). Multiple-attribute decision making with partial information: The comparative hypervolume criterion. *Naval Research Logistics Quarterly*, *25*(2), 279–288.

Chowdhury, M. M. H., & Quaddus, M. A. (2015). A multiple objective optimization based QFD approach for efficient resilient strategies to mitigate supply chain vulnerabilities: The case of garment industry of Bangladesh [Decision Making in Enterprise Risk Management]. *Omega*, *57*, 5–21. https://doi.org/https://doi.org/10.1016/j.omega.2015.05.016

David, H., & Nagaraja, H. (1970). Order statistics wiley. *New York*.

Dotaniya, M., Meena, V., Rajendiran, S., Coumar, M. V., Saha, J., Kundu, S., & Patra, A. (2017). Geo-accumulation indices of heavy metals in soil and groundwater of Kanpur, India under long term irrigation of tannery effluent. *Bulletin of environmental contamination and toxicology*, *98*(5), 706–711. https://doi.org/10.1007/s00128-016-1983-4

Durbach, I. N., Algorta, S., Kantu, D. K., Katsikopoulos, K. V., & Şimşek, Ö. (2020). Fast and frugal heuristics for portfolio decisions with positive project interactions. *Decision Support Systems*, *138*, 113399. https://doi.org/10.1016/j.dss.2020.113399

Ehrgott, M., Figueira, J., & Greco, S. (2010). *Trends in multiple criteria decision analysis* (Vol. 6). Springer.

Eisenführ, F., Weber, M., & Langer, T. (2010). *Rational decision making*. Springer. https://doi.org/10.1007/978-3-642-02851-9

Figueira, J., Greco, S., & Ehrgott, M. (2005). *Multiple criteria decision analysis: State of the art surveys*. Springer Science & Business Media.

Fisher Science. (2014). *Calcium Hydroxide* [Material Safety Data Sheet]. https://beta-static.fishersci.com/content/dam/fishersci/en_US/documents/programs/education/regulatory-documents/sds/chemicals/chemicals-c/S25225.pdf

Fishman, G. S. (1996). Random Tours. *Monte Carlo*, 335–491.

Fox, G. E., Baker, N. R., & Bryant, J. L. (1984). Economic Models for R and D Project Selection in the Presence of Project Interactions. *Management Science*, *30*(7), 890–902. https://doi.org/10.1287/mnsc.30.7.890

Gabriela Cuadrado Quesada, T. H., Lena Breitenmoser. (2020). *Mainstreaming governance on wastewater treatment, water re-use and resource recovery: Learnings from India and the European Union*. The Energy, Resources Institute (TERI) Darbari Seth Block, India Habitat Centre, Lodhi Road, New Delhi – 110 003, India. https://pavitra-ganga.eu/en/events-output/deliverables

Gatian, K. N., & Mavris, D. N. (2015). Enabling technology portfolio selection through quantitative uncertainty analysis, 2436. https://doi.org/10.2514/6.2015-2436

Gigerenzer, G., & Goldstein, D. G. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological review*, *103*(4), 650. https://doi.org/10.1037/0033-295X.103.4.650

Gigerenzer, G., & Todd, P. M. (1999). *Simple heuristics that make us smart*. Oxford University Press, USA.

Golabi, K., Kirkwood, C. W., & Sicherman, A. (1981). Selecting a portfolio of solar energy projects using multiattribute preference theory. *Management Science*, *27*(2), 174–189. https://doi.org/10.1287/mnsc.27.2.174

Greco, S., Figueira, J., & Ehrgott, M. (2016). *Multiple criteria decision analysis* (Vol. 37). Springer.

Haag, F., Reichert, P., Maurer, M., & Lienert, J. (2019). Integrating uncertainty of preferences and predictions in decision models: An application to regional wastewater planning. *Journal of Environmental Management*, *252*, 109652. https://doi.org/https://doi.org/10.1016/j.jenvman.2019.109652

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Herman, J., & Usher, W. (2017). SALib: An open-source Python library for Sensitivity Analysis. *The Journal of Open Source Software*, *2*(9). https://doi.org/10.21105/joss.00097

Hueso, A., Boni, A., & Fernández-Baldor, Á. (2018). Embracing the complexity of policy processes in sanitation: Insights from India. *Development Policy Review*, *36*(2), 203–219. https://doi.org/10.1111/dpr.12246

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Iooss, B., & Lemaître, P. (2015). A review on global sensitivity analysis methods. *Uncertainty management in simulation-optimization of complex systems* (pp. 101–122). Springer. https://doi.org/10.1007/978-1-4899-7547-8_5

Ishigami, T., & Homma, T. (1990). An importance quantification technique in uncertainty analysis for computer models. *[1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, 398–403.

Jajmau Tannery Effluent Treatment Association (JTETA). (2018). "Request for proposal (RFP) for appointment of EPC contractors for implementation of 20 MLD tannery common effluent treatment plant (CETP) upto tertiary treatment along with treated sewage dilution facility for Jajmau leather cluster, Uttar Pradesh". https://nmcg.nic.in/writereaddata/fileupload/28_RFP%20for%20Appointment%20of%20EPC%20Contractor%20for%20Implementation%20of%20Jajmau%2020%20MLD%20CETP.pdf

Jaszkiewicz, A. (2002). On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment. *IEEE Transactions on Evolutionary Computation*, *6*(4), 402–412. https://doi.org/10.1109/TEVC.2002.802873

Jones, D., Tamiz, M. et al. (2010). *Practical goal programming* (Vol. 141). Springer.

Kantu, D. (2021). *A decision support system for selecting research proposals*. University of Cape Town.

Katsikopoulos, K. V., Durbach, I. N., & Stewart, T. J. (2018). When should we use simple decision models? a synthesis of various research strands. *Omega*, *81*, 17–25. https://doi.org/10.1016/j.omega.2017.09.005

Keeney, R. L. (1982). Feature Article—Decision Analysis: An Overview. *Operations Research*, *30*(5), 803–838. https://doi.org/10.1287/opre.30.5.803

Keeney, R. L., & Gregory, R. S. (2005). Selecting Attributes to Measure the Achievement of Objectives. *Operations Research*, *53*(1), 1–11. https://doi.org/10.1287/opre.1040.0158

Keisler, J. (2004). Value of information in portfolio decision analysis. *Decision analysis*, *1*(3), 177–189. https://doi.org/10.1287/deca.1040.0023

Keisler, J. (2011). Portfolio decision quality. *Portfolio decision analysis* (pp. 29–51). Springer. https://scholarworks.umb.edu/msis_faculty_pubs/43

Khatoon, N., Khan, A. H., Rehman, M., & Pathak, V. (2013). Correlation Study For the Assessment of Water Quality and Its Parameters of Ganga River, Kanpur, Uttar Pradesh, India. *IOSR Journal of Applied Chemistry*, *5*(3), 80–90. https://doi.org/10.9790/5736-0538090

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. Jupyter notebooks – a publishing format for reproducible computational workflows (F. Loizides & B. Schmidt, Eds.). In: In *Positioning and power in academic publishing: Players, agents and agendas* (F. Loizides & B. Schmidt, Eds.). Ed. by Loizides, F., & Schmidt, B. IOS Press. 2016, 87–90.

Köksalan, M., Wallenius, J., & Zionts, S. (2013). An early history of multiple criteria decision making. *Journal of Multi-Criteria Decision Analysis*, *20*(1-2), 87–94. https://doi.org/10.1002/mcda.1481

Kurttila, M., Haara, A., Juutinen, A., Karhu, J., Ojanen, P., Pykäläinen, J., Saarimaa, M., Tarvainen, O., Sarkkola, S., & Tolvanen, A. (2020). Applying a multi-criteria project portfolio tool in selecting energy peat production areas. *Sustainability*, *12*(5), 1705. https://doi.org/10.3390/su12051705

Lahdelma, R., Miettinen, K., & Salminen, P. (2003). Ordinal criteria in stochastic multicriteria acceptability analysis (SMAA). *European Journal of Operational Research*, *147*(1), 117–127. https://doi.org/10.1016/S0377-2217(02)00267-9

Lahtinen, T. J., Hämäläinen, R. P., & Liesiö, J. (2017). Portfolio decision analysis methods in environmental decision making. *Environmental Modelling & Software*, *94*, 73–86. https://doi.org/10.1016/j.envsoft.2017.04.001

Liesiö, J., Mild, P., & Salo, A. (2007). Preference programming for robust portfolio modeling and project selection. *European journal of operational research*, *181*(3), 1488–1505. https://doi.org/10.1016/j.ejor.2005.12.041

Liesiö, J., Mild, P., & Salo, A. (2008). Robust portfolio modeling with incomplete cost information and project interdependencies. *European Journal of Operational Research*, *190*(3), 679–695. https://doi.org/https://doi.org/10.1016/j.ejor.2007.06.049

Liesiö, J., Salo, A., Keisler, J. M., & Morton, A. (2021). Portfolio decision analysis: Recent developments and future prospects. *European Journal of Operational Research*, *293*(3), 811–825. https://doi.org/https://doi.org/10.1016/j.ejor.2020.12.015

Lofrano, G., Meriç, S., Zengin, G. E., & Orhon, D. (2013). Chemical and biological treatment technologies for leather tannery chemicals and wastewaters: A review. *Science of The Total Environment*, *461-462*, 265–281. https://doi.org/10.1016/j.scitotenv.2013.05.004

Lorie, J. H., & Savage, L. J. (1955). Three Problems in Rationing Capital. *The Journal of Business*, *28*(4), 229–239. http://www.jstor.org/stable/2351049

Lusk, E. J., & Kersnick, M. (1979). The Effect of Cognitive Style and Report Format on Task Performance: The MIS Design Consequences. *Management Science*, *25*(8), 787–798. https://doi.org/10.1287/mnsc.25.8.787

Martello, S., Pisinger, D., & Toth, P. (2000). New trends in exact algorithms for the 0–1 knapsack problem. *European Journal of Operational Research*, *123*(2), 325–332. https://doi.org/10.1016/S0377-2217(99)00260-X

Marttunen, M., Lienert, J., & Belton, V. (2017). Structuring problems for Multi-Criteria Decision Analysis in practice: A literature review of method combinations. *European Journal of Operational Research*, *263*(1), 1–17. https://doi.org/10.1016/j.ejor.2017.04.041

Matanò, A. (2019). *Decision analysis support for sustainable allocation of newly-arrived refugees according to water security criteria*. Delft University of Technology. http://resolver.tudelft.nl/uuid:90774870-b94f-4b0e-a396-a543e1471576

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

Mild, P., Liesiö, J., & Salo, A. (2015). Selecting infrastructure maintenance projects with robust portfolio modeling. *Decision Support Systems*, *77*, 21–30. https://doi.org/10.1016/j.dss.2015.05.001

Nanami Gange & National Institute of Urban Affairs. (2020). *A strategic framework for managing urban river stretches in the Ganga river basin: Urban river management plan (urmp)*. https://bit.ly/3u489ak

National Mission for Clean Ganga (NMCG). (2016). "Aim & Objective of NMCG". https://nmcg.nic.in/aims_obj.aspx

Palczewski, K., & Sałabun, W. (2019). Influence of various normalization methods in PROMETHEE II: an empirical study on the selection of the airport location [Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES2019]. *Procedia Computer Science*, *159*, 2051–2060. https://doi.org/10.1016/j.procs.2019.09.378

Papadopoulos, C. E., & Yeung, H. (2001). Uncertainty estimation and Monte Carlo simulation method. *Flow Measurement and Instrumentation*, *12*(4), 291–298. https://doi.org/10.1016/S0955-5986(01)00015-2

Parween, M., Ramanathan, A., & Raju, N. (2017). Waste water management and water quality of river Yamuna in the megacity of Delhi. *International Journal of Environmental Science and Technology*, *14*(10), 2109–2124. https://doi.org/10.1007/s13762-017-1280-8

Phillips, L. D., & Bana e Costa, C. A. (2007). Transparent prioritisation, budgeting and resource allocation with multi-criteria decision analysis and decision conferencing. *Annals of Operations Research*, *154*(1), 51–68. https://doi.org/10.1007/s10479-007-0183-3

Pisinger, D. (1997). A minimal algorithm for the 0-1 knapsack problem. *Operations Research*, *45*(5), 758–767. https://doi.org/10.1287/opre.45.5.758

Reddy, B., & Dubey, S. K. (2019). River Ganges water as reservoir of microbes with antibiotic and metal ion resistance genes: High throughput metagenomic approach. *Environmental Pollution*, *246*, 443–451. https://doi.org/10.1016/j.envpol.2018.12.022

Reich, G. (2005). Leather. *Ullmann's encyclopedia of industrial chemistry*. John Wiley & Sons, Ltd. https://doi.org/10.1002/14356007.a15_259.pub2

Rietveld, P., & Ouwersloot, H. (1992). Ordinal data in multicriteria decision making, a stochastic dominance approach to siting nuclear power plants. *European journal of operational research*, *56*(2), 249–262.

Rosenhead, J., & Mingers, J. (2001). *Rational analysis for a problematic world revisited*. John Wiley & Sons.

Roy, B. (1971). Problems and methods with multiple objective functions. *Mathematical programming*, *1*(1), 239–266.

Roy, B., & Słowiński, R. (2013). Questions guiding the choice of a multicriteria decision aiding method. *EURO Journal on Decision Processes*, *1*(1), 69–97. https://doi.org/10.1007/s40070-013-0004-7

Saharan, T., Scholten, L., Anshuman, & Phukan, M. (In preparation). Multi-criteria decision analysis and portfolio models to support regional wastewater management. Deliverable D2.3. EU H2020 Research and Innovation Action project "Pavitra Ganga – Unlocking wastewater treatment, water re-use and resource recovery opportunities for urban and peri-urban areas in India".

Salo, A., Keisler, J., & Morton, A. (2011). *Portfolio decision analysis: Improved methods for resource allocation* (Vol. 162). Springer Science & Business Media.

Saltelli, A., Aleksankina, K., Becker, W., Fennell, P., Ferretti, F., Holst, N., Li, S., & Wu, Q. (2019). Why so many published sensitivity analyses are false: A systematic review of sensitivity analysis practices. *Environmental modelling & software*, *114*, 29–39.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. (2008). *Global sensitivity analysis: The primer*. John Wiley & Sons.

Santy, S., Mujumdar, P., & Bala, G. (2020). Potential impacts of climate and land use change on the water quality of Ganga River around the industrialized Kanpur region. *Scientific reports*, *10*(1), 1–13. https://doi.org/10.1038/s41598-020-66171-x

Scholten, L., Schuwirth, N., Reichert, P., & Lienert, J. (2015). Tackling uncertainty in multi-criteria decision analysis – An application to water supply infrastructure planning. *European Journal of Operational Research*, *242*(1), 243–260. https://doi.org/10.1016/j.ejor.2014.09.044

Simon, H. A. (1955). A behavioral model of rational choice. *The quarterly journal of economics*, *69*(1), 99–118. https://doi.org/10.2307/1884852

Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and computers in simulation*, *55*(1-3), 271–280.

Stummer, C., & Heidenberger, K. (2003). Interactive R&D portfolio analysis with project interdependencies and time profiles of multiple objectives. *IEEE Transactions on Engineering Management*, *50*(2), 175–183. https://doi.org/10.1109/TEM.2003.810819

Tervonen, T., & Lahdelma, R. (2007). Implementing stochastic multicriteria acceptability analysis. *European Journal of Operational Research*, *178*(2), 500–513. https://doi.org/10.1016/j.ejor.2005.12.037

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *science*, *185*(4157), 1124–1131.

United Nations Industrial Development Organization (UNIDO). (2011). "introduction to treatment of tannery effluents". https://www.unido.org/resources/publications/creating-shared-prosperity/agribusiness-and-rural-entrepreneurship/introduction-treatment-tannery-effluents

Vafaei, N., Ribeiro, R., & Camarinha-Matos, L. (2018). Selection of Normalization Technique for Weighted Average Multi-criteria Decision Making. https://doi.org/10.1007/978-3-319-78574-5_4

Van Rossum, G. (2020). *The python library reference, release 3.8.2*. Python Software Foundation.

Vilkkumaa, E., Salo, A., & Liesiö, J. (2014). Multicriteria portfolio modeling for the development of shared action agendas. *Group decision and negotiation*, *23*(1), 49–70. https://doi.org/10.1007/s10726-012-9328-0

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Weingartner, H. M. (1966). Capital Budgeting of Interrelated Projects: Survey and Synthesis. *Management Science*, *12*(7), 485–516. https://doi.org/10.1287/mnsc.12.7.485

A
Linear-additive portfolio value model

A.1. Knapsack problem

In this section of the appendix the creation of portfolios through a linear-additive portfolio value function is presented. This form of problem is also known as the "Knapsack" model (Liesiö et al., 2008).

From Decision Matrix:

Table A.1: Matrix of alternative and criteria performances

	x	w
A_1	x_1	w_1
A_2	x_2	w_2
A_3	x_3	w_3
A_m	x_m	w_n

Let us for the sake of explaining keep the attributes explicit and simple for now. Let there be a room of stuff you want to put in your knapsack. However, there is just so much weight you can take with you. This weight is a constraint, C. Everything in the room has both a value, x, and a weight, w to it. You want to have max x and $w \leq C$. The optimization can be done by comparing the total outputs over all combinations possible. To make these combination the 'binary decision variable, z, is introduced. It means that when a alternative is chosen, A2, for example that this z is a 1, is it is not chosen it is a 0.

$$V(P) = \sum_{i=1}^{m} x_i z_i \tag{A.1}$$

and

$$W(P) = \sum_{i=1}^{m} w_i z_i \leq C, z_i \in \{0, 1\}, \forall\, i \tag{A.2}$$

The symbol $\forall$ means “for all” or “for any”. If you test a combination of project 1 and 3 (P={1,3}) this gives z = [1,0,1,0] If you test a combination of project 2, 3 and n, (P={2,3,m}), this gives z=[0,1,1,1]

The value for the first proposed set will than be:

$$V(P\{1,3\}) = x_1 z_1 + x_2 z_2 + x_3 z_3 + x_n z_n \tag{A.3}$$

$$V(P\{1,3\}) = x_1 * 1 + x_2 * 0 + x_3 * 1 + x_n * 0 = x_1 + x_3 \tag{A.4}$$

Consequently

$$W(P\{1,3\}) = w_1 + w_3 \tag{A.5}$$

Now this is just optimization for 1 criterion under on constraint. Sometimes you want a certain criteria to be high, maximize it, and other times you want to minimize, such as cost. These criteria have than to be balanced to each other to arrive at an optimum. Here the preferences of the DM also comes in place, since one criterion can be much more important than another.

