Jose Malmberg

Data-Driven Interactive Multiobjective Optimization using Cluster Based Surrogate in Discrete Decision Space

Master's Thesis in Information Technology

June 13, 2018

University of Jyväskylä

Department of Mathematical Information Technology

Author: Jose Malmberg

Contact information: jose.j.malmberg@student.jyu.fi

Supervisors: Jussi Hakanen, Vesa Ojalehto

Title: Data-Driven Interactive Multiobjective Optimization using Cluster Based Surrogate in Discrete Decision Space

Työn nimi: Datapohjainen interaktiivinen monitavoitteinen päätöksenteko diskreetin muuttuja-avaruuden klusteripohjaista sijaismallia käyttäen

Project: Master's Thesis

Study line: Computational Sciences

Page count: 99+0

Abstract: This thesis presents a cluster based surrogate model approach for reducing dimension of discrete decision space and so for simplifying integer linear optimization problems. The model is especially aimed for solving data-driven decision making problems interactively, as the surrogate makes interaction more seamless and the interactive NIMBUS method manages well within the product space of the surrogate. The developed cluster based surrogate and method were also applied for a Boreal Forest management problem with promising results.

Keywords: surrogate, metamodel, clustering, cluster based, synchronous NIMBUS, multiobjective optimization, multiobjective decision making, interactive optimization

Suomenkielinen tiivistelmä: Tutkielma esittää klusteripohjaisen sijaismallin diskreetin päätöksentekoavaruuden dimension pienentämiseksi ja lineaaristen kokonaislukuoptimointitehtävien yksinkertaistamiseksi. Sijaismalli on suunnattu erityisesti datapohjaisten päätöksenteko-ongelmien interaktiiviseen ratkaisemiseen, sillä se yhdistää sijaismallin interaktiota helpottavan vaikutuksen ja interaktiivisen NIMBUS menetelmän hyvän suorituskyvyn sijaismallin tuloavaruudessa. Kehitettyä sijaismallia ja metodia myös sovellettiin monitavoitteiseen metsätalousongelmaan hyvin tuloksin.

Avainsanat: sijaismalli, metamalli, surrogaatti, klusterointi, klusteripohjainen, synkroninen NIMBUS, monitavoiteoptimointi, monitavoitteinen päätöksenteko, interaktiivinen päätöksenteko, optimointi

List of Figures

List of Tables

Contents

1 Introduction

Industry, business and everyday life require decision-making. Depending on the situation and the problem at hand, these decisions may be simple and go unnoticed, or they may be very complex and even impossible to solve unambiguously. With simple and unnoticeable problems we usually settle for "good enough" solutions, but more complex the problem the better solutions we usually want. Sometimes we want the decisions to be as good as possible i.e. *optimal*.

The process of systematically exploring alternative solutions and making the optimal decisions is called *optimization*. Depending on if there are just one or multiple conflicting and mathematically equally important objectives, these problems are called *single* or *multiobjective optimization*.

In the single objective optimization there exists only one optimal value. The same optimal value may be attained by multiple different ways, but essentially there is just one unambiguous optimum. In single objective optimization problems the solving then simply means finding that optimum and how it can be achieved.

In the case of multiple objectives, unambiguous optimum for all the objectives cannot be found: There can be just a set of solutions, which all are such that one objective cannot be improved without impairing the value of at least one of the other objective functions. These are called *Pareto optimal solutions*. It is not possible to select the most preferred Pareto optimal solution without some additional information about the multiobjective problem. From the practical perspective, selecting only one of those solutions is still usually desired. In the multiobjective optimization, solving then means finding and somehow selecting the most preferred solution. This process may or may not require finding all the Pareto optimal solutions.

The simplest and probably the most used method to solve multiobjective optimization problems is the weighting method. In this method every objective is given a weighting coefficient and the sum of these weighted objectives is optimized. However, Miettinen

(1999) notes that these weights produce unpredictable results and the weighted sum does not actually have any understandable meaning.

Fortunately a plethora of different and more sophisticated means for optimizing multiple objectives have been developed. These methods differ significantly from one to another, but instead of just assigning seemingly helpful weights for different objectives, these methods involve a *Decision Maker*: a domain expert and person capable of making the final choice between different objectives and solution alternatives.

Usually the nature of multiobjective problem is such that the relationships between different objectives are not clearly observable; not even for an expert of the problem domain. In this kind of situations the real features of the problem and different solution alternatives can be discovered using *interactive decision making methods*. As the name implies, these methods require active involvement of the decision maker during the solving process. This can be a taxing task in some cases, but it also eases the process of finding the most preferred solution.

In addition to just making a decision, the decision is wanted to be justified. This justification has to be based on some arguments and assumptions and in order to find these, relevant information is needed. Vanian (2016) describes how information as such is not always a natural resource but it has to be refined and interpreted from the data that can be collected and stored. Therefore the data is also called “the new oil”, and from it new information can be extracted.

During the last decades, the amount of data collected in certain fields has already been big, but limited resources have been a restriction for making most of it. Along with the development of technology, information handling and computational methods, the resources available have increased and the former restriction decreased or even ceased to exist. This has then created new opportunities for data-driven decision making and optimization, as stated by Press (2013) and Anderson (2008), among others.