For matrix there are 2 criterion, c_1 and c_2, these have also a meaning to them for the DM. c_1 is twice as important as c_2. This gives the *normalised* 'weight' criterion; $w_1 = (2/3)$ and $w_2 = (1/3)$

Table A.2: Matrix of alternative and criteria performances

	c_1	c_2
A_1	x_{11}	x_{12}
A_2	x_{21}	x_{22}
A_3	x_{31}	x_{32}
A_m	x_{m1}	x_{m2}

You have to maximize the:

$$Max\ V(P) = (\sum_{i=1}^{m}(x_{i1}w_1) + \sum_{i=1}^{m}(x_{j2}w_2))z_i \tag{A.6}$$

Or written neatly for c_n criterion, where you want to maximize all criterion:

$$Max\ V(P) = \sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}w_j)z_i \tag{A.7}$$

For Portfolio option P{1,2,m} the z = [1,1,0,1] and suppose you want to calculate the value of this portfolio set:

$$V(P\{1,2,m\}) = (w_1x_{11} + w_2x_{12}) * 1 + (w_1x_{21} + w_2x_{22}) * 1 + (w_1x_{n1} + w_2x_{m2}) * 1 \tag{A.8}$$

Then there can be still the hard constraint C. Many research has been done to develop algorithms that efficiently search trough all combinations. As you might imagine, the amount of combinations increase very fast with increasing alternatives and criterion.

A.2. With interactions

Now let us set one step further and also elaborate on a way to incorporate the interactions or constraints (Stummer & Heidenberger, 2003). First there are the interactions benefits: determine which alternatives (m), have an interaction impact with another alternative for a certain criteria, n. So you have subsets of alternatives for each criterion; These are called *Interaction subsets*: A_j^+.

For example, criterion 2, i = 2, has projects 1, 2 and m that create a synergy for this criteria. For c_2 this would thus be the set:

$$A_2^+\{1, 2, m\} \tag{A.9}$$

Furthermore, it is decided that at least 2, K = 2, of these projects have to be selected in the portfolio in order to get the benefit, as a minimum threshold. When that is established, the difficult part comes from a data perspective: What is the added benefit of the combination? Let for now be that $y(k)$, it changes with each added project, when $k > K$. If this threshold is reached we will use g, similar to z, to 'activate' the benefit.

$$Max\ V(P) = \sum_{i=1}^{m}(\sum_{j=1}^{n} x_{ij}w_j)z_i + \sum_{i=1}^{m}(\sum_{j=1}^{n} y_{ij}(k)w_j)g_i \tag{A.10}$$

The added value due to interactions are initially added to the value column for the corresponding criterion. If interactions can exist over criterion, one has to choose where to add that, that is up to you and formulate a extra interaction subset with corresponding y(k). There is virtually no limit to how many subsets you create with corresponding y(k). Sometimes a clear y(k) can be deducted for many alternatives at once, but it is more probable that is often is for a limited subset. The analysing of all combinations of possible interactions than still can be quite a long process.

For example, for c_2, there are now 4 alternatives to be considered. Alternative 1-3 and m. We established that for this criterion there is the added benefit from two different sets: A_{21}^{+} and A_{22}^{+}.

For the first set there must be all 3 in order to get the benefit. The sum of projects in set A22 must be larger or equal to this threshold.

$$A_{21}^{+}\{1,2,3\},\ for\ k\ \in A_{21}^{+} \geq K_{21} = 3 \tag{A.11}$$

This yields in extra benefit:

$$y_{21}(k) = 5k + 8 \tag{A.12}$$

For the other one there only have to be 2 in there:

$$A_{22}^{+}\{1,2,m\},\ for\ k\ \in A_{22}^{+} \geq K_{22} = 2 \tag{A.13}$$

which yields in extra benefit:

$$y_{22}(k) = 2k^2 + 5 \tag{A.14}$$

Here you can see that there might be different interactions and benefits. These y(k), can also be constants. Another way to form an overview of the added benefit is using a γ interaction multiplier for all combinations in a table (Kantu, 2021).

B
Linear Aggregated Value Function Properties

The function in the MAVT approach to aggregate the separate attribute values generated by the marginal value function is called the "hierarchical aggregation function". In the thesis the linear version is assumed to be a good approximation. Implicitly it also assumed that the attributes are compensatory. This is in contrast to a multiplicative hierarchical aggregation function where attributes are 'non-compensatory'. The attributes and data have to adhere to some properties to be able to assume linear-additivity (Belton and Stewart, 2002; Eisenführ et al., 2010 (Chapter 6)):

- Preference completeness: A preference is complete if the decision maker has a preference for any pair of alternatives; There are no combinations to which the decision maker does not know. It is allowed to value any pair equally, but not to not value the alternatives/portfolios.
- Transitivity: A preference is transitive if:

$$a \succ b \; and \; b \succ c \; then \; a \succ c \tag{B.1}$$

- Mutual preferentially independence: A preference of a value over another value in a particular portfolio: colour white (value) for a BMW (Portfolio) must always be preferred over green (value). If the portfolio changes to a VW; the colour white must still be preferred over green. This an example for a single preferentially independence. Mutual independence simply means for multiple attributes simultaneously (both colour and wheel size for example).

$$(White,\ BMW) \succeq (Green,\ BMW) \sim (White,\ VW) \succeq (Green,\ VW) \tag{B.2}$$

- Difference independence: The impacts of an attribute may not have an impact on the value function of another attribute. In Eisenführs book (page 132-133) the example is made that holiday days off are more worth when the salary is higher. Because you can make fancier trips. In this situation this difference independence property is not met. If, however, the salary impacts for the alternatives are within a more close range one may argue that they all impact the holiday attribute equally. Thus one can choose to say that this property is upheld. This depends on how strict the analyst wants to be.

$$V_{holiday}(days) \mid_{High\ Salary} \approx V_{holiday}(days) \mid_{Low\ Salary} \tag{B.3}$$

C
Stochastic multicriteria acceptability analysis (SMAA)

The aim to deal with limiting information and time cases lead to the need to deal with (non complete) ordinal rankings of attributes. An ordinal ranking must be treated thoroughly by analysing the possible weight vectors. Strictly speaking the weights applied to the aggregated value functions cannot really be individually independently altered due to the definition of C.2 in normalised condition. Even in their non-normalised form the weights are not independent; an increase in one attribute weight results a relative decrease in the other weighs. One can choose to ignore this effect and still vary around a weight to measure the impact, however one should note that the impact of a certain change is more than the change on just that weight, but strengthened by this decrease in the other weights. Due to these considerations and often lacking certain weight information a set of methods for analysis of the so-called *weight space*, W, have been developed for *complete ordinal rankings*. These can than be compared with the hypothetical known attribute weights in cardinal values that correspond with the ordinal ranking to compere the outcomes. So, first the matter of complete ordinal rankings.

By using stochastic methods w is sampled many times from the weight space. These weight vectors have to adhere to the weight simplex (eq. C.2) and to known information, such as cardinal attributes weight data with confidence intervals or ordinal rankings of attributes. Several approaches have been undertaken by Charnetski, Soland, Rietveld and Ouwersloot, for example. (Charnetski and Soland, 1978; Rietveld and Ouwersloot, 1992; Ehrgott et al., 2010 (Chapter 10)) for more analytical approaches. However these methods have serious constraints that are not a problem for numerical methods such as SMAA-O. The family of these methods are known as *stochastic multicriteria acceptability analysis (SMAA)* (Tervonen & Lahdelma, 2007). Since we are dealing with ordinal attribute ranking, the approach taken here is SMAA-O (Lahdelma et al., 2003).

The set of all possible weight options is called the *weight space* (Ehrgott et al., 2010 (Chapter 10)). If weights are non-negative and normalized the weight space is an (n-1)-dimensional simplex in n-dimensional space:

$$W = \left\{ w \in R^n : w \geq 0 \;\; and \sum_{j=1}^{n} w_j = 1 \right\} \tag{C.1}$$

Or to put it more readable:

$$W = \{w | w_j \geq 0 \; and \; w_1 + w_2 \; + \; ... \; w_n = 1\} \tag{C.2}$$

From these sets of *weight vectors* can be extracted;

$$\boldsymbol{w} = [w_1, w_2 ..., w_j] \tag{C.3}$$

An ordinal ranking of attributes gives restrictions to the weight space W, and the possible $\boldsymbol{w}$ that thus can be generated from this space. The more information is available and uncertainty decreases the weight space becomes smaller.

The third possibility discussed is an incomplete or partial ranking of the attributes. This would for example be when only the top two most important attributes are identified. No other information on the other attributes is known; this is different than indifference between attributes.

The first step of a SMAA is the generation of the possible weight vectors that adhere to the ordinal ranking restriction and cover a sufficient area of the weight space:

$$0 \leqslant w_j^{min} \leqslant w_j \leqslant w_j^{max} \leq 1, \quad where\ j \in \{1, ..., n\} \tag{C.4}$$

Because the weights must sum to unity, the n weights w_j are generated according to the following method: first $(j-1)$ independent random numbers are generated from the uniform distribution in interval [0, 1]. These are sorted into ascending order and after that, 1 is inserted as the last number and 0 as the first number of an intermediate array. The intervals between these values are than calculated. Subsequently they are sorted resulting in a weight vector corresponding with the ordinal distribution of j attributes (David and Nagaraja, 1970; Tervonen and Lahdelma, 2007). The corresponding samples cover a constrained subset of the weight space, W', the constraints are formulated by the ordinal ranking. This is visualised for 3 attributes in Figure C.1.

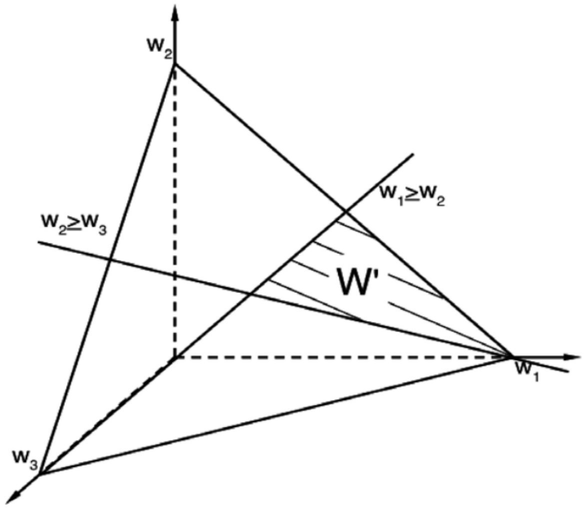


Figure C.1: Feasible weight space of a three-attribute problem with ranking of attributes (Tervonen & Lahdelma, 2007)

Consequently the result in n amount of rankings from which the robust portfolios can extracted. Robustness is defined by acceptability indices. These sampled weight vectors are than used to generate the value distributions, $\rho(V(\xi, w)$ for the portfolios, see Table C.1.

Table C.1: 9 samples from a 11-D simplex, corresponding to 11 attributes; Each row sums to one.

	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Rank 7	Rank 8	Rank 9	Rank 10	Rank 11
0	0.481544	0.165544	0.110538	0.089643	0.041852	0.032425	0.030218	0.027044	0.012029	0.007977	0.001187
1	0.454774	0.148523	0.107389	0.088901	0.044419	0.037636	0.037171	0.034809	0.027544	0.014234	0.004601
2	0.428004	0.131501	0.104241	0.088160	0.047297	0.046987	0.045054	0.043059	0.037192	0.020490	0.008015
3	0.401234	0.114480	0.101092	0.087419	0.058573	0.057424	0.052472	0.049554	0.039576	0.026746	0.011429
4	0.374464	0.097944	0.097459	0.086678	0.074088	0.067551	0.059890	0.052121	0.041960	0.033002	0.014843
5	0.347695	0.094796	0.089603	0.085936	0.080438	0.077677	0.067308	0.054689	0.044343	0.039258	0.018258
6	0.320925	0.105117	0.091647	0.087804	0.085195	0.074726	0.063417	0.057256	0.046727	0.045514	0.021672
7	0.294155	0.120632	0.097931	0.088499	0.084454	0.082144	0.059823	0.051770	0.049110	0.046396	0.025086
8	0.267385	0.136147	0.108057	0.089562	0.085350	0.083713	0.062391	0.058026	0.051494	0.029375	0.028500
9	0.240616	0.151661	0.118184	0.096980	0.082971	0.082202	0.064958	0.064282	0.053878	0.031915	0.012353

From these weights distributions and the partial value distribution functions the rank of each portfolio is determined via the ranking function:

$$rank(i, \xi, w) = 1 + \sum_{k=1}^{m} \rho(V(\xi_k, w) > V(\xi_i, w)) \tag{C.5}$$

Here $\rho(true)$ = 1 and $\rho(false)$ = 0 when comparing V_k and V_i. Better performing means higher rank.

For differing weight vectors in the Weight space, the portfolio remains ranked r. This leads to a subset of the weight space, $W_i^r(\xi)$ where the portfolio i is ranked r.

$$W_i^r(\xi) = \{w \in W' : \ rank(i, \xi, w) = r\} \tag{C.6}$$

The portfolios with large weight spaces subsets on high ranks (high expected value $E[X]$) are the most preferred and most robust portfolios. This is calculated in so-called *acceptability indices*, b: The proportion(or percentage) of W' which results in r for portfolio i. Technically this is the integral over the subset W' for $W_i^r(\xi)$ and also the $E[X]$:

$$b_i^r = \int_X f(\xi) \int_{W_i^r(\xi)} f(w) \ dw \ d\xi \tag{C.7}$$

And this can be read as the portfolio for which the values $E[X]$ lead to the rank, r (the first integral) for the share of the weight space (second integral).

D
Sobol Method

The Sobol sensitivity analysis is the is one of the best methods of doing sensitivity analysis (Iooss & Lemaître, 2015). In contrast to the One-At-The-Time approaches it samples all possible inputs defined by a range on which the inputs may vary. For example for the first variable, x_1, it measures the impact for a fixed x_1 whilst all the other variables are changed. Then the mean value or expected value of the output distribution is determined (Ishigami & Homma, 1990):

$$E_{X\sim x1}(Y|x1) \tag{D.1}$$

In order to do this first the input variables have to be determined for a certain equation. In this case the weights. Then the range that the variables can occupy, in this case [0,1]. Since there are 11 weight as an input for the global sensitivity analysis a large amount of samples have to be generated. The rule of thumb using Sobol for D variables; 2N(D + 1) samples and N>1000 before indices stabilize. This has to be determined on the deviation on the indices. For only first-order Sobol indices the amount of samples needed is considerable less: N(D+1) samples (Azzini et al., 2021). However as in the cited paper, there is done some research to these 'estimators'. A more efficient sampling can be achieved by using Latin Hypercube sampling (LHS) and such, however there are also some issues with the randomness of the method. Therefore the original Monte Carlo sampling is used.

This is then repeated for the entire sample range of x_1. This generates the variance V_{x1} over the sample set. This variance divided by the variance of the entire output gives the Sobol index to indicate the strength of the analysed variable on the system output:

$$S1_{x1} = \frac{V_{x1}[E_{X\sim x1}(Y|x1)]}{V(Y)} \tag{D.2}$$

If the system is completely linear the sum of the first-order Sobol indices of all variables equals 1. Hence the total sobol index, ST, also equals 1:

$$\sum_{j=1}^{m} S_{1j} = \sum_{j=1}^{m} S_{Tj} = 1 \tag{D.3}$$

However, when there are interactions between the variables, the so-called second order, S2, impacts this is not the case. Technically even higher order interactions could be researched.

The Second order or S2 interactions are determined between for each variable set; (x1 - x2) and (x1-x3). The variance found of changing this set whilst covering the entire sample set of the other variables is than subtracted by the individual variances. The found variance is then caused by the interaction not found by the individual. Otherwise the outcome would be zero.

$$S2_{x1,x2} = \frac{E_{X\sim x1,x2}[V_{x1,x2}(Y|X \sim x1,x2)]}{V(Y)} - S1_{x1} - S1_{x2} \tag{D.4}$$

Similarly one could repeat this process even for higher order interactions. The current state of computer hardware and open-source cooperation makes this model ever more fast where it was too slow and computational expensive in the past. This method is done using the SALib library in python (Herman & Usher, 2017), see code in Appendix K.

E
Normalisation Procedures

To compare criteria in an alternative performance matrix the data usually has to be normalised. This is needed to compare the different metrics and different ranges of the values. The impact of normalization procedure can affect the outcomes of the rankings considerably (Palczewski & Sałabun, 2019) and is not neutral! In this appendix often used normalisation techniques are considered that are often applied (Vafaei et al., 2018). For the thesis multiple of these methods are implemented in the procedure and impacts are assessed. There is not one overall best or neutral normalisation technique and the choice depends on the circumstances such as MCDA method applied; for TOPSIS vector normalization is shown to be most robust (Chakraborty & Yeh, 2009) and for SAW it is Linear Sum (Vafaei et al., 2018). The reason that the normalisation procedure can impact is due the different sort of ranges the normalisation outcomes take up. For a further discussion on this mathematical problem I refer to the papers cited. Let us consider the Matrix E.1, where cleaning capacity in millions of liters a day and profits in dollars are considered.

Table E.1: Matrix of alternative and criteria performances

	$Capacity(MLD)$	$Profit(\$)$
A_1	15	3
A_2	37	2
A_3	7	9
A_m	23	2.7

E.1. Linear: Max

For m alternatives/rows the maximum of minimum value is taken, depending on whether a crition is to be maximised or minimised. The criteria values of this column are than divided by this value. This procedure creates normalised values in domain $(0 < r_{min}, 1]$.

For benefit-criteria:

$$r_{ij} = \frac{x_{ij}}{x_{max}} \tag{E.1}$$

For cost-criteria:

$$r_{ij} = 1 - \frac{x_{ij}}{x_{max}} \tag{E.2}$$

E.2. Linear Max-Min

Creates a value domain of [0,1]. For benefit-criteria:

$$r_{ij} = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}} \tag{E.3}$$

For cost-criteria:

$$r_{ij} = \frac{x_{max} - x_{ij}}{x_{max} - x_{min}} \tag{E.4}$$

E.3. Linear Sum

This procedure creates normalised values in domain $(0 < r_{min}, r_{max} < 1)$. For benefit-criteria:

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \tag{E.5}$$

For cost-criteria:

$$r_{ij} = \frac{1/x_{ij}}{\sum_{i=1}^{m} 1/x_{ij}} \tag{E.6}$$

E.4. Vector Normalization

This procedure creates normalised values in domain $(0 < r_{min}, r_{max} < 1)$. For benefit-criteria:

$$r_{ij} = \frac{x_{ij}}{\sqrt{(\sum_{i=1}^{m} x_{ij}^2)}} \tag{E.7}$$

For cost-criteria:

$$r_{ij} = 1 - \frac{x_{ij}}{\sqrt{(\sum_{i=1}^{m} x_{ij}^2)}} \tag{E.8}$$

Example of vector normalisation of Table E.1: First calculate the magnitude of the vector, which is the formula in the denominator, for each criterion:

$$\sqrt{\sum_{i=1}^{m} x_{i,Capacity}^2} = \sqrt{15^2 + 37^2 + 7^2 + 23^2} = \sqrt{2172} \tag{E.9}$$

$$\sqrt{\sum_{i=1}^{m} x_{i,Profit}^2} = \sqrt{3^2 + 2^2 + 8^2 + 2.7^2} = \sqrt{101.29} \tag{E.10}$$

The individual values, x_{ij}, are then divided by the corresponding magnitudes of the criteria columns and yields the vector normalized matrix in the case of only benefit criteria. (Otherwise this value has to be deducted first from one as in equation E.8.)

For $x_{1,1}$ this means $15/\sqrt{2172} = 0.322$ and for $x_{1,2}$ this means $3/\sqrt{101.29} = 0.298$. Transforming table E.1 to Table E.2:

Table E.2: Original (upper) and Vector Normalized (lower) Matrix of alternative and criteria performances

	$Capacity(MLD)$	$Profit(\$)$
A_1	15	3
A_2	37	2
A_3	7	9
A_m	23	2.7
	$Capacity(MLD)$	$Profit(\$)$
A_1	0.322	0.298
A_2	0.794	0.199
A_3	0.150	0.894
A_m	0.494	0.268

F
Simulated Data Sets

Table F.1: Simulated Dataset used of impacts of Alternatives on Attributes

Alternative	Name	Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	Attribute_10	Attribute_11
1	A1	0	0	0	0	0	0	0	0	0	0	0
2	A2	30	60	0	85	19	-5	12	700	0	0	2
3	A3	40	80	20	66	18	0	10	500	2	0	0
4	A4	300	150	30	75	22	-15	0	2000	80	4	10
5	A5	350	240	20	0	20	-15	2	1520	5	100	2
6	A6	40	48	22	10	20	-3	18	650	0	0	2
7	A7	0	0	0	0	12	10	0	200	0	0	0
8	A8	50	65	35	0	20	0	20	-150	1	120	0
9	A9	185	36	90	85	20	-15	0	860	40	80	-10

Table F.2: Simulated Dataset used of deviations of the Alternatives on Attributes expressed in standard deviations expressed normal distributions

Alternative	Name	Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	Attribute_10	Attribute_11
1	A1	0.0	0.0	0.05	7.64	0.0	0.0	0.0	0	0.0	0.0	0.0
2	A2	3.0	6.0	0.00	8.50	1.9	0.5	1.2	70	0.0	0.0	0.2
3	A3	4.0	8.0	2.00	6.60	1.8	0.0	1.0	50	0.2	0.0	0.0
4	A4	30.0	15.0	3.00	7.50	2.2	1.5	0.0	200	8.0	0.4	1.0
5	A5	35.0	24.0	2.00	0.00	0.0	1.5	0.2	152	0.5	10.0	0.2
6	A6	4.0	4.8	2.20	1.00	0.0	0.3	1.8	65	0.0	0.0	0.2
7	A7	0.0	0.0	0.00	0.00	0.0	1.0	0.0	20	0.0	0.0	0.0
8	A8	5.0	6.5	3.50	0.00	0.0	0.0	2.0	15	0.1	12.0	0.0
9	A9	18.5	3.6	9.00	8.50	2.0	1.5	0.0	86	4.0	8.0	1.0

Table F.3: Definition of whether a cost of benefit value function is used for the attributes

Attribute_1	Attribute_2	Attribute_3	Attribute_4	Attribute_5	Attribute_6	Attribute_7	Attribute_8	Attribute_9	Attribute_10	Attribute_11
benefit	benefit	benefit	benefit	benefit	benefit	cost	cost	benefit	benefit	cost

Table F.4: Definition of units and boundaries of the simulated dataset

	Unit	Scale	Boundaries for global scales
0	Days/Year	global	(0, 365)
1	Days/Year	global	(0, 365)
2	%	local	0
3	%	local	0
4	Rupees	local	0
5	%	local	0
6	mg/g	local	0
7	Rs/year	local	0
8	Days/Year	global	(0, 365)
9	Days/Year	global	(0, 365)
10	Ha	local	0

Table F.5: Combinations of Alternatives that produce a 'synergy' value in the simulated dataset

	Combination pair 1	Combination pair 2
0	2	3
1	2	4
2	5	6

Table F.6: Ordinal ranking of the attribute importance for Actor X

	Ordinal Ranking Actor X
Attribute_1	2
Attribute_2	3
Attribute_3	4
Attribute_4	5
Attribute_5	6
Attribute_6	7
Attribute_7	8
Attribute_8	1
Attribute_9	10
Attribute_10	11
Attribute_11	9

Table F.7: Ordinal ranking of the attribute importance for Actor Y

	Ordinal Ranking Actor Y
Attribute_1	1
Attribute_2	2
Attribute_3	3
Attribute_4	4
Attribute_5	5
Attribute_6	6
Attribute_7	7
Attribute_8	8
Attribute_9	9
Attribute_10	10
Attribute_11	11

G
Case Study Information

G.1. Background Information: Waste water treatment in India

Northern India has several import rivers, such as the Yamuna, Narmada, Sindhu of course the Ganga river. These rivers are not only of paramount importance to India as means of water supply and economic activity but are also of religious importance. Due to the large demographic and economic growth the current wastewater treatment systems have not been able to keep up with the increased effluent waste water. Next to demographic growth also urbanisation, the legacy of city planning and changing industrial production methods have contributed to increased pollution of rivers from cities and industrial clusters. From the onset of Indian independency a legacy of colonial city building and classes have divided the population in those who do have and those who do not have access to sanitation, clean drink water and proper disposal (Chaplin, 2011). After independency the post-colonial state and interest group driven politics in combination with rapid population growth resulted in even a starker inequality to access proper sanitation and sewage services. Thus, the upper-class neighbourhoods do have access to both drinking water supply and a sewage system. Today the largest discharge of untreated or improper treated wastewater from residential groups derive from unofficial settlements, e.g. 'slums' (Chaplin, 2011). These neighborhoods do not have or have insufficient access to clean drinking water, sanitation and sewage (Hueso et al., 2018). The effluent liquad and solid waste, both residential and industrial, is often dumped in the rivers directly or in open sewers and storm-drains systems that end up in the river or farmlands downstream. As a result the river pollution reached levels that cross dangerous thresholds and the health of the population using the river and effluent discharge levels is affected (Parween et al., 2017; Reddy and Dubey, 2019; Ahmed and Ismail, 2018).

For this reason several governments have pledged to clean the rivers. Most recently the Union Government under Modi has done so again; In 2014 *Namami Gange Programme* was launched under the guidance of the *National Mission for Clean Ganga (NMCG)* with the goal of (National Mission for Clean Ganga (NMCG), 2016):

- To ensure effective abatement of pollution and rejuvenation of the river Ganga by adopting a river basin approach to promote inter-sectoral co-ordination for comprehensive planning and management
- To maintain minimum ecological flows in the river Ganga with the aim of ensuring water quality and environmentally sustainable development

The Nanami Gange Programme and ministry of Urban Affairs have prepared a framework to guide the policy development in order to get to clean the Ganga river; *the Urban River Management Plan (URMP)* (Nanami Gange & National Institute of Urban Affairs, 2020). The Urban Local Bodies (ULB), similar to municipalities, are responsible to create plans, regulations and policies to solve the problems and to dovetail the Urban River Management Plan Framework (URMP) into the future municipal development plans, the so-called Master Plans. (Nanami Gange & National Institute of Urban Affairs, 2020). From the federal level there seems to be the political will to change the current dire situation. The current challenge is to translate this political will into actions along the banks of the long river. Here the Pavitra Ganga project enters the stage.

G.1.1. Working Directive 3: MCDA

G.1.2. Summary Case Study

To summarize the situation in the case study: In the Pavitra Ganga project there is a group of stakeholders with conflicting interests. They have different preferences for certain attributes and alternatives. Moreover, the different actors are able to intervene in different ways in the system. Furthermore, the alternatives are different in their nature: from soil remediation to technical installations. Then there is the aspect of interactions between the alternatives. It can be reasonably argued that the case study can be formulated as both a portfolio and multi-criteria problem. As already presented the current body of literature does not present a methodological framework that deals with such as situation. How to support such a decision-making situation? Firstly, the matter of different agencies of the actors; The complete challenge of cleaning the Ganga River and creating a water-reuse system and to initiate resource recovery spans the influence of multiple actors. How to create portfolios that combine the different alternatives the different actors can undertake? How to communicate the successful portfolios to inform and convince an actors. Here also the matter of interactions come about, the individual actor might not see the outcome of its own action on its own (if others are also not doing anything). The other aspect might be shared responsibility; How to deal with the potential outcome that the dominating portfolio sets would comprise a multitude of alternatives from one actor, who would see this as 'unfair'? When such questions are answered a shared support base for joint action will arise.

Next to the matter of confidence there is another matter that makes this situation difficult. The time and accessibility to the actors is very limited. Therefore, preference modelling has to be done with little information. Another matter is the lack of time and access to model possible impact synergies between alternatives; again information is scarce. The Pavitra Ganga Project has several work packages and this thesis hopes to contribute within the framework of the PG project and work package "Water Governance". However, overlap exists between the work packages.

A sub-deliverable of the PG Water Governance work package is deliverable D2.3 (Saharan et al., In preparation): "Multi criteria decision analysis and portfolio models to support regional water management including performance assessment of the individual technologies and broader strategy portfolios". The need for this sub-deliverable is also found by a previous finding in the project from from Gabriela Cuadrado Quesada in *Mainstreaming governance on wastewater treatment, water re-use and resource recovery: learnings from India and the European Union*, section 4.2 (pg. 17) (Gabriela Cuadrado Quesada, 2020): "The choice of technology to treat and recycle wastewater and sewage sludge has to be guided by the physical and socio-economic constraints as well as the intended uses of the treated wastewater (fit- for-purpose treatment)."

Deliverable D2.3; Multi-Criteria Decision Analysis gives a systematic overview of all the relevant possible ways to move forward. An important challenge in this analysis are the different stakeholders. To this extend workshops are/have been organized with selected stakeholders from the Kanpur case (Saharan et al., In preparation).

The way a suitable analysis is carried out is via a certain analysis. Such as Cost-Benefit Analysis, Multi-Criteria Decision Analysis etc. Due to the nature of the situation the project agreed upon doing a MCDA approach incorporating different relevant criteria. For both a technical and socio-technical approach a multi-criteria analysis can be rather helpful, especially when dealing with a policy arena where structured decision-making can help the process. There are multiple ways such a MCDA can be carried out depending on the pursued types outcomes, data available and type of problem. For the PG Project the initial project idea was to compare the innovative technologies developed for the two case studies. However, the outcomes of the workshops might broaden the analysis such that it might also incorporate socio-technical aspects, such as legislation, awareness campaigns, training programs etc etc.

G.2. Decision Matrix

Table G.1: Attribute Legenda (Saharan et al., In preparation)

Percentage_of_produced_WW_that_is_reused	A1
Percentage_WW_discharched_better_or_equal_to_discharche_standards.	A2
Percentage of Hazardeous Sludge Processed	A3
Fossil_energy_used_for_WW_treatment	A4
Time_irrigation_water_meets_reuse_standards	A5
Profits_from_hides_processing	A6
Crop_Productivity_of_farmland	A7
Chrome_concentration_in_Soil	A8
Annualized_Investment_cost_percentage_avg_household	A9
Annualized_O&M_cost_percentage_avg_household	A10
Percentage_of_O&M_cost_covered_by_sewer_tax	A11
Time_without_electricity_interruptions	A12
Time_irrigation_water_demand_is_met	A13

Table G.2: Decision Matrix, Alternative Performances (Saharan et al., In preparation)

Alternative	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
1	0	0	0	0	0	0	0	0	0	0	100	0	0
2	19	315	0	0.45	275	0	0	250.88	2	0.5	50	0	0
3	9	300	75	0	250	-0.5	7	395.88	0	0.05	100	0	120
4	0	0	0	0	0	0	10	405.88	0.05	0.05	50	0	0
5	0	60	0	0.55	50	0	8	285.88	0.05	0.05	50	8	50
6	39	0	0	0	40	0	0	0	0.05	0.05	50	0	0
7	0	300	0	0.45	250	0	2	150	0.02	0	0	9	0
8	9	200	56	0	150	0	0	355.88	0.02	0.05	50	0	100
9	9	60	0	0.55	50	0	10	285.88	0.005	0.05	50	8	50
10	9	200	80	0	250	0	6	5	0.74	-0.06	100	12	0
11	0	0	80	0.1	0	0	0	0	0	0.03	0	0	0
12	0	300	85	0	300	0	15	405.88	0.05	0.05	50	0	0

Table G.3: Decision Matrix, Alternative Uncertainty Range (Standard Distribution) (Saharan et al., In preparation)

Alternative	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
1	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1.9	31.5	0	0.045	27.5	0	0	25.088	0.2	0.05	5	0	0
3	0.9	30	7.5	0	25	0.05	0.7	39.588	0	0.005	10	0	12
4	0	0	0	0	0	0	1	40.588	0.005	0.005	5	0	0
5	0	6	0	0.055	5	0	0.8	28.588	0.005	0.005	5	0.8	5
6	3.9	0	0	0	4	0	0	0	0.005	0.005	5	0	0
7	0	30	0	0.045	25	0	0.2	15	0.002	0	0	0.9	0
8	0.9	20	5.6	0	15	0	0	35.588	0.002	0.005	5	0	10
9	0.9	6	0	0.055	5	0	1	28.588	0.0005	0.005	5	0.8	5
10	0.9	20	8	0	25	0	0.6	0.5	0.074	0.006	10	1.2	0
11	0	0	8	0.01	0	0	0	0	0	0.003	0	0	0
12	0	30	8.5	0	30	0	1.5	40.588	0.005	0.005	5	0	0

Table G.4: Information whether the attribute partial value functions are benefit of cost functions

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
benefit	benefit	benefit	cost	benefit	benefit	benefit	benefit	cost	cost	benefit	benefit	benefit

Table G.5: Scales and Unit Information of the Attributes of the Case Study (Saharan et al., In preparation)

	Unit	scale type	scale
Percentage_of_produced_WW_that_is_reused	%	local	'N.A.'
Percentage_WW_discharched_better_or_equal_to_discharche_standards.	%	local	'N.A.'
Percentage of Hazardeous Sludge Processed	%	local	'N.A.'
Fossil_energy_used_for_WW_treatment	kWh/m3	local	'N.A.'
Time_irrigation_water_meets_reuse_standards	Days/Year	global	(0,365)
Profits_from_hides_processing	Rupees/sq Foot	local	'N.A.'
Crop_Productivity_of_farmland	%	local	'N.A.'
Chrome_concentration_in_Soil	mg/g	local	'N.A.'
Annualized_Investment_cost_percentage_avg_household	%	local	'N.A.'
Annualized_O&M_cost_percentage_avg_household	%	local	'N.A.'
Percentage_of_O&M_cost_covered_by_sewer_tax	%	local	'N.A.'
Time_without_electricity_interruptions	Days/Year	global	(0,365)
Time_irrigation_water_demand_is_met	Days/Year	global	(0,365)

G.3. Synergy Information

Table G.6: Combinations pairs of interactions for the alternatives: X = cannot be combined, 0 = no synergy, 1 = positive synergy, -1 = negative synergy (Saharan et al., In preparation)

Nr.	1	2	3	4	5	6	7	8	9	10	11	12
1	X											
2	X	X										
3	X	-1	X									
4	X	0	0	X								
5	X	0	0	0	X							
6	X	1	1	0	0	X						
7	X	1	0	0	1	0	X					
8	X	1	0	0	0	1	1	X				
9	X	0	0	0	-1	0	0	X	X			
10	X	0	0	0	X	1	0	0	X	X		
11	X	0	0	0	0	0	0	0	0	1	X	
12	X	-1	X	0	1	0	0	X	0	0	1	X

H Waste Water Treatment Systems Kanpur

In this appendix performances of current and tested sewage technologies are shown.

H.1. PETP Process Flow Diagram

The PETP present in the Indian tanneries might differ slightly from each other. The standard working of a PETP for tannery effluent, including solid sludge management is shown in figure H.1

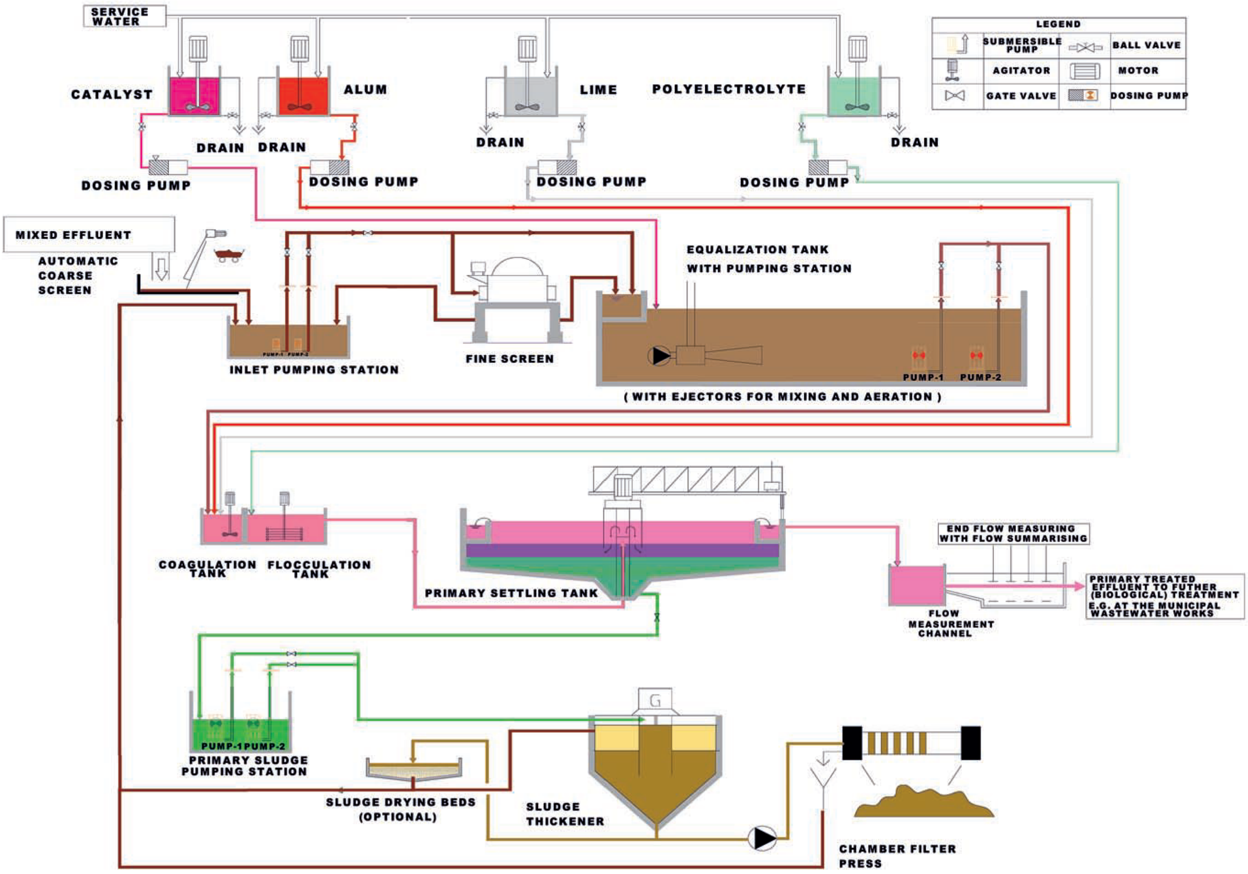


Figure H.1: Simplified flowchart of a full-fledged tannery effluent treatment plant (United Nations Industrial Development Organization (UNIDO), 2011)

An even more simplified flowchart of the PETPs in Jajmau is presented by an actor, shown below. These Flowcharts and thus the PETPs lack the solid sludge management.