Although former restriction are gone, new ones have emerged. Maney (2014) has brought out that even though the processing power has increased significantly during

the last decades, the amount of data available and ready to be used has increased even more. Because of this, the size of the problems that we are able and willing to solve are mathematically of much larger scale than before. This has led to situations where the problems as such are not very complex (e.g. only linear mixed integer problems) but memory demands and processing power requirements to solve them are very high. This can then either make the problems completely unsolvable or make solving them excessively slow.

In this thesis, this challenge of solving mathematically large scale integer linear problems is tackled by developing and using a mathematically cheaper *surrogate model*. The developed surrogate model is based on clustering the variables in the decision space and using only a representative subset of variables to approximate the entire decision space.

In order to solve the surrogate based problem an interactive multiobjective optimization method, namely the synchronous NIMBUS approach by Miettinen and Mäkelä (2006) is used. This kind of approach is chosen because interactive methods have come in handy for solving complex multiobjective problems and are more likely to reveal possible modeling errors in the surrogate models than non-interactive multiobjective optimization methods, as Sindhya et al. (2014) state. Particularly the synchronous NIMBUS method was chosen because the cluster based surrogate transforms the result space of the multiobjective problem so that finding desired solutions may become challenging. The NIMBUS methods is able to overcome this problem by utilizing four different scalarizing functions, which increases greatly the probability of finding at least one satisfactory solution from the result space.

This thesis concerns solving data-driven multiobjective optimization problems using a cluster based surrogate and interactive methods. Especially the focus on this study is on the situations when the data to be used in the optimization is so large, that it is not possible to do meaningful interactive decision making without using any surrogate. This research problem and its sub-problems are formally defined:

> " *How to solve large data-driven multiobjective optimization problem interactively using surrogate?"*
>
> 1. *How to form a cluster based surrogate from the data?*
> 2. *How to formulate a multiobjective optimization problem using cluster based surrogates?*
> 3. *How to solve a multiobjective optimization problem using cluster based surrogate and interactive methods?*

The rest of this thesis is organized as follows: Multiobjective optimization, its basic concepts and interactive methods are presented in Chapter 2. Chapter 3 explains data-driven optimization including surrogate models. Chapter 4 documents the cluster based multiobjective optimization approach and is the main contribution of this thesis. Chapter 5 is a case study of the developed method presenting and solving a Boreal Forest planning problem using interactive methods and a cluster based surrogate. Discussion about the developed method can be found in Chapter 6. The final conclusions including future research topics are presented in Chapter 7.

2 Multiobjective Optimization

In this chapter in Section 2.1 the general formulation of multiobjective optimization problem, its basic concepts and some approaches how to solve this kind of problems are introduced. Section 2.2 then focuses on solving multiobjective problems interactively. Finally Section 2.3 defines scalarizing functions, a common approach for solving multiobjective optimization problems.

2.1 Basic concepts

Miettinen (1999) defines the general formulation of multiobjective optimization problems (MOPs) as:

$$\begin{aligned} &\text{minimize } \{f_1(x), ..., f_k(x)\} \\ &\text{subject to} \qquad x \in S \end{aligned} \tag{2.1}$$

with k(≥ 2) *objective functions* $f_i(x) : S \to R$. The set S is the set of *feasible decisions* in the *decision space* R^n. This means all the decision variable vectors $\mathbf{x} = (x_1, ..., x_n)^T$ that are within all the constraints of the given problem. The corresponding image of S in *objective space* R^k is called *feasible objective* set and is denoted by Z. The elements of Z are *feasible objective vectors*, $z = (z_1, ..., z_k)^T$, where $z_i = f_i(x)$ for all $i = 1, ..., k$. In Figure 1 blue points illustrate these feasible solutions. In (2.1), and later on this thesis, all the objective functions are assumed to be minimized, but if some objective f_i should be maximized instead, it would be equivalent with minimizing $-f_i$.

A feasible solution $x' \in S$ and the corresponding $f_i(x') \in Z$ are said to be *weakly Pareto optimal* if there does not exist another feasible solution $x \in S$ such that $f_i(x) < f_i(x')$ for all $i = 1, ..., k$. In addition they are said to be *Pareto optimal* if there does not exist another feasible solution $x \in S$ such that $f_i(x) \leq f_i(x')$ for all $i = 1, ..., k$ and $f_j(x) < f_j(x')$ for at least some $j \in \{1, ..., k\}$. From this follows that all the Pareto optimal solutions are also weakly Pareto optimal, but not vice versa. In Figure 1 this is seen in points

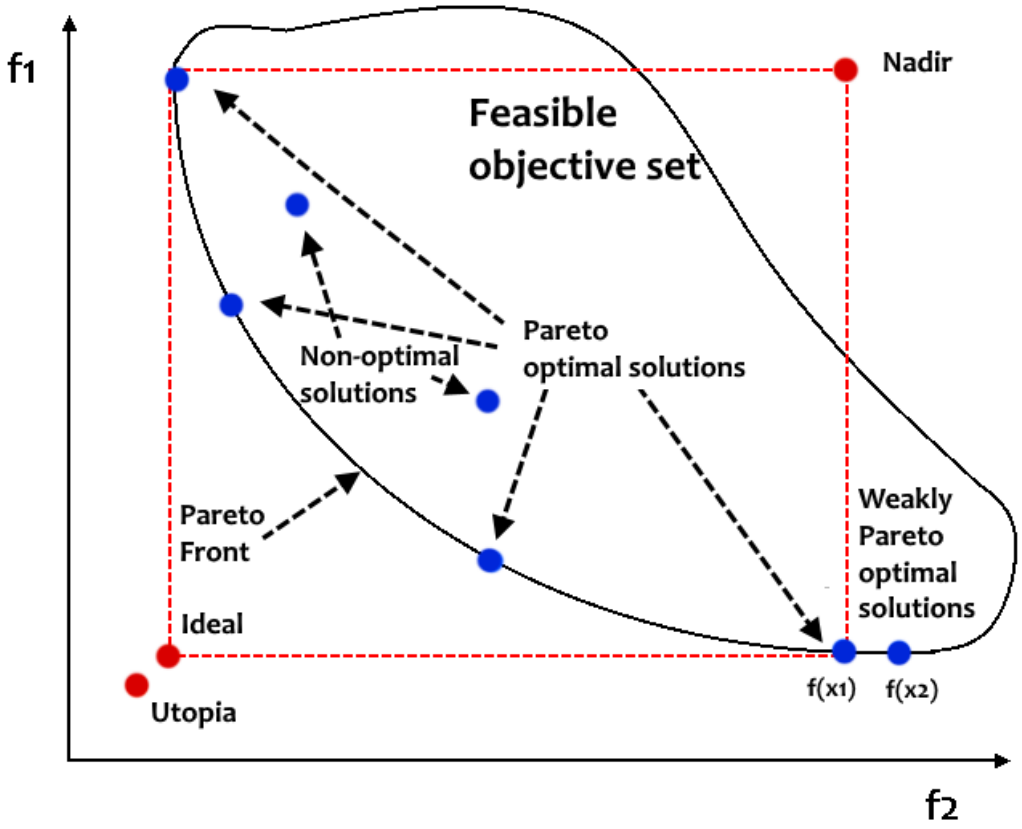

Figure 1: Ideal, nadir and utopia points of two-objective optimization problem, where both objectives are to be minimized.

x_1 and x_2, which both are weakly Pareto optimal, but only x_1 is also Pareto optimal ($f_1(x_1) < f_1(x_2)$, but $f_2(x_1) = f_2(x_2)$.) The set of all strictly Pareto optimal solutions in the objective space is called *Pareto front* or *Pareto optimal set*.

In order to solve multiobjective optimization problem, it is desired that the ranges for all the objective functions in the objective space are known. The lower bound $z_i^\star \in Z$ for individual objective function $f_i(x)$ is attained by minimizing the function f_i individually. When all these lower bounds are combined together to one vector $z^\star = \{z_1^\star, ..., z_k^\star\}$ the vector is called *ideal objective vector*. A vector $z^{\star\star} \in Z$ that is strictly better than the ideal vector $z^\star \in Z$ is called *utopian vector*. In practice the utopian vector is calculated by setting $z_i^{\star\star} = z_i^\star - \varepsilon$ for all $i = 1, ..., k$ and for some positive small scalar ε.

The collection of the upper bounds of the objective functions in the objective space is called *nadir objective vector*, z^{nad}. Miettinen, Ruiz, and Wierzbicki (2008) admit that in practice there does not exist any comprehensive way to calculate exact nadir vectors for nonlinear problems. Estimations for the vector can be made using a so-called payoff table, but the accuracy of the resulting vector is not guaranteed. Ideal, nadir and utopian vectors are also illustrated in Figure 1.

In multiobjective optimization, the problem is considered mathematically solved when all the Pareto optimal solutions have been found. Miettinen (1999) also calls that *vector optimization*. However, usually there are multiple (possibly infinitely many) Pareto optimal solutions but from the practical perspective normally only one solution can be chosen as the preferred one. This implies that some aspects beyond pure mathematics have to be considered. In many cases these aspects mean domain knowledge and human expertise. The person having these characteristics and being in the position of being capable of making the final decision is called *Decision Maker* (DM). Especially in the real life multiobjective optimization problems the role of DM is essential.

As originally classified by Hwang and Masud (1979), there are four different classes of optimization according to the role of the decision maker in the solution process:

1. DM giving no articulation of her/his preferences (no-preference methods)
2. DM giving a posteriori articulation of her/his preferences (a posteriori methods)
3. DM giving a priori articulation of her/his preferences (a priori methods)
4. DM giving progressive articulation of her/his preferences (interactive methods)

The first class – no-preference methods – means that decision maker does not give any information about her/his preferences and the best solution is chosen without her/his involvement. This kind of optimization approach is useful in situations when there is no decision maker available, for example in real time optimal control of some industrial process. When this approach is used with a decision maker, her/his role is to simply accept or reject the result. If DM is available, other optimization approaches are preferred, however.

In the a posteriori methods, the decision maker gives no preference information before the optimization. In the optimization phase, a representative set of Pareto optimal solutions is calculated and presented to DM, who then chooses the most preferred solution. This way DM is able to see all the options and compromises between objectives and so is able to make a justified choice. However if the number of possible solutions or objectives is large, it becomes challenging for DM to genuinely perceive and comprehend all the possibilities and trade-offs between objectives.