H.2. CETP and STP information

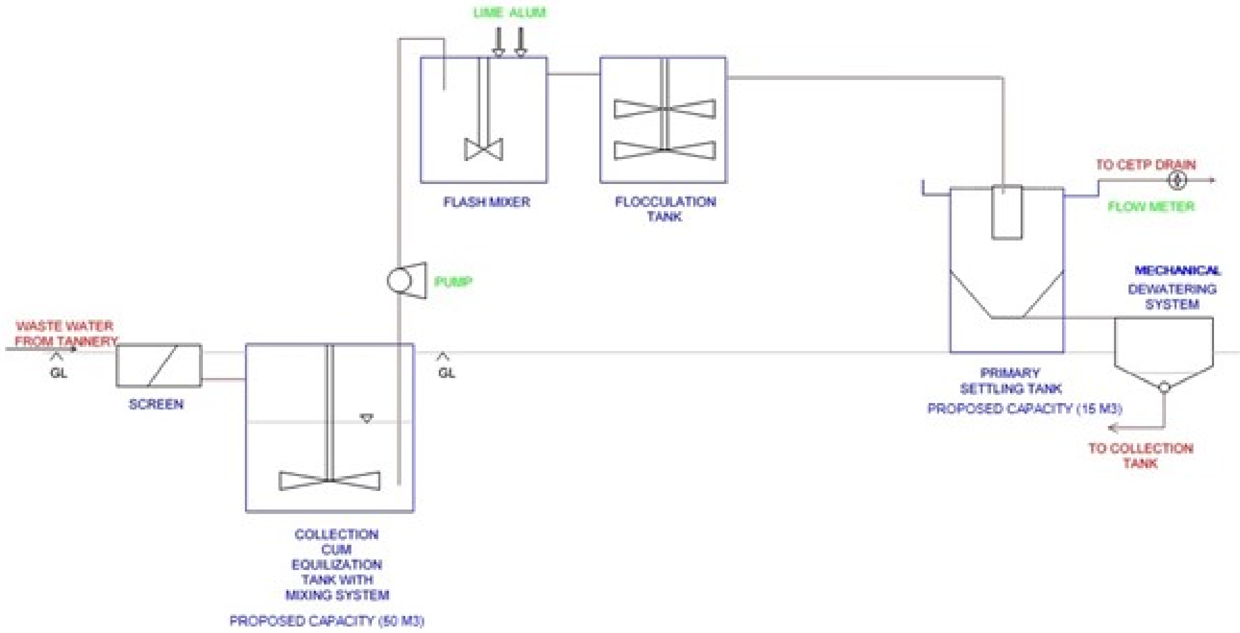
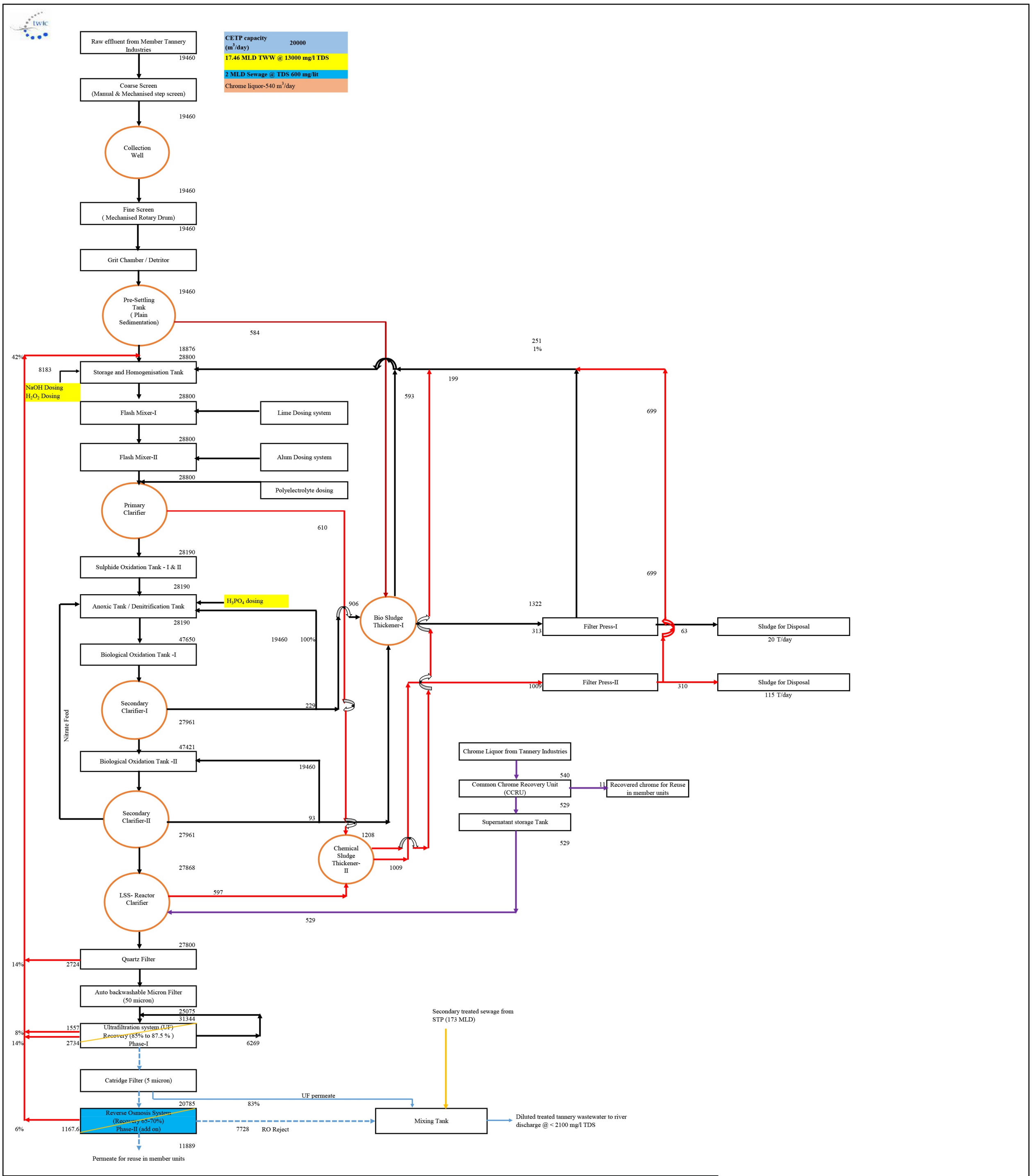


Figure H.2: Simplified flowchart of a Jajmau tannery effluent treatment plant



Legend

Raw effluent line/Treated effluent	
Chemical sludge/thickener overflow/filtrate	
Chrome supernatant	
Bio sludge/ Thickener overflow/filtrate	
Phase-I Components	
Phase-II Components	

This Process Flow Diagram (PFD) contains confidential property information of Tamilnadu Water Investment Company Limited, any unauthorised exhibition or copying of this material is strictly prohibited unless otherwise authorised in writing by TWIC, Chennai.

WATER BALANCE				
Recovery	RO Recovered water for reuse in member Tannery units of CETP	m³/day	11889	59.45%
Losses	RO Reject for disposal	m³/day	7728	38.64%
	Bio sludge to SLF	m³/day	63	0.31%
	Chemical sludge to Cement Industry	m³/day	310	1.55%
	Recovered chrome for reuse at member Tannery units	m³/day	11	0.05%
TOTAL		**m³/day**	**20000**	**100.00%**

NAME OF THE CLIENT		*NATIONAL MISSION FOR CLEAN GANGA (NMCG)*					
TITLE		*PRCOESS FLOW DIAGRAM WITH WATER BALANCE FOR A 20 MLD TANNERY BASED CETP (Up to UF) AT JAJMAU, KANPUR*					
DRG. NO		*NMCG/TAN/JAJ/KNR/CETP/TTP/2017*					*SHEET*
1	*NTS*	*01.11.2017*					
REV	*SCALE*	*DATE*	*DRAWN BY*	*GEN. BY*	*REVWD. BY*	*APPD. BY*	*ISSD. BY*

Figure H.3: Process Flow Diagram new 20 MLD CETP, Jajmau (18 MLD Tannery waste water, 2 MLD Sewage (Jajmau Tannery Effluent Treatment Association (JTETA), 2018)

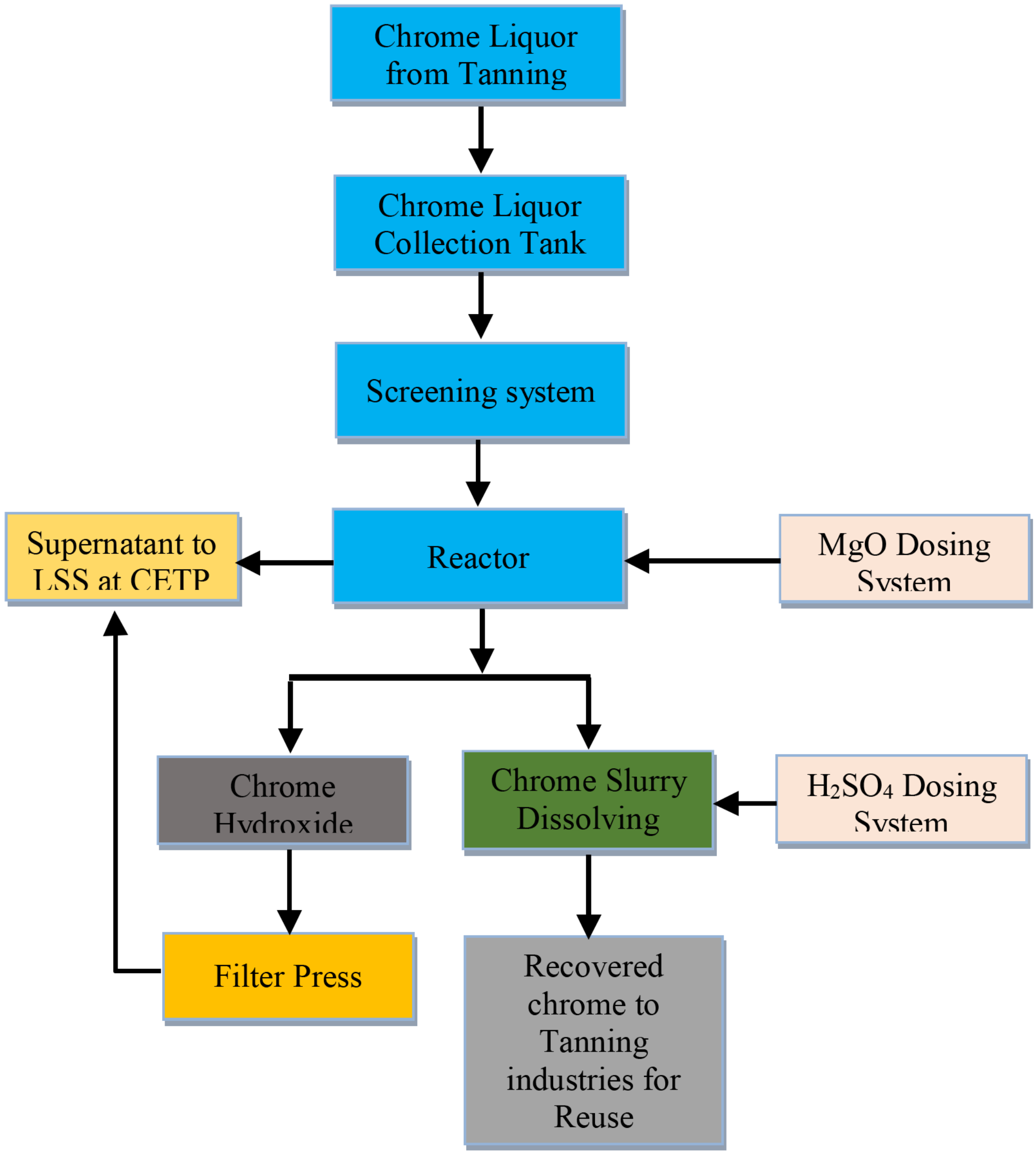


Figure H.4: Process Flow Diagram new 900 KLD CCRU, Jajmau (Jajmau Tannery Effluent Treatment Association (JTETA), 2018)

I
Pollution Measurements

I.1. Pollution levels Raw Effluent

Table I.1: An example of average total pollution load – concentration in combined raw effluent, conventional process, water consumption: 45 m^3/tonne (United Nations Industrial Development Organization (UNIDO), 2011)

Parameter	Unit	Average total pollution load	Typical limits, surface waters
BOD_5	mg O_2/l	2,000	30-40
COD	mg O_2/l	4,000	125-250
Suspended solids (SS)	mg/l	2,000	35-100
Cr^{3+}	mg Cr/l	150	1.5-2.0
S^{2-}	mg S/l	160	1.0-2.0
Total nitrogen (TKN)	mg N/l	160	100
Cl^-	mg Cl/l	5,000	Locally specific
SO_4^{2-}	mg SO_4/l	1,400	Locally specific
Oil and grease	mg/l	130	Locally specific
TDS*	mg/l	10,000	Locally specific
pH		6-9	5.5-9.5

I.2. Pollution Levels in CETP

Table I.2: Analysis Result of 36 MLD, CETP, Jajmau, Kanpur (Central Pollution Control Board, 2021)

Table1: Analysis Result of 36 MLD CETP, Jajmau, Kanpur

Parameters	Unit	Sample Location, Code and Standards											
		Tannery Inlet Channel (9MLD) (RD 1)	**PETP Outlet & Tannery Channel before mixing with domestic sewage (UPPCB Prescribed Standards)**	Domestic Sewage Inlet Channel (27 MLD) (RD 2)	CETP Inlet (after mixing of Tannery effluent with domestic sewage in ratio of 1:3) (RD 3)	**CETP Inlet (after mixing of Tannery effluent with domestic sewage in ratio of 1:3) (UPPCB Prescribed Standards)**	UASB Reactor O/L (RD 4)	CETP Outlet (RD 5)	**CETP Discharge Standards (MoEF& CC Notified)**		After Mixing of 130 MLD STP & CETP O/L at TEPH (RD 6)	Irrigation Canal (After mixing of all CETP & STPs) (RD 7)	**STP Discharge Standards as per NGT Order**
									Into inland surface water	On land for irrigation			
pH	-	7.46	**6.5 - 9.0**	7.4	7.6	**6.5 - 9.0**	7.68	8.1	**6.0 - 9.0**	**6.0 - 9.0**	-	8.57	**5.5 - 9.0**
TSS	mg/l	7701	**600**	411	1005	**600**	570	146	**100**	**100**	-	117	**20**
TDS	mg/l	10403	-	932	4964	-	4449	4464	-	-	2023	1795	-
FDS	mg/l	9619	-	771	4516	-	4080	4174	**2100**	**2100**	1822	1602	-
Chloride	mg/l	-	-	-	-	-	-	1990	**1000**	**1000**	-	627	-
Sulphide	mg/l	-	-	-	-	-	-	182	**2**	**2**	-	34	-
Total Phosphorus	mg/l	-	-	BDL*	-	-	-	-	-	-	-	BDL*	**1**
Total Nitrogen	mg/l	-	-	107.84	-	-	-	-	-	-	-	89.91	**10**
Oil & Grease	mg/l	-	-	-	-	-	-	8.8	**10**	**10**	-	13.8	-
COD	mg/l	-	-	321	1326	-	861	431	**250**	**250**	-	212	**50**
BOD	mg/l	-	-	104	492	-	310	194	**30**	**100**	-	76.4	**10**
Total Chromium	mg/l	132	**10**	3.6	16.8	**4**	-	8.4	**2**	**2**	-	6.6	-
Fecal Coliform	MPN/ 100 ml	-	-	-	-	-	-	-	-	-	-	4.5×10^2	**230**

*BDL means < 0.5 mg/l.

23

Page **2** of 5

J
Tanning Sector

J.1. Tannery Process

The first step in the production of leather is the slaughter process from which the hides are obtained. The important question is whether the skins can be immediately processed, such fresh skins are called "Green Skins" in jargon. However most of the time, and definitely the case in Jajmau, the skins are first preserved. This preservation of skins is known as *curing*.

J.1.1. Curing

These skins are preserved using large quantities of sodium chloride salt (NaCl) dehydrating them of all moisture; a skin of 40 to 50 kilogram requires approximately 20 to 25 kilogram of salt. Both wet and dry salting is possible, resulting in 'Salted Skins'. This salt, once used, is difficult to re-use due to the need of purity. Impurities lead to worse conservation of the skins. The treatment of these salts is often a major issue, and often just dumped somewhere. These salted skins are also prone to vermin infestation, so careful dry storage is important.