On the contrary to the a posteriori method, in the a priori method the decision maker gives her/his preferences before the optimization phase. In the optimization phase it is then desired to generate Pareto optimal solutions corresponding to the given preferences as well as possible. This way it is not needed to calculate all the Pareto optimal solutions, but only the ones corresponding the preferences. This way the final decision may also became easier to make. However, if DM has unrealistic expectations about the results and it may be ambiguous which Pareto optimal solutions would really be corresponding to her/his preferences.

In interactive methods the most preferred Pareto optimal solution is reached through interaction with the decision maker. In such a method the solution is found iteratively: After seeing the attainable bounds for all the objectives DM gives some initial preferences for the solution. Then after seeing the corresponding results DM indicates how (s)he would like the solutions be changed i.e. gives new preferences. These steps are repeated until DM is satisfied with the produced solution. The interactive methods are described in Section 2.2 as they are used later on in this thesis.

One common way of communicating the preferences of the decision maker is to use reference points. This means that the decision maker specifies some desirable values for all the objectives and then a Pareto optimal solution corresponding to those values is generated or discovered. More of the reference points is described in Section 2.2.1.

2.2 Interactive Methods

According to Miettinen and Mäkelä (2006) the goal of interactive multiobjective optimization methods is to find a single preferred solution from the set of all the Pareto optimal solutions. Therefore the interactive methods are not aimed for solving the problem mathematically i.e. finding all the Pareto optimal solutions of the problem. As the name implies, the process of finding the single optimal solution requires the decision maker being actively involved in the solution finding process. This kind of decision making has proved to be cognitively and computationally effective as Miettinen and Mäkelä (2006) state.

In addition to finding a desired solution, interactive multiobjective optimization is a process during which the different stakeholders of the decision making learn more about the problem itself. Even though DM would be an expert on the problem domain, it may not be trivial for her/him to understand how the variables and objectives are actually related to one another. Particularly this may be the case if the number of variables or objectives is large. Wierzbicki (1997) has also observed that decision making of experts is in many respects based on intuition, which also emphasizes the importance of learning during the interactive optimization process. Even though the intuition is already built along with gaining the expertise, it may still be supported by additional learning during the process.

Though the mathematical formulations of the optimization problems are tested by simulations etc. before the actual optimization, it is still possible that some weaknesses or flaws of the problem formulations remain unnoticed. Sindhya et al. (2014) state that in those kind of cases it is possible that the decision maker during the iterative process notices that the problem is not behaving according to her/his expectations and previous experiences. When further studied, errors in the mathematical model may be revealed. This will then lead to correcting the mathematical formulation of the problem and so ends up as a learning process for the analysts formulating the problem also.

As separated by Miettinen, Ruiz, and Wierzbicki (2008), the interactive multiobjective optimization process usually includes two phases: the learning phase and the decision making phase. In the learning phase the decision maker gets used to the decision making process and learns more about the structure of the problem at hand. After the learning, the final decision process takes place in the decision making phase. Depending on the situation, these two phases may also be intertwined.

According to Miettinen, Hakanen, and Podkopaev (2016) the interactive multiobjective optimization methods contain six general steps:

1. Initialize: The ideal and nadir values are calculated and presented to DM.
2. Generate a Pareto optimal solution to be a starting point: either a neutral compromise solution or a solution provided by DM.
3. Ask preference information from DM: a reference point or how many new solutions (s)he wants to be calculated, for example
4. Generate one or more Pareto optimal solutions according to the preferences of DM.
5. If several solutions were generated ask DM to choose the best one.
6. If DM is satisfied with the solution or does not want to continue, then stop. Otherwise, go to step 3

For stopping the interactive method there are three main reasons stated by Miettinen, Hakanen, and Podkopaev (2016): DM finds a desirable solution and wants to stop, DM gets tired and stops or some algorithmic stopping criterion is fulfilled. As one can see, the interactive optimization process and its termination are not about the mathematical convergence, but depend on the decision maker. Thus Miettinen, Ruiz, and Wierzbicki (2008) define that the interactive methods are based on the psychological convergence and not on the mathematical one.

In Miettinen, Hakanen, and Podkopaev (2016) there are listed five different ways to relay preference information between DM and the optimization frameworks: reference point approaches, classification based methods, comparison based methods, trade-off based methods and navigation methods. From these five, reference point approaches

and classification based methods are presented in Section 2.2.1 and Section 2.2.2, for they will be used as part of synchronous NIMBUS method later used in this thesis.

In addition to these, there exists also many more ways to express preference information, but they are out of the scope of this study. For more information about these, one may turn to Hwang and Masud (1979), Miettinen, Ruiz, and Wierzbicki (2008), Miettinen (1999), and Miettinen, Hakanen, and Podkopaev (2016).

The following sections 2.2.1 *Reference Point Methods* and 2.2.2 *Classification Based Methods* are based on definitions and observations by Miettinen, Ruiz, and Wierzbicki (2008) if not otherwise denoted.

2.2.1 Reference Point Methods

In the reference point approaches the preferences of the decision maker are communicated using reference points. These reference points mean that DM defines some specific values for all the different objectives in the form of a vector $\bar{z} \in R^k$. Then the optimization system aims to generate Pareto optimal solution(s) corresponding to these preferences as well as possible. After seeing the corresponding Pareto optimal solutions DM is free to give any new reference point according to her/his liking. This way DM is able to learn more about the objective space and to explore the Pareto optimal solutions. As in all interactive methods, DM is free to stop or to continue exploring new solutions as (s)he likes.

The fundamental assumption behind reference point methods is that the decision maker has some internal preference structure in her/his mind, that cannot and should not be defined explicitly by mathematical means. This means admitting that the mathematical formulation of the problem is always in a way limited and therefore DM should be encouraged to use her/his expertise freely. Wierzbicki (1997) also defines that the fundamental goal of reference point approaches is to empower the intuition of DM: reference point methods are just a tool for that and their purpose in the decision making is to prefer intuition to the rational mathematical functions.

So, instead of being fixed aspiration levels, the reference points given are considered as a tool for learning. Miettinen, Hakanen, and Podkopaev (2016) elaborate how this approach also enables DM to change her/his mind during the process and it does not force her/him to understand all the dynamics of the multiobjective problem in order to find meaningful solutions. Naturally, in the final decision making phase the reference points must also be used for choosing the single preferred solution. The concept behind the reference point approach is still more about the learning and exploring than fixing the aspirations. This way the learning that occurs when using the reference point approach aids the intuition the best.

In addition to these, one aim of the reference point approaches is to encourage the preferences of the decision maker be non-linear, which are natural for human decision makers. Using the reference points also gives to the decision maker a more holistic way to express her/his preferences, when all the objectives are presented equally important. This way DM can assess the situation more objectively and really utilize her/his expertise more broadly.

Because in the reference point methods the decision maker is giving the values for all the objectives completely according to her/his liking, a Pareto optimal solution corresponding to these preferences has to be found. That is an application specific task but usually some achievement scalarizing functions are used. These are to be introduced in Section 2.3.

2.2.2 Classification Based Methods

In the classification based methods the decision maker explores only the set of Pareto optimal solutions. Starting from some Pareto optimal solution DM classifies which objectives (s)he wants to improve and which ones are allowed to impair. Then according to these preferences another Pareto optimal solution is presented to her/him and the process is repeated.

This way by implying how (s)he would like to change the current Pareto optimal solution, DM is able to move about and explore the Pareto optimal set. It has been

shown by Larichev (1992) (as presented in Miettinen, Ruiz, and Wierzbicki (2008)) that the classification is a cognitively valid way for a human decision maker to express preference information. By moving only in the set of Pareto optimal solutions, all the solutions that are presented to the decision maker are meaningful to her/him, and so the interaction is more intuitive.

One implementation of the classification based approaches is the NIMBUS method, as described by Miettinen and Mäkelä (1995, 2000, 2006). There are five different index classes for the objective functions in this classification:

- $I^{<}$, whose values should be improved from the current level as much as possible
- $I^{\leq}$, whose values should be improved till some aspiration level $\hat{z}_i$
- $I^{=}$, whose values should stay the same as in the current level
- $I^{\geq}$, whose values can impair from the current level till some bound
- $I^{\diamond}$, whose values can change freely

According to the definition of Pareto optimality, this moving always requires trading off the current level of some objective for improvement of some other objective. From this follows that in the classification if some objective is classified to be improved, at least one other has to be classified to impair i.e. $I^{<} \cup I^{\leq} \neq \emptyset$ and $I^{\geq} \cup I^{\diamond} \neq \emptyset$. Also all the objective functions of the problem have to be classified.

The classification based methods are closely related to the reference point approaches. For example, the classification of NIMBUS method can be transformed into a reference point by setting: $\overline{z_i} = z_i^{\star}$ for $i \in I^{<}$, $\overline{z_i} = \hat{z}_i$ for $i \in I^{\leq}$, $\overline{z_i} = z_i^h$ for $i \in I^{=}$, $\overline{z_i} = \varepsilon_i$ for $i \in I^{\geq}$ and $\overline{z_i} = z_i^{nad}$ for $i \in I^{\diamond}$. Similarly, a reference point can be translated into a classification if the current Pareto optimal solution is known.

The main difference between using a reference point or a classification based approach is that in the former approach the reference point can be any point in the objective space but any classification has to be feasible. In some classification based methods, like in NIMBUS, the result of the optimization also has to be obeying the classification constraints set by the different classes (objective functions set to be improved really have to improve etc.). From this follows that sometimes it is not possible to generate

any solution according to classification preferences of DM, whereas in reference point based methods some solution is always generated.

2.3 Scalarizing Functions

According to Miettinen (1999) the scalarizing functions are important in the multiobjective optimization in order to transform multiobjective problems into single objective ones. Traditionally this has been needed and desired because only single objective optimization solvers have been available and scalarizing has been the only way to solve problems with multiple objectives.

In reference point based multiobjective optimization the purpose of a scalarizing function is to find a Pareto optimal solution corresponding to the given reference point $\bar{z} \in R^k$. Mathematically this means projecting $\bar{z}$ to the Pareto front when possible and otherwise searching a Pareto optimal solution within minimal distance from the general direction given by the projection. When using scalarizing functions with reference points this is the very manner of transforming multiple objectives into only single one. After transforming the problem into a form of having only one objective, it can be solved using traditional single-objective optimization methods.