J.1.2. Pre-Treatment

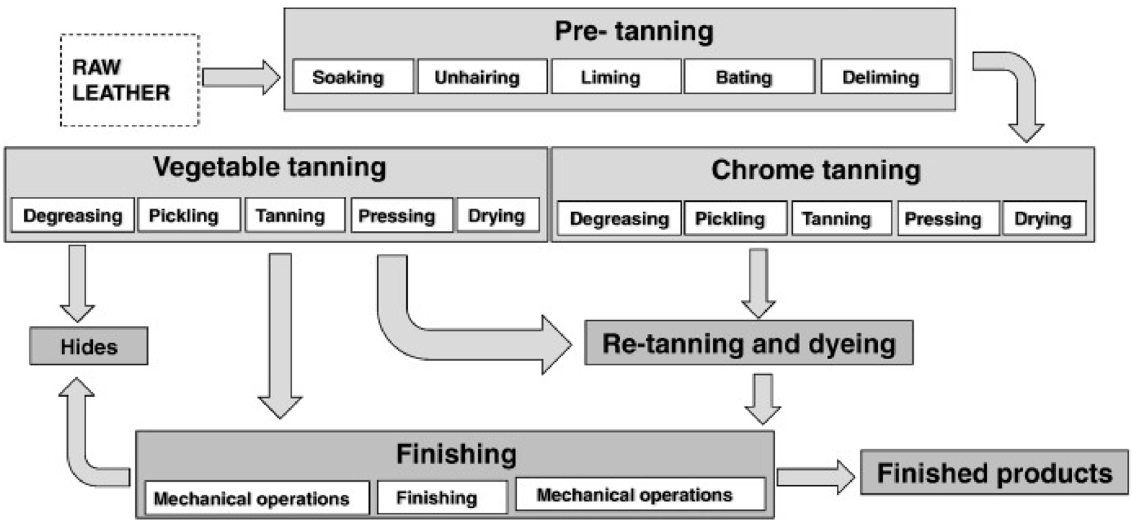


Figure J.1: A typical process flow sheet in an integrated leather tannery industry (Lofrano et al., 2013)

Salted skins come in with flesh, fat and hair still attached to them. The first step than is the pre-tanning done in the "beamhouse" (Lofrano et al., 2013). The hides will be soaked in an alkali solution ("Lime water", which is a saturated calcium hydroxide solution ($Ca(OH)_2(aq)$)) making them appropriate for further treatment later in the process. Lime alone takes a long time (6 days) to loosen the top hair layer that has to be removed, the so-called epidermis. Addition of sodium sulfide (Na_2S) or sodium hydrosulfide ($NaHS$) drastically accelerates dehairing by reductive cleavage (Reich, 2005). This has become standard practice but also increases the COD of the wastewater. After this soaking and liming step hair, fat and flesh is removed and the useless edges of the skin are cut. The alkali solution is washed out and the hides are brought to a proper acidic pH level. Sometimes further steps using enzymatic agents is done here for further pre-treatment.

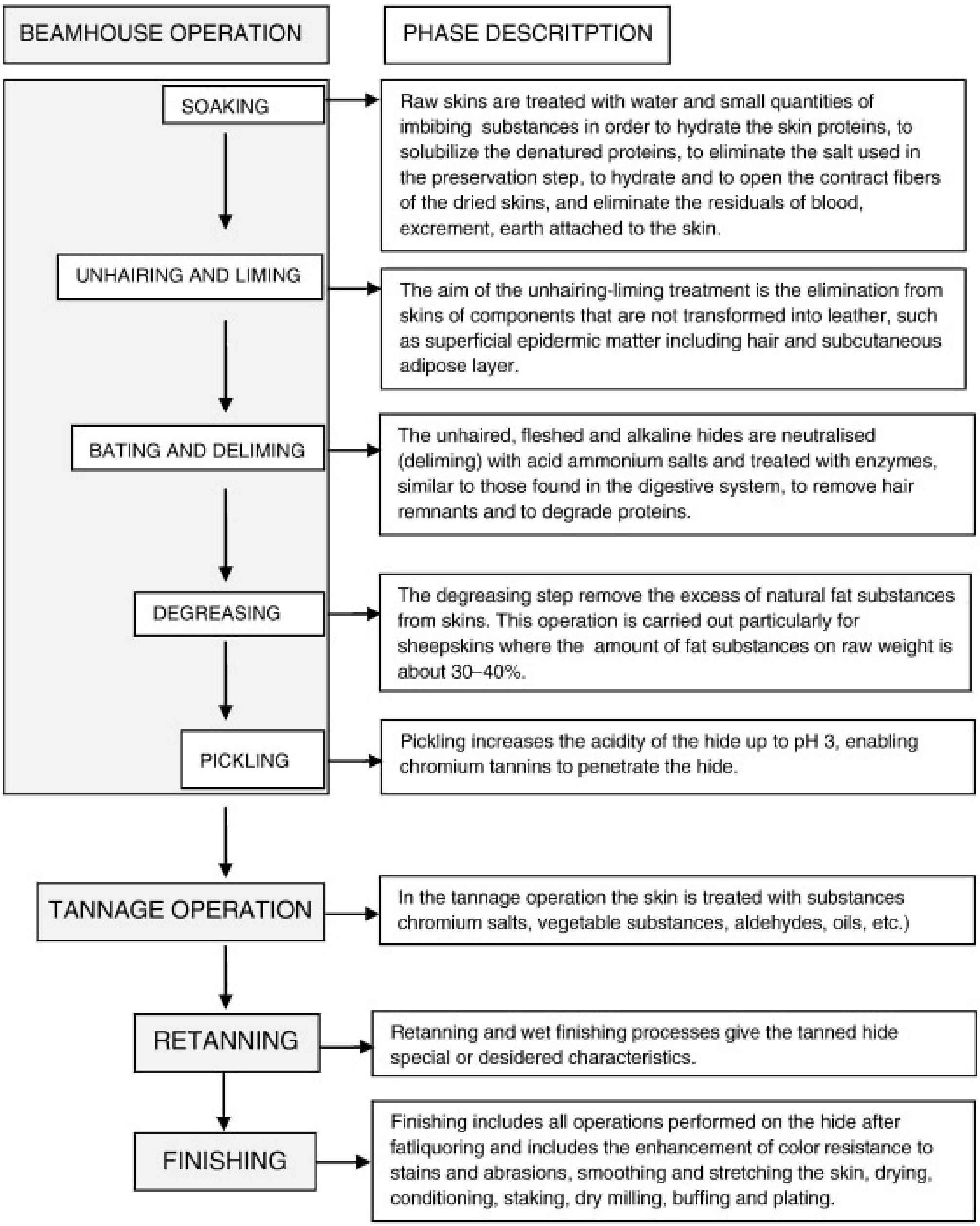


Figure J.2: Description of unit processes in leather production (Lofrano et al., 2013)

J.1.3. Bating and Deliming, Degreasing and Pickling

Deliming finally is necessary to remove the Ca^{2+} ions from the skin. This is done using mild acids like lactic acid ($CH3CH(OH)COOH$) or ammonium salts ($R^{-}NH_4^{+}$). If necessary a degreasing step is inserted, depends on the skins processed. Sheepskins in particular need to be degreased. Than the question arises what organic solvents are used to degrease the skins.

The last pre-treatment step is the pickling of the skins. The pelt is treated for several hours with dilute sulfuric acid (H_2SO_4) or formic acid (H_2CO_2) with addition of salt ($NaCl$) to prevent swelling of collagen and lower the pH. This is important for the next step; Tanning

Tanning
Tanning is the stabilising of the collagen fibres in the skins. This ensures that the skin remains flexible and does not decay. If tanning does not occur one gets parchment. The stabilising of the collagen fibres is done using chemicals called *tannins*. There are two types of tanning; reversible and irreversible tanning. The first type of tanning can be washed out whereby the properties of the leather are lost. Therefore, most of these reversible methods have been replaced with one or a combination of the three existing irreversible methods:

- Chrome Tanning using chrome containing agents.
- Vegetable Tanning or Natural Tanning using tannins derived from plants and trees, for example oak.
- Synthetic Tanning using compounds such as formaldehyde, glutaraldehyde, phenols and acrylates (Often used in combination with the other two, due to otherwise inferior quality products)

The tanned leather skins of these processes can be recognized by their colour. Chrome tanning is recognizable by the blueish skins it produces, known as *"wet-blue"*. The synthetic tanning procedure produces yellowish/white skins; known as *"wet-white"*. The vegetable tanning process produces brown skins. In the Jajmau region, as in the rest of the world, chrome tanning is dominant (Lofrano et al., 2013).

Regarding the stages of leather production in Jajmau the actor of Solidaridad (Tannery) mentioned the following:

"So, there are three or four different stages where the leather is processed. The first, the basic, more water consuming is called wet blue making. These wet blue makers in Jajmau area all are working for the bigger tanneries as job workers. The big units, they supply the leather chemicals, even the processing recipes. These people only just follow the recipes and can work the raw materials into wet blue, and then supply it back." "Leather, when you take the rawhide, till you make it as a shoe or upholstery or garment, it has to be processed in various stages. The first two ones are called raw to wet blue, where the more water, hair removing, all the excess substances will be removed, tanning will be this chromium and also will be done. This is called raw to wet blue. Then we have blue to crust, then crust to finishing. These are the three stages the tanning operations will be there. After finished leather, then the product will go to the product factory where they make shoes and garments, bags."

Therefore chrome tanning is assumed to be the standard process in the Jajmau area.

J.1.4. Chrome Tanning
The acidic pre-treated skins are put in a drum and then mixed with chromium tanning agents. Chrome tanning agents are usually a mixture of chromium sulfate ($Cr_2(SO_4)_3$) and chromium(III) oxide (Cr_2O_3). Mostly stored as a green powder. During the process the pH of the environment is slowly raised, often using so-called basifying agents as magnesium oxide (MgO) or sodium bicarbonate ($NaHCO_3$).

J.2. Alternative Generation Tannery Domain
From the most important steps in the leather production outlined above some questions arose regarding the Jajmau cluster. These question have to be answered with experts on both the current situation in Jajmau and with those with state-of-the-art knowledge of the existence and impacts of certain green-technology.

1. Regarding on the (re-)use and waste treatment of curing and pickling salts. Are high saline levels of the wastewater as result from the used salt in the process an issue?
2. If so, are there alternatives to decrease or change the current practices for ecological improvement.

3. Regarding the use of lime water there are health questions. How is current calcium hydroxide ($Ca(OH)_2(s)$) stored and handled in the production of lime water, and does this affect the workers in current practices? Unprotected exposure to $Ca(OH)_2(s)$ can cause severe skin irritation, chemical burns, blindness, lung damage or rashes (Fisher Science, 2014).
4. How is the current basic lime water treated after use and how does current discharge affect water pollution levels?
5. Are alternatives considered to the use of lime water and what would be the impact of these?
6. Regarding the use of dehairing sulfides, what extend is this used and does this create COD pollution?
7. What alternatives exist to come to ecological better dehairing processes? For example enzymatic dehairing?
8. The storage of sulfides (sodium sulfide (Na_2S) and/or sodium hydrosulfide ($NaHS$)) for dehairing purposes can be dangerous when it can come into contact with acidic environment creating $H_2S(g)$; how does this effect the tannery workers?
9. What is the impact of dilute sulfuric acid (H_2SO_4) or formic acid (H_2CO_2) in the pickling phase on the waste water discharge?
10. Are there environmental more friendly ways to reduce pollution load for acidification in the pickling step?
11. In the chrome tanning process, how much excess chrome is used that is left in the process water.
12. Is there a feasible alternative to decrease or eliminate chrome in the current tanning process? For example use enzymes or combine with synthetic tanning in first tanning or re-tanning step?

In short, for each use of chemical, there were questions on what the current impact is and, consequently, are there alternatives to each step? The discussion with experts on these questions led to the formulation of alternatives suitable for the Jajmau tannery industry. Also the impacts of these alternatives are derived from discussion with these experts.

K
Code

In this appendix the portfolio creation code and markdown descriptions are represented. They are exported from Jupyter Notebook files.

Any code from this appendix is also stored at GitHub, see Repo (https://github.com/pcjmeerman/PDA_ANALYSIS_CODE). If someone wants to use this code contact me or copy paste it from the files below.

The first section explains how the code creates the portfolios as explained in the method section.

Appendix Code

March 16, 2022

1 Implementation

1.1 Data and Uncertainty

First obtain a matrix with the information of alternatives on attributes, including the uncertainty ranges

[ ]:
```
import pandas as pd
import numpy as np
import time
import itertools as it
from scipy.stats import norm
from warnings import warn
from matplotlib import pyplot as plt
import seaborn as sns
import math

class Dataset():


    def __init__(self, dataset, sheetname_matrix):
        self.dataset = dataset
        self.sheetname_matrix = sheetname_matrix

    def read_dataset(self):
        return pd.read_excel(self.dataset, self.sheetname_matrix)

    def isolate_alternatives(self):
        self.alternatives_array = self.read_dataset().values[:,0].astype(int)
        return self.alternatives_array

    def column_names(self):
        return list(self.read_dataset().columns)

    def criteria_weights(self, sheetname_criteria = "Criteria"):
        return pd.read_excel(self.dataset, sheetname_criteria).values

    def uncertainty(self, sheetname_uncertainty = "Deviation"):
```

```
        return pd.read_excel(self.dataset, sheetname_uncertainty)

    def metrics(self, sheetname_metrics =  "Units"):
        return pd.read_excel(self.dataset, sheetname_metrics)

    def scales_gather(self, sheetname_metrics =  "Units"):
        return pd.read_excel(self.dataset, sheetname_metrics).iloc[[1,2]]

    def interactions(self, sheetname_interactions =  "Interactions"):
        return pd.read_excel(self.dataset, sheetname_interactions)

Complete_Matrix = Dataset("C:/Users/paulu/Documents/Epa/Thesis/Complete_Matrix.
 ↪xlsx", sheetname_matrix ='Sheet2')
print("Data loaded")

Complete_Matrix.criteria_weights()
```

1.1.1 Generating Box plot representation of the impacts of the alternatives using Uncertainty_Alternatives Class

In order to present the uncertainty of the impacts of an alternative the Class "Uncertainty_Alternatives()" has been written. The distribution of the uncertainty can have different forms, but since that information is lacking a normal distribution is assumed. If another form of distribution is used this will be noted. In order to be able to generate the boxplots the corresponding deviations, have to be filled in in the excel file. For this there is a sheetname called "Deviation" and on the corresponding places of the means (same position on the grid in the other sheet), the standard deviations have to be filled in. For now often assumed to be 10% of the mean value.

From this range a normal distribution is to be generated in the methods of the class setting the numpy.random.seed(0) and the using np.random.normal methods. the outcome is a set of data used to represent the uncertainty in the boxplots. However, if the alternatives generate no impact == 0, on an attribute, for clarity sake, no boxplot is generated. (This would represent just a line on the x-axis).

If one alternative on one attribute has to be generated one can use the boxplot() method. Here alt = the alternative to be represented in integer format, n = 150, the amount of random generated data is used for the boxplot and the attribute is the name of the attribute to be represented in string format.

The process is the same for the other methods, "alternatives()" and "all_alternatives()", which present all the alternative on 1 attribute and all alternatives on all attributes, respectively.

[ ]:
```
class Uncertainty_Alternatives():
    def __init__(self, dataset = Complete_Matrix):
        self.dataset = dataset
        self.uncertainty_df = dataset.uncertainty()
        self.alternatives_df = dataset.read_dataset()
        self.temp_df = pd.DataFrame()
        sns.set(rc={"figure.figsize":(7, 5)})
```

```
        sns.set_style('whitegrid')


        #If you have filled in the deviations of an alternative in log-normal␣
↪distributions values you can add these to this list for
        #that particulare attribute.
        #the code will draw n random numbers from a lognormal distributions␣
↪using the given mean and sigma from the excell file and not use a normal␣
↪distribution.
        #Be aware that in the alternative_uncertainty method, the mean value is␣
↪put in a log function; So you don't have to do that in the excell file␣
↪itself.
        #however, instead of the standard deviation in the "Deviation Tab" you␣
↪need to fill in the sigma function.
        self.lognormal_list_alt = []
        self.uniform_list_alt = []

        return

    def boxplot(self, alt, n = 100000, attribute = "Attribute_3"):
        if alt < 1:
            return "alternatives number must be greater than zero, otherwise␣
↪things go wrong, trust me, than it starts from the other side of the matrix"
        np.random.seed(0)
        if alt not in self.lognormal_list_alt and alt not in self.
↪uniform_list_alt:
            print("Normal distribution found for alternative:", str(alt))
            Distribution_generation = np.random.normal(loc=self.
↪alternatives_df[attribute].values[alt-1], scale=self.
↪uncertainty_df[attribute].values[alt-1], size = n)
            if np.sum(Distribution_generation) == 0.0:
                print("Impact of Alternative ", str(self.
↪alternatives_df["Name"].values[alt-1]), "is zero on the selected attribute")
                return
            ax = sns.boxplot(y = Distribution_generation, showfliers = False)
            ax.set_title("Impact on " + attribute)
            ax.set_xlabel("Alternative " + str(alt) + ": "+ str(self.
↪alternatives_df["Name"].values[alt-1]));
            ax.set_ylabel(self.dataset.metrics()[attribute].values[0])
            print("Scale = " + str(self.uncertainty_df[attribute].
↪values[alt-1]))
            print("Loc = "+ str(self.alternatives_df[attribute].values[alt-1]))
            #ax.set(ylim=(8, 16))
            plt.show()

            return
```

```
        if alt in self.lognormal_list_alt:
            #0.03*self.uncertainty_df[attribute].values[alt-1]
            self.sigma = 1.2
            print("Log-normal distribution found for alternative:", str(alt))
            Distribution_generation = np.random.lognormal(mean=math.log(self.
↪alternatives_df[attribute].values[alt-1]), sigma=self.sigma, size = n)
            if np.sum(Distribution_generation) == 0.0:
                print("Impact of Alternative ", str(self.
↪alternatives_df["Name"].values[alt-1]), "is zero on the selected attribute")
                return
            ax = sns.boxplot(y = Distribution_generation, showfliers = False)
            ax.set_title("Impact on " + attribute)
            ax.set_xlabel("Alternative " + str(alt) + ": "+ str(self.
↪alternatives_df["Name"].values[alt-1]));
            ax.set_ylabel(self.dataset.metrics()[attribute].values[0])
            #ax.set(ylim=(8, 16))
            #self.uncertainty_df[attribute].values[alt-1]
            print("Sigma = " + str(self.sigma))
            print("Mean = "+ str(self.alternatives_df[attribute].values[alt-1]))
            plt.show()

        if alt in self.uniform_list_alt:
            #0.03*self.uncertainty_df[attribute].values[alt-1]
            self.low = self.alternatives_df[attribute].values[alt-1] - self.
↪uncertainty_df[attribute].values[alt-1]
            self.high = self.alternatives_df[attribute].values[alt-1] + self.
↪uncertainty_df[attribute].values[alt-1]
            print("Uniform distribution found for alternative:", str(alt))
            Distribution_generation = np.random.uniform(low=self.low, high =␣
↪self.high, size = n)
            if np.sum(Distribution_generation) == 0.0:
                print("Impact of Alternative ", str(self.
↪alternatives_df["Name"].values[alt-1]), "is zero on the selected attribute")
                return
            ax = sns.boxplot(y = Distribution_generation, showfliers = False)
            ax.set_title("Impact on " + attribute)
            ax.set_xlabel("Alternative " + str(alt) + ": "+ str(self.
↪alternatives_df["Name"].values[alt-1]));
            ax.set_ylabel(self.dataset.metrics()[attribute].values[0])
            #ax.set(ylim=(8, 16))
            #self.uncertainty_df[attribute].values[alt-1]
            print("Low = " + str(self.low))
            print("High = "+ str(self.high))
            plt.show()
```

```python
    def distributions(self, alt = 3, n = 100000, attribute = "Attribute_3"):
        if alt < 1:
            #print("alternatives number must be greater than zero, otherwise␣
↪things go wrong, trust me, than it starts from the other side of the matrix")
            return
        if alt not in self.lognormal_list_alt and alt not in self.
↪uniform_list_alt:
            np.random.seed(1)
            Distribution_generation = np.random.normal(loc=self.
↪alternatives_df[attribute].values[alt-1],scale=self.
↪uncertainty_df[attribute].values[alt-1], size = n)
            if np.sum(Distribution_generation) == 0.0:
                #print("Impact of Alternative " + str(self.alt) + " " +␣
↪str(self.alternatives_df["Name"].values[self.alt-1]), "is zero")
                return
            self.temp_df["Alt_" + str(alt)] = Distribution_generation
            return self.temp_df
        if alt in self.lognormal_list_alt:
            self.sigma = 0.2
           # 0.0135*self.uncertainty_df[attribute].values[alt-1]
            np.random.seed(1)
            Distribution_generation = np.random.lognormal(mean=math.log(self.
↪alternatives_df[attribute].values[alt-1]), sigma= self.sigma, size = n)
            if np.sum(Distribution_generation) == 0.0:
                #print("Impact of Alternative " + str(self.alt) + " " +␣
↪str(self.alternatives_df["Name"].values[self.alt-1]), "is zero")
                return
            self.temp_df["Alt_" + str(alt)] = Distribution_generation
            return self.temp_df

        if alt in self.uniform_list_alt:
            self.low = self.alternatives_df[attribute].values[alt-1] - self.
↪uncertainty_df[attribute].values[alt-1]
            self.high = self.alternatives_df[attribute].values[alt-1] + self.
↪uncertainty_df[attribute].values[alt-1]
           # 0.0135*self.uncertainty_df[attribute].values[alt-1]
            #print("Uniform distribution found for alternative:", str(alt))
            np.random.seed(1)
            Distribution_generation = np.random.uniform(low=self.low, high =␣
↪self.high, size = n)
            if np.sum(Distribution_generation) == 0.0:
                #print("Impact of Alternative " + str(self.alt) + " " +␣
↪str(self.alternatives_df["Name"].values[self.alt-1]), "is zero")
                return
            self.temp_df["Alt_" + str(alt)] = Distribution_generation
```

```
            return self.temp_df

    def alternatives(self, attribute = "Attribute_3"):
        self.temp_df = pd.DataFrame()
        sns.set(rc={"figure.figsize":(7, 5.5)})
        sns.set_style('whitegrid')
        for i in range(self.alternatives_df.values.shape[0]):

            #i becomes to large probably; look at alt = i+1
            self.distributions(alt = i+1, attribute = attribute)
        ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.
 ↪temp_df), showfliers = False, palette = "Blues")
        ax.set_title("Impact on: " + str(attribute))
        ax.set_xlabel("Alternatives")
        ax.set_ylabel(self.dataset.metrics()[attribute].values[0])

        return plt.show()

    def all_alternatives(self):
        for j in range(2, (len(self.dataset.column_names()))):
            self.alternatives(attribute = self.dataset.column_names()[j])
            plt.show()
        return

Uncertainty_Data_Alternatives = Uncertainty_Alternatives()
#Uncertainty_Data_Alternatives.alternatives_df.values.shape[0]
#Uncertainty_Data_Alternatives.boxplot(alt = 4, attribute =␣
 ↪'Chrome_concentration_in_Crops')
#Uncertainty_Data_Alternatives.boxplot(alt = 5, attribute =␣
 ↪'Chrome_concentration_in_Crops')
#Uncertainty_Data_Alternatives.boxplot(alt = 8, attribute =␣
 ↪'Chrome_concentration_in_Crops')
#test = Uncertainty_Data_Alternatives.boxplot(alt = 2, attribute =␣
 ↪"Attribute_8")
```