Miettinen and Mäkelä (2006) describe how usage of scalarizing functions is not dependent on what kind of decision making method is used in the optimization: In the no-preference method the "best solution" can be calculated by using some scalarizing. In an a priori method, the solution corresponding to the preferences of the decision maker has to be calculated using scalarizing. In an a posteriori method, all the Pareto optimal solutions can be calculated using multiple reference points and scalarizing functions. In interactive methods the solutions corresponding to the decision maker's preferences at each iteration can also be calculated using scalarizings.

Considering the scalarizing functions Miettinen and Mäkelä (2002) emphasize that different scalarizing functions give different solutions with the same reference points. Miettinen and Mäkelä (2006) also warn that by choosing one scalarizing the method developer fixes the solution corresponding to the given references. As an example,

differences of three different scalarizing functions in a two-objective problem are presented in Figure 2.

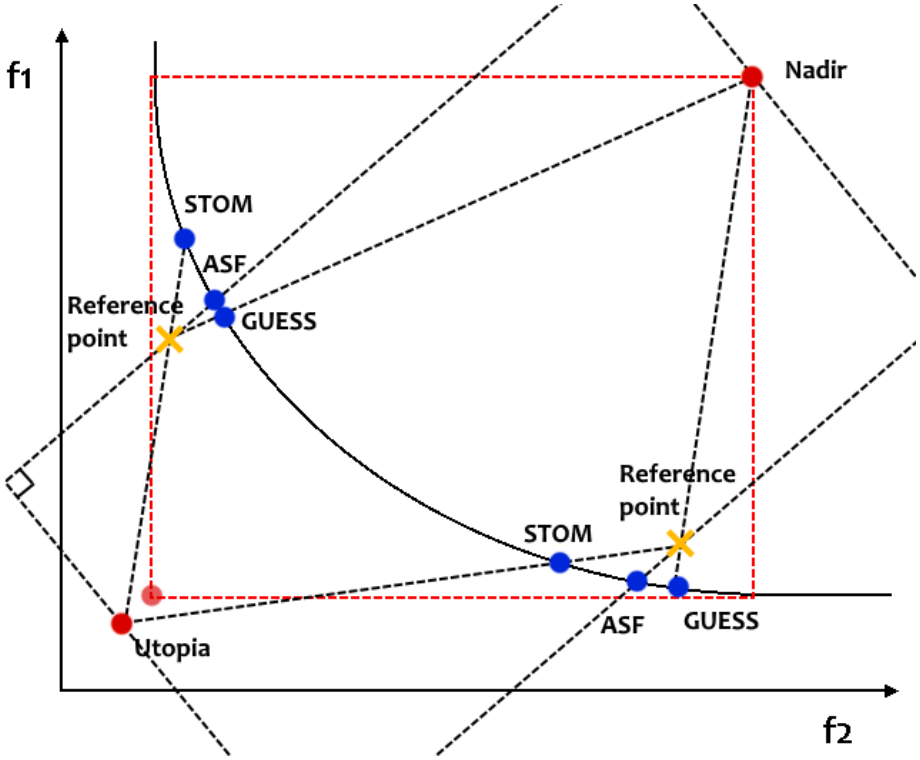

Figure 2: The projecting directions of ASF, STOM and GUESS scalarizings.

In Figure 2 two reference points are marked with yellow crosses and corresponding Pareto optimal solutions in blue. The black dotted lines mark the main search directions of different scalarizing functions: STOM scalarizing uses reference point and utopia point, GUESS scalarizing uses reference point and nadir point and ASF function uses all of these points for deciding the directions. Depending on the reference point and the Pareto front there may not be any difference between two different scalarizing functions.

Because of the differences in the behavior of the scalarizing functions, multiple different scalarizings can be used in an individual optimization process. For example in the synchronous NIMBUS approach by Miettinen and Mäkelä (2006) four different scalarizings are used. These scalarizing functions, Achievement Scalarizing Function (ASF), with its variations Satisficing Trade-Off Method (STOM) and GUESS, and also the NIMBUS scalarizing are presented in the following section.

2.3.1 Achievement Scalarizing Functions

The achievement scalarizing functions are formed for each problem using all the objectives of the multiobjective optimization problem and the preference information given in the form of a reference point, as stated by Miettinen and Mäkelä (2006). Originally the achievement scalarizing functions (ASF) have been introduced by Wierzbicki (1982).

According to Wierzbicki (1982) the purpose of a system using ASF is to inform user if her/his preferences are attainable or not and to generate Pareto optimal solutions. If the preferences are unattainable, the system should give Pareto optimal solutions closest to the given preferences. Where the preferences are over-attainable i.e. worse than what is possible, the system should give Pareto optimal solutions that are better than the preferences. If the given preferences are already just attainable, so Pareto optimal, then the system should give the same solution.

When using ASFs in collaboration with reference points, Wierzbicki (1986) has proved that these achievement functions have a property of full controllability. Miettinen, Ruiz, and Wierzbicki (2008) explain that in practice this means that the decision maker can find every existing Pareto optimal solution using different reference points. Thus, reference points and scalarizing functions provide a meaningful tool for the optimization.