1.1.2 Generate the Solution Space

To generate the solution space all alternative combinations are generated. Some alternatives are mutually exclusive or dependent, this information is than also added. For example, alternative 1 is the BUA case and is exclusive with all alternatives. Therefore this alternative is isolated before the solution space (“def solspace()”) is generated.

[ ]:
```
class Explicit_Space():
    def __init__(self, choices, alternative_set = Complete_Matrix):
        self.n = choices
        self.set = alternative_set.isolate_alternatives()
        self.subseq = []  # subsequent activities
```

```
        self.mut_exc = []  # mutually exclusive
        self.solution_space = "Not yet generated"

    def add_mutually_exclusive(self, a, b):

        # if a >= self.n:
        #     _msj = ('a is out of bounds. Maximum value is {0} and \
        #             got {1}'.format(self.n, a))
        #     raise ValueError(_msj)

        # if b >= self.n:
        #     _msj = ('b is out of bounds. Maximum value is {0} and \
        #             got {1}'.format(self.n, b))
        #     raise ValueError(_msj)

        _ac = [a, b]
        _ca = [b, a]
        if _ac not in self.mut_exc or _ca not in self.mut_exc:
            self.mut_exc.append(_ac)
        else:
            warn('Mutualy exclusive activity already added')

    def add_subsequent(self, parent, sub):

        #if parent >= self.n:
         #    _msj = ('parent is out of bounds. Maximum value is {0} and \
         #            got {1}'.format(self.n, parent))
          #   raise ValueError(_msj)

        #if sub >= self.n:
        #     _msj = ('sub is out of bounds. Maximum value is {0} and \
         #            got {1}'.format(self.n, sub))
         #   raise ValueError(_msj)

        _ac = [parent, sub]
        if _ac not in self.subseq:
            self.subseq.append(_ac)
        else:
            warn('Subsequent activity already added')

    def generate(self):
        for m in range(self.n+1):
            for item in it.combinations(self.set, m):
                #print(item)
                _flag = True
                #for si in self.subseq:
                  #  if si == si:
```

```
                        #_flag = False
                        #print(_flag)
                  #       break
                   # print("Dependecy found")
            # Check for mutually exclusive



                for si in self.subseq:
                    if si[0] in item and si[1] not in item:
                        _flag = False
                        break

                if _flag:
                    for ei in self.mut_exc:
                  # print((ei[0], ei[1]))
                        if  ei[0] in item and ei[1] in item:
                        #print(item)
                        #print(ei[0], ei[1])
                            _flag = False
                            break
                        #print(_flag)


                if _flag:
                    yield item

    def solspace(self):
        list = []
        for item in self.generate():
            list.append(item)
        self.solution_space = list
        return list

    def size(self):
        return len(self.solspace())

print("Explicit Class created")
```

1.1.3 Data Acquisistion

From the generated solution space the corresponding information of each alternative on each attribute is collected in the class "Data_Acquisition"

[ ]:
```
class Data_Acquisition():
    def __init__(self, dataset = Complete_Matrix, n = 1,):
        self.dataset = dataset
```

```
        self.n = n
        self.stop = False
        self.collected_data = "No Data Generated as of yet"
        self.temp = Explicit_Space(self.n, self.dataset)

        #Fill in what mutually exclusive project there are:
        #For now, alternative one represents the BAU case.
        for i in range(2,self.dataset.alternatives_array.shape[0]):
            self.temp.add_mutually_exclusive(1,i)
        self.temp.add_mutually_exclusive(2,9)
        self.temp.add_mutually_exclusive(3,9)
        self.temp.add_mutually_exclusive(4,9)
        self.list_combinations = self.temp.solspace()
        self.collected_df = "No Data Generated as of yet"
        self.df_aggregated = "No Data Generated as of yet"

    def collect(self):
   # array = np.asarray(decision_matrix)
    #print(decision_matrix)
    #print(list_combinations)
        decision_matrix = self.dataset.read_dataset()
        list_combinations = self.list_combinations

    #First create an empty set in which the different alternatives are stored
        Collected_sets = []
    #iterate trought the combinations created by the Explicit_Space Class.
        for i in list_combinations:
            if self.stop == True:
                if len(Collected_sets) > self.portfolio_set:
                    self.stop = False
                    return Collected_sets
        #From the original dataset each row with al the data has to be selected/
↪collected.
        #For this first an empty set is made to obtain all the information for␣
↪each combination set and store this one in T_set
            T_set = []
        #For each element, j, in the ith set of combination the information is␣
↪obtained.
        #If element i = 4, contains the Alternatives, 5, 6, 19 and 22 than for␣
↪these the information has to be collected
            for j in i:
            #The information is collected by searching trought the original␣
↪data set (decision_matrix) by compairing the Alternative numbers

                A_selected = decision_matrix.loc[decision_matrix[self.dataset.
↪column_names()[0]] == j]
```

```
            #For collection purposes the pandadataframe is converted to a␣
↪simple list.
            #The [0] in the next section is needed to get the list out of the␣
↪lists.
            #To many lists [[[5]]].
                A_converted = A_selected.values.tolist()[0]
                A_converted[0] = int(A_converted[0])
            #The values are appended to the list until all combinations within␣
↪the ith set are added.
                T_set.append(A_converted)
            #After this the set is added to the collected set and reset for the␣
↪next ith combination.
            #It is thus a temporary set.
        #print(T_set)
            Collected_sets.append(T_set)
    #print(Collected_sets)
        #print("For ", self.n, " action(s) to chooce from the dataset, the␣
↪combinations are:")
        self.collected_data = Collected_sets
        return self.collected_data

    def collect_dataframe(self):
        df = []
        for i in range (1, self.temp.size()):
            test = (self.collected_data[i])
            dataframe = pd.DataFrame(test, columns = self.dataset.
↪column_names())
            dataframe["Portfolio_set"] = i
            dataframe_temp = dataframe
            df.append(dataframe_temp)
            self.collected_df_seperated = df
            self.collected_df = pd.concat(self.collected_df_seperated)
        #print("For portfolio number", self.portfolio_set, "we obtain:")
        self.stop = True
        return self.collected_df
        #self.collected_df = pd.DataFrame((Collected_sets), columns = self.
↪dataset.column_names())


    def aggregate(self):
        self.sp_temp = []
        df_23=[]
        for j in range (1, self.temp.size()):
            self.sp = self.collected_df.loc[self.collected_df['Portfolio_set']␣
↪== j]
            self.sp_temp.append(self.sp)
```

```
        for i in range(len(self.list_combinations)-1):
            result = self.sp_temp[i]
            result.pop("Portfolio_set")
            result.pop("Alternative")
            result.pop("Name")
            result_2 = result.sum(axis = 0).to_frame().T
            #check wheter there are attributes expressen in percentages in the dataset that have to be calculated differently.
            for k in self.sp_temp[i].T.index:
                if k == "Alternative" or k == "Name" or k == "Portfolio_set":
                    continue
                if Complete_Matrix.metrics()[str(k)][0] == "%":
                    #print(k + " is expressed in percentages")
                    temp_value = np.asarray(self.sp_temp[i].T.loc[k]/100)
                    new_value = temp_value.sum() + np.prod(temp_value)
                    result_2[k] = result_2[k].replace(result_2[k][0],new_value*100)
            df_23.append(result_2)
        self.df_aggregated = pd.concat(df_23)
        self.df_aggregated["Strategic_Portfolio"] = list(range(1, self.df_aggregated.shape[0]+1))
        for z in self.df_aggregated.columns:
            if z == "Strategic_Portfolio":
                continue
            if Complete_Matrix.metrics()[z][1] == 'global':
                #print("local attribute found: ", z, eval(Complete_Matrix.metrics()[z][2])[1])
                self.df_aggregated.loc[self.df_aggregated[z] > eval(Complete_Matrix.metrics()[z][2])[1], z] = eval(Complete_Matrix.metrics()[z][2])[1]
                self.df_aggregated.loc[self.df_aggregated[z] < eval(Complete_Matrix.metrics()[z][2])[0], z] = eval(Complete_Matrix.metrics()[z][2])[0]


        return self.df_aggregated
        return self.df_aggregated


    def single_portfolio(self, portfolio_set = 1):
        if self.stop == False:
            self.collect_dataframe()
        selected_portfolio = self.collected_df.loc[self.collected_df['Portfolio_set'] == portfolio_set]
        return selected_portfolio
```

```
    def aggregated_single_portfolio(self, portfolio_set = 2):
        return self.aggregate_old()[portfolio_set-1:portfolio_set]


    def one_value(self, weight_1 = 0.5 , weight_2 = (1/3), weight_3 = (1/6)):
        store=self.aggregate_old()
        return pd.
↪DataFrame(store["Crit_1"]*weight_1+store["Crit_2"]*weight_2+store["Crit_3"]*weight_3
    , columns = ["Crit_total"])

    def run(self):
        self.collect()
        self.collect_dataframe()
        self.aggregate()


Portfolio_Data = Data_Acquisition(Complete_Matrix, 5)
#Portfolio_Data.dataset
Portfolio_Data.run()
print("Portfolio Data Collected")

#Portfolio_Data.collected_df
#Portfolio_Data.single_portfolio(91)
#Portfolio_Data.df_aggregated
```

[ ]:
```
#runtime_plot.drop(runtime_plot.tail(2).index,inplace=True)
#sns.lineplot(data = runtime_plot, x = "Amount of Portfolios", y = "Run time␣
↪(s)").set(title='Runtime Portfolio Creation [14 Alternatives]')

#plt.show()
```

1.1.4 Uncertainty in Portfolios

In order to analyse the performances of the portfolios the uncertainty of the performance has to be taken into account. There are two ways to approach this problem, but for both approaches the uncertainty distribution of the alternatives themselves have to be available for the attributes. Before aggregation of the alternative values normalisation of both the values and the uncertainty has to take place.

Since a portfolio is a set of alternatives the uncertainties are propogated. One way is to add the uncertainties analytically. The difficulty of this approach is then the situation where uncertainties that are distributed differently have to be combined. The other way is computationally, which is the way taken here. A Monte Carlo simulation is done for the portfolios. This encompasses selecting a value within each distribution of each alternative. These uncertainty distributions have many forms. Then these, for example 5 values are then aggregated as the portfolio value. This process is then iterated many times to approach the uncertainty distribution of the portfolio. From these aggregated values the boxplot performance of a portfolio is then presented.

Graphically presenting the performances of all portfolios simultaniously is difficulty since these will be 257 portfolios to be shown. So from the analysis best performing

[ ]:
```
class Uncertainty_Portfolios():
    def __init__(self, dataset_portfolio = Portfolio_Data, dataset_alternatives␣
↪= Complete_Matrix):
        self.dataset = dataset_portfolio
        self.uncertainty_df = dataset_alternatives.uncertainty()
        self.alternatives_df = dataset_alternatives.read_dataset()
        self.dataset_alternatives_col_names = dataset_alternatives.
↪column_names()
        self.temp_df = pd.DataFrame()
        sns.set(rc={"figure.figsize":(4, 4)})
        sns.set_style('whitegrid')
        self.list_combination = dataset_portfolio.list_combinations
        self.temp_df_2 = []
        self.attribute = "not defined"

        #If you have filled in the deviations of an alternative in log-normal␣
↪distributions values you can add these to this list for
        #that particulare attribute.
        #the code will draw n random numbers from a lognormal distributions␣
↪using the given mean and sigma from the excell file and not use a normal␣
↪distribution.
        #Be aware that in the alternative_uncertainty method, the mean value is␣
↪put in a log function; So you don't have to do that in the excell file␣
↪itself.
        #however, instead of the standard deviation in the "Deviation Tab" you␣
↪need to fill in the sigma function.
        self.lognormal_list_alt = []
        self.uniform_list_alt = []

    def alternative_uncertainty(self, attribute, alt, n =100000):
        np.random.seed(0)
        if alt not in self.lognormal_list_alt and alt not in self.
↪uniform_list_alt:
            np.random.seed(1)
            Distribution_generation = np.random.normal(loc=self.
↪alternatives_df[attribute].values[alt-1], scale=self.
↪uncertainty_df[attribute].values[alt-1], size = n)
            self.temp_df[str(alt)] = Distribution_generation
            return self.temp_df
        if alt in self.lognormal_list_alt:
            self.sigma = 0.2
            print("log-normal distribution found for alternative:", str(alt))
            np.random.seed(1)
```

```
            Distribution_generation = np.random.lognormal(mean=math.log(self.
↪alternatives_df[attribute].values[alt-1]), sigma=self.sigma, size = n)
            self.temp_df[str(alt)] = Distribution_generation
            return self.temp_df

        if alt in self.uniform_list_alt:
            print("log-normal distribution found for alternative:", str(alt))
            self.low = self.alternatives_df[attribute].values[alt-1] - self.
↪uncertainty_df[attribute].values[alt-1]
            self.high = self.alternatives_df[attribute].values[alt-1] + self.
↪uncertainty_df[attribute].values[alt-1]
           # 0.0135*self.uncertainty_df[attribute].values[alt-1]
            #print("Uniform distribution found for alternative:", str(alt))
            np.random.seed(1)
            Distribution_generation = np.random.uniform(low=self.low, high =␣
↪self.high, size = n)
            if np.sum(Distribution_generation) == 0.0:
                #print("Impact of Alternative " + str(self.alt) + " " +␣
↪str(self.alternatives_df["Name"].values[self.alt-1]), "is zero")
                return
            self.temp_df["Alt_" + str(alt)] = Distribution_generation
            return self.temp_df

    def portfolio_uncertainty(self, attribute =␣
↪"Time_underperforming_due_to_electricity_shortages", n = 100000,␣
↪portfolio_number = 5):
        #instead of the current list a portfolio has to be added
        np.random.seed(0)
        self.temp_df = pd.DataFrame()
        for i in self.list_combination[portfolio_number]:
            self.alternative_uncertainty(attribute = attribute, alt = i, n = n)
        Portfolio_data_boxplot = self.temp_df.values.sum(1)
        self.portfolio = pd.DataFrame(Portfolio_data_boxplot, columns =␣
↪["Portfolio number: " + str(portfolio_number)])
        return self.portfolio

    def plot_portfolio(self, portfolio_number = 1, attribute =␣
↪"Time_underperforming_due_to_electricity_shortages", n = 100000):
        if portfolio_number > (len(self.list_combination)-1):
            print("The requested Portfolio does not exist in this dataset, look␣
↪at how many portfolios you have generated:" + str(len(Portfolio_Data.
↪list_combinations)-1))
            return
        self.portfolio_uncertainty(portfolio_number = portfolio_number,␣
↪attribute = attribute, n = n)
```

```
        ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.
↪portfolio), showfliers = False, palette = "Greens")
        ax.set_title("Impact on: " + str(attribute))
        ax.set_xlabel(None)
        ax.set_ylabel(self.dataset.dataset.metrics()[attribute].values[0])
    #Fist obtain the portfolio combinations via explicit space,

    def plot_all_portfolios(self, attribute =␣
↪"Time_underperforming_due_to_electricity_shortages", n = 100000, exclude =␣
↪True, plot = True):
        self.attribute = attribute
        sns.set(rc={"figure.figsize":(11, 25)})
        sns.set_style('whitegrid')
        if len(self.temp_df_2) != 0:
            print("Distributions already created")
        else:
            self.temp_df_2 = pd.DataFrame()
            for i in range(1, len(self.list_combination)):
                if exclude == True:
                    if self.portfolio_uncertainty(attribute = attribute, n = n,␣
↪portfolio_number = i).values.sum(1)[0] == 0.0:
                        continue
                self.temp_df_2[str(i)]=(self.portfolio_uncertainty(attribute =␣
↪attribute, n = n, portfolio_number = i)["Portfolio number: " + str(i)])
            #if i == 15:
                #break
        if plot == True:
            ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.
↪temp_df_2), showfliers = False)
            ax.set_title("Impact on: " + str(attribute))
            ax.set_xlabel("Portfolios")
            ax.set_ylabel(self.dataset.dataset.metrics()[attribute].values[0])
        return

    def plot_all_portfolios_all_attributes(self, n = 100000):
        for j in range(2, (len(self.dataset_alternatives_col_names))):
            self.plot_all_portfolios(attribute = self.
↪dataset_alternatives_col_names[j])
            plt.show()
        return

    def plot_core_portfolios(self, n = 100000, core = [3, 5], attribute =␣
↪"Time_underperforming_due_to_electricity_shortages"):
        sns.set(rc={"figure.figsize":(11, 7)})
        sns.set_style('whitegrid')
        self.temp_df_core = pd.DataFrame()
```

```
        for i in core:
            if self.portfolio_uncertainty(attribute = attribute, n = n,␣
↪portfolio_number = i).values.sum(1)[0] == 0.0:
                continue
            self.temp_df_core[str(i)]=(self.portfolio_uncertainty(attribute =␣
↪attribute, n = n, portfolio_number = i)["Portfolio number: " + str(i)])
            #if i == 15:
                #break
        ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.
↪temp_df_core), showfliers = False)
        ax.set_title("Impact on: " + str(attribute))
        ax.set_xlabel("Portfolios")
        ax.set_ylabel(self.dataset.dataset.metrics()[attribute].values[0])
        return

    def plot_core_portfolio_all_attributes(self, core = [3, 5] ):
        for j in range(2, (len(self.dataset_alternatives_col_names))):
            self.plot_core_portfolios(core = core, attribute = self.
↪dataset_alternatives_col_names[j])
            plt.show()
        return

    def plot_one_portfolio_all_attributes(self, n = 100000, portfolio_number =␣
↪2):
        for j in range(2, (len(self.dataset_alternatives_col_names))):
            self.plot_portfolio(portfolio_number = portfolio_number, attribute␣
↪= self.dataset_alternatives_col_names[j])
            plt.show()

Boxplots = Uncertainty_Portfolios()
#Boxplots.plot_core_portfolios(core = Portfolios_with_Synergy.
↪portfolios_with_synergy["Strategic_Portfolio"].values)
#Boxplots.plot_core_portfolio_all_attributes(core = Portfolios_with_Synergy.
↪portfolios_with_synergy["Strategic_Portfolio"].values)
```