Nowadays there exist many variants of ASF and in this thesis the basic formulation of the problem is considered:

$$\begin{aligned} &\text{minimize} \quad \max\left[\frac{f_i(x)-\bar{z}_i}{z_i^{nad}-z_i^{\star\star}}\right]+\rho\sum_{i=1}^{k}\frac{f_i(x)}{z_i^{nad}-z_i^{\star\star}} \\ &\text{subject to} \quad x\in S \end{aligned} \tag{2.2}$$

where $\bar{z}_i \in R^k$ is the reference point given by the decision maker, $\rho > 0$ is a relatively small scalar, called *augmentation coefficient* and the other variables are as defined in Section 2.1. The additional term including ρ in the formulation is called the

augmentation term, and it guarantees that the generated solution is indeed Pareto optimal instead of being just weakly Pareto optimal, as Miettinen and Mäkelä (2006) have proved.

The Satisficing Trade-off method by Nakayama and Sawaragi (1984) and the GUESS method by Buchanan (1997) can be considered as variations of the original ASF method. As can be seen from the following, their only differences to the basic ASF formulation are in the denominators of the min-max terms and augmentation terms.

The formulation for the STOM is of form:

$$\begin{aligned} &\text{minimize} \quad \max\left[\frac{f_i(x)-z_i^{\star\star}}{\bar{z}_i-z_i^{\star\star}}\right]+\rho\sum_{i=1}^{k}\frac{f_i(x)}{\bar{z}_i-z_i^{\star\star}} \\ &\text{subject to} \quad x\in S \end{aligned} \tag{2.3}$$

where each element $\bar{z}_i$ of the reference vector $\bar{z}$ has to be strictly greater than the corresponding element $z_i^{\star\star}$ of the utopian objective vector $z^{\star\star}$.

The GUESS method is formulated as:

$$\begin{aligned} &\text{minimize} \quad \max\left[\frac{f_i(x)-z_i^{nad}}{z_i^{nad}-\bar{z}_i}\right]+\rho\sum_{i=1}^{k}\frac{f_i(x)}{z_i^{nad}-\bar{z}_i} \\ &\text{subject to} \quad x\in S \end{aligned} \tag{2.4}$$

where each element $\bar{z}_i$ of the reference vector $\bar{z}$ has to be strictly smaller than the corresponding element z_i^{nad} of the nadir objective vector z^{nad}.

2.3.2 NIMBUS

In addition to the classification presented in Section 2.2.2, the NIMBUS method also includes its own scalarizing. There exists multiple NIMBUS variants, but the so-called synchronous NIMBUS method by Miettinen and Mäkelä (2006) is presented here. After the decision maker has classified the objectives according to her/his preferences, a scalarizing of the following form is used:

$$\begin{aligned} \text{minimize} \quad & \max_{\substack{i\in I^{<} \\ j\in I^{\leq}}} \left[\frac{f_i(x)-z_i^{\star}}{z_i^{nad}-z_i^{\star\star}}, \frac{f_j(x)-\hat{z}_j}{z_j^{nad}-z_j^{\star\star}} \right] + \rho \sum_{i=1}^{k} \frac{f_i(x)}{z_i^{nad}-z_i^{\star\star}} \\ \text{subject to} \quad & f_i(x) \leq f_i(x^h) \ \text{ for all } \ i \in I^{<} \cup I^{\leq} \cup I^{=}, \\ & f_i(x) \leq \varepsilon_i \ \text{ for all } \ i \in I^{\geq}, \\ & x \in S, \end{aligned} \tag{2.5}$$

where variables are as defined in (2.2) and the classes $I^{<}, I^{\leq}, I^{=}, I^{\geq}$ and $I^{\diamond}$ as described in Section 2.2.2.

In the synchronous NIMBUS method, there is possibility to generate up to four different Pareto optimal solutions from the given classification. This is done by transforming the classification into a reference point $\hat{z}$, and by using that point with the ASF, STOM and GUESS scalarizings functions in addition to the NIMBUS scalarizing function.

3 Data-driven Optimization

In many optimization problems the variables and functions for all the objectives are well known from the previous research or it is possible to formulate them by studying carefully the problem at hand. Good examples of these are Sindhya et al. (2017) where previous research and Chugh, Sindhya, Miettinen, et al. (2017) where studying the problem were used. For the analyst this kind of situations are ideal.

Today, in the era of big data, the problems that are encountered and desired to be optimized are all the time more complex and unique. The problems may be so specific that there does not exist any previous research and in many cases, the exact or even approximate connections between variables and objectives are not easily discovered. Luckily, there may be available some measurement or simulation data concerning the problem. The modeling and optimization of a blast furnace by Chugh, Chakraborti, et al. (2017) is a good example of this.

Contrary to solving the traditional purely mathematical optimization problems, solving this kind of optimization problems that are based solely on data is called *data-driven optimization*, as Wang, Jin, and Jansen (2016) define. They also elaborate the definition by continuing that the data may be included in the mathematical formulation of the problem as coefficients or there may not be any mathematical formulation at all at the beginning.

3.1 Basics of data-driven optimization

According to Knowles and Nakayama (2008) the problem that emerges with real life data-driven optimization problems is that they do not have any analytic form and it is extremely challenging to describe them accurately using mathematical functions. For that purpose, there still exist numerical methods that enable simulating complex systems and phenomena. The simulations used in drug discovery (Grave, Ramon, and Raedt: 2008) and aerodynamics testing (Giannakoglou: 2002) are good examples of these. These numerical models do not give us the explicit mathematical models

but only enable replicating the phenomenon in a controlled environment. From this it follows that it is not possible to find the optimum directly and the only means are iterative methods.