1.2 Value Functions

From the obtained portfolio the mean performances are converted to values: The first step to obtain these values is to create a value function. The value function presented here are linear value functions per attribute. The scales are either local or globally determined, depending on the relevant attribute. Secondly, the value functions are created: The portfolios generate a corresponding value per partial value function.

The third step is to aggregate the values of each partial value functiond. The fourth is to do this for the portfolio uncertainty distribution generated in the Portfolio_Uncertainty class to obtain the PDF of the values per attribute and then per aggregated portfolio.

```
[ ]: class Values():
    def __init__(self, dataset = Portfolio_Data, dataset_2 = Complete_Matrix):
        self.dataset_2 = dataset_2
        self.portfolio_data = dataset
        self.dataset = dataset.df_aggregated
        #
        #Indexes of the criteria that are deemed 'cost criteria'
        #For days attribute the global scales are minimal 0 days, maximal 365␣
↪days.
        #Per attribute these have to be defined in the excell file.
        self.scl = []
        self.attribute = "Profits_from_hides_processing"
        self.info = self.portfolio_data.dataset.scales_gather()
    def scales(self):
        #from the excell file the scale information is loaded:
        #If the scales are globally identified they are obtained from the given␣
↪input in the Excel.
        if self.info[self.attribute].iloc[0] == "global":
            print("A global scale has been identified for attribute: " + self.
↪attribute)
            self.scl = eval(self.dataset_2.scales_gather()[self.attribute].
↪iloc[1])
        #If it local variable the min and max are obtained from the␣
↪Portfolio_Dataset to obtain the ranges.
        #Since the ranges are dealing with point data and the portfolio sets␣
↪have a PDF the highest and lowest found datapoints
        #from the Monte Carlo simulations are taken as min and max values.
        else:
            print("A Local scale has been identified for attribute: " + self.
↪attribute)
            Boxplots_values = Uncertainty_Portfolios()
            Boxplots_values.plot_all_portfolios(self.attribute, exclude =␣
↪False, plot = False)
            self.temp = Boxplots_values.temp_df_2
            self.scl = [np.min(Boxplots_values.temp_df_2.values), np.
↪max(Boxplots_values.temp_df_2.values)]

        return


    def value_function(self, attribute =␣
↪"Space_requirement_for_Waste_Water_Treatment", p = 10):
        self.a_type = self.dataset_2.criteria_weights()[1][self.info.columns.
↪tolist().index(attribute)]
        #check wheter new scales have to be loaded for the Value class when new␣
↪attribute is chocen.
```

```
        if attribute != self.attribute:
            self.attribute = attribute
            self.scales()
        #first test wheter the scales are loaded, if not load them in via the␣
↪scales method.
        #if len(self.scl) == 0:
        #print('Scales are loaded')
        #For performance p for p element of self.scale:
        #For linear function
        if p< min(self.scl) or p>max(self.scl):
            print("Performance value, p, exceeds value function scale: p =",␣
↪str(p), "when range is"+str(self.scl)),
            return
        if self.a_type == "cost" or self.a_type == "Cost":
            return 1-(1/(max(self.scl)-min(self.scl)))*p+((min(self.scl)*-1)/
↪((max(self.scl)-min(self.scl)))), self.a_type
        else:
            return (1/(max(self.scl)-min(self.scl)))*p+((min(self.scl)*-1)/
↪((max(self.scl)-min(self.scl)))), self.a_type



    def Single_Portfolio_Single_Attribute(self, attribute =␣
↪"Space_requirement_for_Waste_Water_Treatment", Portfolio = 1, a_type =␣
↪"benefit",  plot = True):
        #Now we are using the value function created to calculate the portfolio␣
↪value distribution (PDF)
        #For this we have to generate the portfolio_distribution corresponding␣
↪first using the uncertainty class
        #If the scales are obtained locally this has already been done and the␣
↪flag is set to True.
        #if the scales are obtained globally the distribution still has to be␣
↪generated.
        self.a_type = self.dataset_2.criteria_weights()[1][self.info.columns.
↪tolist().index(attribute)]
        if attribute != self.attribute:
            self.attribute = attribute
            print("Uncertainty Distributions are first generated")
            Boxplots_values = Uncertainty_Portfolios()
            Boxplots_values.plot_all_portfolios(self.attribute, exclude =␣
↪False, plot = False)
            self.scl = [np.min(Boxplots_values.temp_df_2.values), np.
↪max(Boxplots_values.temp_df_2.values)]
            self.temp = Boxplots_values.temp_df_2
        if self.a_type == "cost" or self.a_type == "Cost":
```

```
            self.df = 1-((1/(max(self.scl)-min(self.scl))*self.
↪temp[str(Portfolio)].values+(min(self.scl)*-1)/((max(self.scl)-min(self.
↪scl)))))
        else:
            self.df = 1/(max(self.scl)-min(self.scl))*self.temp[str(Portfolio)].
↪values+(min(self.scl)*-1)/((max(self.scl)-min(self.scl)))
        if plot == True:
            ax = sns.boxplot(data = self.df, showfliers = False)
            ax.set_title("Impact on: " + str(attribute))
            ax.set_xlabel("Portfolio:"+str(Portfolio))
            ax.set_ylabel("Value")
        return self.df


    def collect_attributes_portfolios(self, attribute =␣
↪"Space_requirement_for_Waste_Water_Treatment", a_type = "benefit"):
        self.Portfolios_n = np.linspace(1,self.dataset.values.shape[0], self.
↪dataset.values.shape[0])
        self.a_type = self.dataset_2.criteria_weights()[1][self.info.columns.
↪tolist().index(attribute)]
        if attribute != self.attribute:
            self.attribute = attribute
            #print("Uncertainty Distributions are first generated")
            Boxplots_values = Uncertainty_Portfolios()
            Boxplots_values.plot_all_portfolios(self.attribute, exclude =␣
↪False, plot = False)
            self.scl = [np.min(Boxplots_values.temp_df_2.values), np.
↪max(Boxplots_values.temp_df_2.values)]
            self.temp = Boxplots_values.temp_df_2
        if self.a_type == "cost" or self.a_type == "Cost":
            self.df_all = 1-((1/(max(self.scl)-min(self.scl))*self.temp.
↪values+(min(self.scl)*-1)/((max(self.scl)-min(self.scl)))))
        else:
            self.df_all = 1/(max(self.scl)-min(self.scl))*self.temp.
↪values+(min(self.scl)*-1)/((max(self.scl)-min(self.scl)))
        return self.df_all

    #Method that generates the ranking for cardinal weights w.

    def all_attributes(self):
        self.summed_over_attributes = 0
        self.generated_attribute_stored = []
        weight = self.portfolio_data.dataset.criteria_weights()[0]
        for attribute in self.info.columns:
            w = weight[self.info.columns.tolist().index(attribute)]
            print(attribute + " with weight:" + str(w))
```

```
            #add weights
            self.generated_attribute = self.
↪collect_attributes_portfolios(attribute)
            self.generated_attribute_stored.append(self.generated_attribute)
            #This one immidiatly rewrites and regenerates the analysis. But xi␣
↪is not stored.
            self.summed_over_attributes += (self.generated_attribute*w)
        self.F = np.asarray(self.generated_attribute_stored)
        self.portfolio_values_df = pd.DataFrame(self.summed_over_attributes,␣
↪columns = list(range(1,self.dataset.values.shape[0]+1)))
        return self.portfolio_values_df

    #Method to create the entire value dataset.

    def xi(self):
        self.generated_attribute_stored = []
        for attribute in self.info.columns:
            #print(attribute)
            #add weights
            self.generated_attribute = self.
↪collect_attributes_portfolios(attribute)
            self.generated_attribute_stored.append(self.generated_attribute)
        self.F = np.asarray(self.generated_attribute_stored)
        print("Finished")
        return

    #method all_attributes_2 is quicker than the first version because it does␣
↪not recreate all the uncertainties unncessary.
    #Furterhmore is immidiatly converts to the means of the portfolios on the␣
↪attributes.
    def all_attributes_2(self, weight = np.array([0, 0, 0, 0, 0, 1, 0, 0, 0, 0,␣
↪0])):
        np.random.seed(9)
        self.summed_over_attributes_2 = 0
        #weight = self.portfolio_data.dataset.criteria_weights()[0]
        self.weight = weight
        for attribute in self.info.columns:
            w = self.weight[self.info.columns.tolist().index(attribute)]
            #print(attribute + " with weight:" + str(w))
            self.summed_over_attributes_2 += np.mean(self.F[self.info.columns.
↪tolist().index(attribute)], axis = 0)*w
            #self.portfolio_values_df_2 = pd.DataFrame([self.
↪summed_over_attributes_2], columns = list(range(1,self.dataset.values.
↪shape[0]+1)))
        return self.summed_over_attributes_2
```

```
    def rank_portfolios_average(self):
        #Mean is the expected value from a distribution of discrete numbers.
        self.sorted_series =  pd.DataFrame([self.summed_over_attributes_2],␣
↪columns = list(range(1,self.dataset.values.shape[0]+1))).mean().
↪sort_values(ascending = False).index
        return self.sorted_series.values

    def ranks(self, samples = 5):
        self.r_1 = []
        weight = self.sample_weight_space(samples)
        for i in weight:
            self.r_1.append(pd.DataFrame([self.all_attributes_2(weight = i)],␣
↪columns = list(range(1,self.dataset.values.shape[0]+1))).mean().
↪sort_values(ascending = False).index)
        self.r_1 = np.asarray(self.r_1)
        self.r_1 = np.transpose(self.r_1)
        return self.r_1[0]

    def sample_weight_space(self, samples, n=10):
        a = np.linspace(np.append(np.insert(arr = np.sort(np.random.
↪uniform(size = n)), obj=0, values = [0]), 1), np.append(np.insert(arr = np.
↪sort(np.random.uniform(size = n)), obj=0, values = [0]), 1), samples)*-1
        return np.sort(np.diff(a[::-1]))*-1

    def ranks_2(self, samples = 5):
        start = time.time()
        self.samples = samples
        np.random.seed(9)
        self.summed_over_attributes_2 =  0
        weight = self.sample_weight_space(samples)
        self.matrix = 0
        self.dsb = {}
        self.r_1 = []
        self.weight_rank = self.info.columns.values[np.argsort(self.dataset_2.
↪criteria_weights()[0])]

        for attribute in self.weight_rank:
            #self.summed_over_attributes_2 += np.mean(self.F[self.info.columns.
↪tolist().index(attribute)], axis = 0)
            #Takes the mean of all 137 portfolios on that attribute. This one␣
↪do want to multiply vectorised before continuing to the next attribute
            #Now it recalculates and resums everything unnecissarily.
            #matrix is the output of the portfolios on the rows, and the sample␣
↪of the weight vector per colums.
```

```
            #This matrix is for the attribute and thus still has to be summed␣
↪with the other attribute matrices.
            #This produces a matrix of values for each portfolio on the rows␣
↪and sampled weights.
            #From this the ranking can occur much faster than iterate over each␣
↪weight vector.


            #last part here ignores the weights : "self.info.columns.tolist().
↪index(attribute)"
            self.matrix += np.outer(np.mean(self.F[self.info.columns.tolist().
↪index(attribute)], axis = 0), weight.T[self.info.columns.values[np.
↪argsort(self.dataset_2.criteria_weights()[0])].tolist().index(attribute)])
      #p_vector is the sorted row.
        #compare index values to match a portfolio from the matrix index to␣
↪rank index.
        self.matrix_sorted = np.argsort(self.matrix, axis = 0)[::-1]+1
        #self.r_1 = pd.DataFrame(matrix_sorted.T, columns = list(range(1,self.
↪dataset.values.shape[0]+1)))
        #matrix.sort(axis = 0)
        unique, counts = np.unique(self.matrix_sorted[1], return_counts=True)
        self.dsa = dict(zip(unique, counts))

        for i in range(1, self.matrix_sorted.shape[0]+1):
            unique, counts = np.unique(self.matrix_sorted[i-1],␣
↪return_counts=True)
            self.dsa = dict(zip(unique, counts))
            self.dsb["rank " + str(i)] = self.dsa

        #self.test = unique, counts
      # print(self.dsa)
        end = time.time()
        elaped_time = end-start
        print("Elaped time was: ", elaped_time, " for: ", samples, 'samples')
        return

    def acceptability_index(self, rank = "rank 2"):
        self.dsc = {}
        for x in self.dsb[rank]:
            #print("acceptability for", rank, "and portfolio", x, "is ", self.
↪dsb[rank][x]/self.samples)
            self.dsc[x] = {self.dsb[rank][x]/self.samples}
        return
```

```
    def acceptability_indeces(self, amount_of_ranks_included = 10, port_amount␣
↪= 10):
        self.dsx = {}
        for j in list(self.dsb.keys()):
            self.acceptability_index(rank = j)
            self.dsx[j] = self.dsc
        list_best = []


        for i in range(1,amount_of_ranks_included+1):
            list_best.append(list(self.dsx['rank ' + str(i)].keys()))
        flat_list = list(it.chain(*list_best))
        self.best = pd.DataFrame(pd.unique(np.asarray(flat_list)), columns =␣
↪["Strategic_Portfolio"])[0:port_amount]
        return self.best

    def dict_df(self):
        self.acceptability = pd.DataFrame(columns = ["Strategic Portfolio",␣
↪"Acceptibility Index", "rank"])
        for i in list(self.dsx.keys()):
            regular_list = list(list(self.dsx[i][k]) for (k) in self.dsx[i].
↪keys())
            flat_index = [item for sublist in regular_list for item in sublist]
            df = pd.DataFrame(list(self.dsx[i].keys()), columns = ["Strategic␣
↪Portfolio"])
            df["Acceptibility Index"] = flat_index
            df["rank"] = i
            self.acceptability = pd.concat([self.acceptability, df])
        return self.acceptability



    #Attribute = on which the portfolios are ranked as best performing.
    #Amount = How many of top performing are returned, maximum would be the␣
↪amount of portfolios generated ranked from low to high.
    def heuristic(self, attribute = 'Not Specified', amount_of_ranks_included =␣
↪10):\
        #looks at how many attributes there are and uses this lenght to slice␣
↪the appropiate columns, the other columns are not attributes but name and␣
↪such.
        length = np.asarray(self.dataset_2.column_names()).shape[0]
        attribute_array = np.asarray(self.dataset_2.column_names())[2:length]
        #Find the corresponding index position for the attribute.
#          if attribute == "Not Specified":
#              attribute = 1
        self.index_value = np.where(attribute_array == attribute)
```

```
        #plus one is to count the first portfolio as portfolio 1 and not as␣
↪portfolio 0
        self.heuristic_outcome = pd.DataFrame(self.F.mean(axis = 1)[self.
↪index_value].argsort().T+1, columns = ["Portfolios"]).
↪iloc[-amount_of_ranks_included:][::-1]
        self.heuristic_outcome["Rank"] = list(range(1,amount_of_ranks_included␣
↪+1))
        self.heuristic_outcome.set_index("Rank", inplace = True)
        return self.heuristic_outcome


    def return_info_portfolio(self, portfolio = 1):
        return self.portfolio_data.single_portfolio(portfolio)

    def return_info_partial_values(self):
        self.partial_values = pd.DataFrame(self.F.mean(axis = 1).T, columns=␣
↪["Attribute 1", "Attribute 2", "Attribute 3", "Attribute 4", "Attribute 5",␣
↪"Attribute 6", "Attribute 7", "Attribute 8", "Attribute 9", "Attribute 10",␣
↪"Attribute 11"])
        return self.partial_values

    def core(self, actor = "Not Specified"):
        core_array = []
        for i in self.best.values.flatten():
            core_array.append(self.return_info_portfolio(i)["Alternative"].
↪to_list())
        core_array_flat = list(it.chain(*core_array))
        core_df = pd.DataFrame(core_array_flat, columns = ["Core Alternative"])
        self.core_index = core_df.value_counts().to_frame(name = "Alternative␣
↪Count")
        self.core_index["Actor"] = actor
        self.core_index["Core Index"] = self.core_index.values[:, 0]/self.best.
↪shape[0]
        self.core_index.to_csv("C:/Users/paulu/Documents/Epa/Thesis/Portfolios␣
↪" + str(actor) + ".csv")
        Core_actor = pd.read_csv("C:/Users/paulu/Documents/Epa/Thesis/
↪Portfolios " + str(actor) + ".csv")
        self.CI = Core_actor[["Core Alternative", "Core Index"]].
↪rename(columns={"Core Index": "Actor " + str(actor)})
        self.CI.to_csv("C:/Users/paulu/Documents/Epa/Thesis/CI " + str(actor) +␣
↪".csv", index = False)
        return self.core_index

    #When wanting to search for all the portfolio that contain specific␣
↪alternatives one can run this search_alt method.
```

```python
    #Additional ques can be set in the if statement if more alternatives have to be included.
    #The mehod returns a list with all the portfolio containing alt_1 and alt_2
    def search_alt(self, alt_1 = 5, alt_2 = 8):
        Portfolio_including = []
        for elements in range(1, np.asarray(dataset.list_combinations).shape[0]):
            if alt_1 in np.asarray(dataset.list_combinations)[elements] and alt_2 in np.asarray(dataset.list_combinations)[elements]:
                Portfolio_including.append(elements)
        return Portfolio_including

    def portfolio_value_dist(self, attribute = "Attribute_1", portfolio = 118):
        self.portfolio_dist = pd.DataFrame(self.F.T[portfolio-1].T[self.info.columns.tolist().index(attribute)], columns = ["Portfolio number: " + str(portfolio)])
        return self.portfolio_dist

    def portfolio_value_dist_all(self, portfolio = 118):
        self.portfolio_dist_merged = pd.DataFrame()
        for attribute in self.info.columns:
            self.attribute_temp = attribute
            self.portfolio_value_dist(attribute = attribute, portfolio = portfolio)
            self.portfolio_dist_merged[attribute] = self.portfolio_dist["Portfolio number: " + str(portfolio)].values

        return self.portfolio_dist_merged

    def boxplot_portfolio_values(self, attribute = "Attribute_1", portfolio = 118):
        sns.set(rc={"figure.figsize":(4, 5)})
        sns.set_style('whitegrid')
        self.portfolio_test = pd.DataFrame(self.F.T[portfolio-1].T[self.info.columns.tolist().index(attribute)], columns = ["Portfolio number: " + str(portfolio)])
        ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.portfolio_test), showfliers = False, palette = "Blues")
        ax.set_title("Impact on: " + str(attribute))
        ax.set_xlabel(None)
        return

    def boxplot_portfolio_values_all(self, portfolio = 118):
        sns.set(rc={"figure.figsize":(7.5, 6)})
        sns.set_style('whitegrid')
```

```
        self.portfolio_all = self.portfolio_value_dist_all(portfolio =␣
↪portfolio)
        ax = sns.boxplot(x = 'variable', y = 'value', data = pd.melt(self.
↪portfolio_all), showfliers = False, palette = "Blues")
        ax.set_title("Portfolio " + str(portfolio) + ": Impact on all␣
↪attributes")
        ax.set_xlabel(None)
        plt.setp(ax.get_xticklabels(), rotation=40, horizontalalignment='right')
        plt.show()
        return



V = Values()
V.xi()
V.ranks_2(samples = 100000)
V.acceptability_indeces()
V.dict_df()
#V.return_info_partial_values()
#V.boxplot_portfolio_values(attribute = "Attribute_5", portfolio = 118)
#V.info.columns
```

[ ]:
```
V.core("Y")
```

[ ]:
```
Core_Y = V.core_index
```

[ ]:
```
Core_Y["Alternative"] = Core_Y.index
Core_Merged
```

[ ]:
```
#To Produce the Barplot this code is used to generate the countings of the␣
↪barplots
#Be AWARE that Core_Y first needs to be generated unto which you do the first␣
↪merger. After this you can merge the Core_Merger with Core_X
#Replace the old Core_Merger with the new one.
def barplot_data(actor = "X"):
    #The V.core(actor) runs the Value Class method so have the attribute␣
↪ranking also right for the corresponding actor.
    V.core(actor)
    Core_temp = V.core_index
    Core_temp["Alternative"] = Core_temp.index
    Core_Merged = Core_Y.merge(Core_temp, how = "outer")

    #After all actors are incorporated into the Core_Merged the alternative␣
↪values are obtained from the weird tuple format:
    Alternative_List = []
    for i in Core_Merged["Alternative"].values:
        Alternative_List.append(i[0])
```

```
    Core_Merged["Alternative"] = Alternative_List
    return Core_Merged

#Plot the Barplot
Core_merged = Core_merged.replace(np.nan,0)
#Core_merged["Combined CI"] = Core_merged["Actor Y"]*Core_merged["Actor X"]
Core_merged.sort_index()
sns.set(rc={"figure.figsize":(7, 5)})
sns.set_style('whitegrid')
sns.barplot(data =Core_merged, x = "Alternative", y = "Alternative Count", hue␣
↪= "Actor", palette = "Blues")
plt.show()


#Plot the Heatmap
df_wide = Core_Merged.pivot_table(index='Alternative', columns='Actor',␣
↪values='Core Index')
rdgn = sns.diverging_palette(h_neg=10, h_pos=133, s=99, l=55, sep=3,␣
↪as_cmap=True)
sns.set(rc={"figure.figsize":(8, 5.5)})
sns.heatmap(df_wide , center=0.5, cmap=rdgn)
```

1.3 Interaction Values

The selection of best performing portfolios can be fine tuned by selecting from alreadybest ranking portfolios only portfolios with synergies. (So, highest Values with Synergy). In this second situation it is also assumed that the best performing portfolios have similar outcomes in which the existence of a synergy can decide in favour of very similar Portfolios.

First all the alternative combinations with positive synergy values have to be loaded in. Secondly, the "top" performing portfolios are screened wheter these interactions do occur. This happens after a general MAUT ranking which already includes attribute preferences. Then the rankings are compared for the 5 best performing portfolios and the core projects in these portfolios.

[ ]:
```
class Interaction():
    #dataset_1 is the original data from which the synergy combinations are␣
↪extracted loaded into a panda dataframe.
    #dataset_2 is the dataset of ranked portfolios in a panda dataframe.
    def __init__(self, dataset_1 = Complete_Matrix, dataset_2 = Portfolio_Data.
↪df_aggregated, top = V):
        self.interaction_sets = dataset_1.interactions()
        #here specify the top ranking portfolios to be included in the synergy␣
↪check.
        self.portfolios = dataset_2
        self.top_ranking = top
        self.top = np.unique(np.asarray(top.acceptability_indeces(10)))
        return
```

```
    def check(self):
        #from the input of dataset_2 the portfolios are checked on the␣
↪existence of synergies from the Excel File.
        #These are than isolated from the other portfolios and returned in a␣
↪dataframe.
        self.portfolios_with_synergy = pd.DataFrame()
        for i in range(1, len(Complete_Matrix.interactions())+1):
            set_1 = Complete_Matrix.interactions().values[i-1][0],␣
↪Complete_Matrix.interactions().values[i-1][1]
            for j in self.top:
                if set_1[0] in Portfolio_Data.list_combinations[j] and set_1[1]␣
↪in Portfolio_Data.list_combinations[j]:
                    #print(set_1[0], "and", set_1[1],"are in portfolio",␣
↪str(j), ": ", Portfolio_Data.list_combinations[j])
                    self.portfolios_with_synergy = self.portfolios_with_synergy.
↪append(self.portfolios.iloc[[j-1]])
        return np.unique(self.portfolios_with_synergy["Strategic_Portfolio"].
↪values)
    #to obtain only the strategic portfolio number add␣
↪[["Strategic_Portfolio"]].values to the above returned method.



    def core(self, amount_of_ranks_included = 6, actor = "Not Specified"):
        self.top = np.unique(np.asarray(self.top_ranking.
↪acceptability_indeces(amount_of_ranks_included = amount_of_ranks_included)))
        core_array = []
        for i in self.check().flatten():
            core_array.append(self.top_ranking.
↪return_info_portfolio(i)["Alternative"].to_list())
        core_array_flat = list(it.chain(*core_array))
        core_df = pd.DataFrame(core_array_flat, columns = ["Core Alternative"])
        self.core_index = core_df.value_counts().to_frame(name = "Alternative␣
↪Count")
        #self.core_index["Actor"] = actor
        self.core_index.to_csv("C:/Users/paulu/Documents/Epa/Thesis/Portfolios␣
↪" + str(actor) + ".csv")
        return self.core_index

Core_test = Interaction()
Core = Core_test.core(2, actor = "Y")

#Portfolios_with_Synergy = Interaction()
#a = Portfolios_with_Synergy.check()
```

```
#a = pd.DataFrame(a, columns = ["Best Performing Portfolios for Actor Y"])

#Portfolios_with_Synergy.portfolios
```

1.4 Normalisation Techniques

Mainly used for exploratory purposes

[ ]:
```
class Normalisation():
    def __init__(self,  dataset = Portfolio_Data):
        self.inhereted_class = dataset
        self.dataset = dataset.df_aggregated
        self.x = self.dataset.values
        #Indexes of the criteria that are deemed 'cost criteria'
        self.cost = [6, 7, 8, 10]
        self.z = "Dataset not yet normalised"
        self.z_df = "Dataframe not yet constructed"
        self.raise_norm = False
        self.raise_mul = False
        self.zw = "Normalised Data not yet weighted, run weights_added() first"
        self.zw_df = "Dataframe not yet constructed"


    def v_norm(self):
        k = np.array(np.cumsum(self.x**2, axis=0))
        #next step is to take the root of this numer k to obtain the magnitude␣
↪of each column.
        #Each element is divided by this magnitude to have it normalised.
        #For row i and column j the elements are scaled by the corresponding␣
↪magnitude ratio for that column.
        #For example the first criteria, vector has a certain magnitude.
        z = np.array([[round(self.x[i, j] / np.sqrt(k[self.x.shape[0] - 1, j]),␣
↪3) for j in range(self.x.shape[1])]
        for i in range(self.x.shape[0])])
        #print("The vector normalised matrix yields:\n")
        self.z = z
        self.raise_norm = True
        return self.z

    def v_norm_df(self):
        if self.raise_norm == False:
            self.v_norm()
        self.z_df = pd.DataFrame(self.z, columns = list(self.dataset.columns))
        self.z_df["Strategic_Portfolio"] = np.array(range(self.z.shape[0])) + 1
        return self.z_df

    def norm_cost(self):
```

```
        if self.raise_norm == False:
            print("First run norm()")
            return
        print('For criteria in indexes of self.cost, namely: ', self.cost, "␣
↪the values are transformed to cost criteria, altering the normalisation")
        for i in self.cost:
                self.z[:, i:(i+1)] = 1 - self.z[:, i:(i+1)]
        return self.z

        #For a criteria in the criteria column list, if it is one to be␣
↪decreased, then the outcomes on that column have to substracted from one and␣
↪replaced
        #Or do this in the normalisation step to, with extra if statement,␣
↪search via dataset.aggregate()
        #T = 1 - self.z[:,0:1]
        #self.z[:,0:1] = T
        #return self.z


    def weights_added(self):
        self.weights = self.inhereted_class.dataset.criteria_weights()[0]
        if self.weights.shape[0] != self.z.shape[1]:
            print("The amount of criteria do not match the amount of weights in␣
↪the excel files; they have to be equal.")
            return
        #weight= [range(self.z.shape[1])]
        #self.w = np.ones(self.z.shape[1])
        self.zw = np.array([[self.z[i, j] * self.weights[j]
            for j in range(self.z.shape[1])]
            for i in range(self.z.shape[0])])
        #print("The weighted vector normalised matrix yields:\n {}.".
↪format(self.zw))
        self.raise_mul = True
        return self.zw

    def weights_added_df(self):
        if self.raise_mul == False:
            self.weights_added()
        self.zw_df = pd.DataFrame(self.zw, columns = list(self.dataset.columns))
        self.zw_df["Strategic_Portfolio"] = np.array(range(self.zw.shape[0])) +␣
↪1
        return self.zw_df

    def run(self):
        self.v_norm()
        self.norm_cost()
```

```
        self.v_norm_df()
        self.weights_added()
        self.weights_added_df()
        print("The Portfolio Matrix has been normalised.")
        return

Portfolio_Normalised = Normalisation()
Portfolio_Normalised.run()
```

1.5 Topsis Implementation

Used in exploratory first understanding of MCDA ranking techniques. Is deemed less adaquate than MAVT.

[ ]:
```
class Topsis():
    def __init__(self,  dataset = Portfolio_Normalised):
        self.inhereted_class = dataset.inhereted_class
        self.dataset = dataset.z
        self.zw = dataset.zw
        self.raise_zenith = False
        self.normalisation = "Not Specified"
        self.nadir_method = "Not Specified"

    def zenith_nadir(self):
        """ zenith and nadir virtual action function; self.u is the
        weighted normalized decision matrix and method is the
        action used. For min/max input 'm' and for absolute
        input enter 'a'
        """
        u = self.zw
        if self.nadir_method == 'm':
                bb = []
                cc = []
                for i in range(u.shape[1]):
                    bb.append(np.amax(u[:, i:i + 1]))
                    b = np.array(bb)
                    cc.append(np.amin(u[:, i:i + 1]))
                    c = np.array(cc)
                #print("The zenith is {} and the nadir is {}.".format(b, c))
                self.zenith = b
                self.nadir = c
                self.raise_zenith = True
                return (self.zenith, self.nadir)
        else:
                #creates a vector of ones and zeros of length of matrix X
                b = np.ones(u.shape[1])
                print(b)
```

```
            c = np.zeros(u.shape[1])
            #print("The zenith is {} and the nadir is {}.".format(b, c))
            self.zenith = b
            self.nadir = c
            self.raise_zenith = True
            return (self.zenith, self.nadir)


    def distance(self):
        u = self.zw
        """ calculate the distances to the ideal solution (di+)
        and the anti-ideal solution (di-); u is the result
        of mul_w() and b, c the results of zenith_nadir()
        """
        distance = []
        ideal_i = []
        non_ideal_i = []
        for i in range(u.shape[0]):
            #Alternatives
            #The j is the amount of criteria.
            g = 0
            o = 0
            for j in range(u.shape[1]):
                #criteria
                a = u[i, j] - self.zenith[j]
                g += a**2
                b = u[i, j] - self.nadir[j]
                o += b**2
            g = math.sqrt(g)
            o = math.sqrt(o)
            ideal_i.append(g)
            non_ideal_i.append(o)
        distance.append(g)
        distance.append(o)
        return np.asarray(ideal_i), np.asarray(non_ideal_i)

        return (np.sqrt(sum(a, 1)), np.sqrt(sum(b, 1)))

        #a = np.array([[(u[i, j] - self.b[j])**2
        #    for j in range(self.zw.shape[1])]
        #    for i in range(self.zw.shape[0])])
       # print(a)
        #b = np.array([[(u[i, j] - self.c[j])**2
       #     for j in range(self.zw.shape[1])]
       #     for i in range(self.zw.shape[0])])
       # return (np.sqrt(sum(a, 1)), np.sqrt(sum(b, 1)))
```

```
    def single_portfolio(self, portfolio_set = 1):
        self.portfolio_set = portfolio_set
        return self.inhereted_class.single_portfolio(portfolio_set)

    def aggregated_single_portfolio(self, portfolio_set = 1):
        return self.dataset.aggregated_single_portfolio(portfolio_set)




    def topsis(self, pl = 'no', normalisation = "v", nadir_method = "a"):
        self.normalisation = normalisation
        self.nadir_method = nadir_method
        if self.raise_zenith == False:
            #print("First calculate zenith and nadir by running zenith_nadir()")
            self.zenith_nadir()
        """ matrix is the initial decision matrix, weight is
        the weights matrix, norm_m is the normalization
        method, method is the action used by zenith_nadir(), and pl is 'yes'
        for plotting the results or any other string for
        not.

        """
        #z = self.zw
        s, f = self.zenith, self.nadir

        p, n = self.distance()
        final_s = np.array([n[i] / (p[i] + n[i]) for i in range(p.shape[0])])
        #Here merge the final_s outcomes with the original dataframe of the␣
↪portfolios as to be able to select the best x performing portfolios.
        #Now only the max is selected.
        #C_F_list = pd.DataFrame(final_s, columns = ["Closeness coefficient"])
        #self.list_total = self.dataset.join(C_F_list).sort_values("Closeness␣
↪coefficient", ascending = False)


        if pl == 'yes':
            q = [i + 1 for i in range(self.zw.shape[0])]
            plt.plot(q, p, 'p--', color = 'red',
                markeredgewidth = 1, markersize = 3)
            plt.plot(q, n, '*--',  color = 'blue',
                markeredgewidth = 1, markersize = 3)
            plt.plot(q, final_s, 'o--', color = 'green',
                markeredgewidth = 1, markersize = 3)
            plt.title('TOPSIS results')
            plt.legend(['Distance from the ideal',
                'Distance from the anti-ideal',
```

```
                'Closeness coefficient'])
            #plt.xticks(range(self.zw.shape[0]+5))
            if self.nadir_method == "m":
                plt.axis([0, self.zw.shape[0] + 1, 0, 1.2])
            else:
                plt.axis([0, self.zw.shape[0] + 1, 0, 5])
            plt.xlabel('Portfolios')
            plt.grid(True)
            plt.show


            #Determine to which attribute the highest closseness coefficient␣
↪corresponds:
        place = np.where(final_s == final_s.max())[0]+1
            #print("The place of the max attribute is ", place)
        print("The maximum value of the closeness coefficients is: {}, which␣
↪corresponds to portfolio {} from the decision matrix.".format(final_s.max(),␣
↪place))

        self.raise_mul = False
        self.raise_norm = False
        self.raise_zenith = False
        self.normalisation = "Not Specified"
        self.nadir_method = "Not Specified"
        #print("The closeness coefficients are: {}, with maximum value {},␣
↪which corresponds to portfolio {} from the decision matrix".format(final_s.
↪self, final_s.self.max(), place))
        return self.single_portfolio(int(place))


Topsis_Method = Topsis()
```