A commonly used approach for solving that kind of data-driven optimization problems are *Evolutionary Algorithms* (EAs) (Deb: 2012). The strength of these EAs is that they do not require the optimization to be at any analytic form. Instead of exploring or exploiting mathematical properties of the problem, optimizing is based on finding the optimum by generating and evaluating solutions iteratively.

Deb (2012) describes the evolutionary algorithms being based on nature-inspired ideas of reproduction, mutation, recombination and selection of the most fit individuals. In short, these algorithms have an initial population of randomly generated solutions and fitness of each solution is measured by using some predefined fitness metric. After evaluating all the solutions, the fittest ones are selected for mating. Mating of solutions means recombining two or more solutions in order to generate new solutions, using so called crossover and mutation operators. The aim of this procedure is to transfer the best features of the existing solutions to the next generation and to generate new hopefully better features for the offspring. The fitness values of the new generation are then also measured and the best ones from the entire population are selected to be the starting population for the next iteration. This way increasingly better solutions are generated via multiple iterations and the process is repeated until good enough solution is found or the pre-set limit for iterations is reached.

The problem with data-driven problems and using EAs is that the number of solution evaluations required for finding the optimum is usually large. Whether using original data or complex simulations about the original phenomena, the computational cost may be high. This can cause the usage of EAs to become expensive and even unfeasible.

These difficulties of using the complex model can be overcome by simplifying the complex model into so called *meta-model* or *surrogate*. This can be done by modeling only the relevant parts of the original complex model and not all the underlying processes and relationships. Knowles and Nakayama (2008) describe how these

computationally cheaper models can be used in the optimization reducing significantly the complexity of the task and thus enabling optimization of data-driven problems. More of these surrogates is described in Section 3.2 and one such a model is presented in Chapter 4 as the main contribution of this thesis.

Wang, Jin, and Jansen (2016) divide data-driven optimization into two categories: to online and offline data-driven optimization. The online data-driven optimization means that there is new data available during the optimization phase whereas in the offline data-driven optimization no new data is available.

In the online data-driven optimization, the new data may be available either by conducting new experiments, running simulations or by some other means. Wang, Jin, and Jansen (2016) also divide the online data-driven optimization into two subclasses: 1) Online data-driven optimization with uncontrolled incremental data, and 2) Online data-driven optimization with controlled incremental data. In the case of uncontrolled incremental data, the data may come from some industrial process etc. where it is not possible to control the new data or the controlling would alter the conditions of the problem studied. One example of this is the optimal control of Hematite grinding process by Dai, Chai, and Yang (2015). In the controlled online optimization, the data can be completely controlled by the optimization process, so new data comes from explicit function evaluations, simulations or, for example, from a chemical process as in the Grave, Ramon, and Raedt (2008).

In the offline data-driven optimization, all the data is gathered before the optimization. It can be that the data used in the optimization is collected from events that occur only by accident, or some other way that cannot be controlled. For example, the optimization of a Fused Magnesium furnace by Guo et al. (2016), uses previously collected data to optimize future operations. In practice, these different classes of data-driven optimization mean different strategies for managing the surrogates during the optimization.

More of these strategies and surrogates in general is described in Section 3.2 and the general challenges related to the data-driven optimization and usage of surrogates are discussed in Section 3.3.

3.2 Surrogates

The surrogates are computationally cheaper and simpler models that are used in calculations instead of the original computationally expensive complex models. Their purpose is to capture only those aspects of the original problem that are relevant for solving the problem. Many times the word meta-model – *model of the model* – or *function approximation* are used to describe this. However, because the purposes of a computationally cheaper model can be many, the more general word *surrogate* is preferred and used in this thesis.

As classified by Jin (2005), surrogates can be applied by three different ways in optimization: as *problem approximation*, as *functional approximation* or as *objective approximation*. The problem approximation means replacing the original problem with some problem that is similar enough, but easier to solve. Examples of this are using computational fluid dynamics (CFD) simulations instead of real wind tunnel experiments. The functional approximation means using some explicit expression instead of real evaluations of the original model, e.g. using a constructed mathematical model instead of CFD evaluations. The objective approximation, or *evolutionary approximation*, is usually specific for EAs, and it means approximating objective values using one or more already known objective values, that are somehow close or related to the new values. From these three the function approximation approach is the most common.

Knowles and Nakayama (2008) mention two benefits of using a surrogate model. Firstly, the number of full cost evaluations of the original model can be greatly reduced while still having accurate surrogate model. This is essential when trying to solve an optimization task using a surrogate model, for instance. The second advantage is that

the surrogate model may present valuable information about the problem itself in a simpler and possibly more understandable form.

The following Section 3.2.1 describes the steps for constructing surrogates, Section 3.2.2 presents different types of surrogates and Section 3.2.3 shows some special aspects when using surrogates for multiple objectives.

3.2.1 Constructing surrogates

The process of building a surrogate model can be described using the following steps that are combined from Knowles and Nakayama (2008) and Forrester and Keane (2009):

1. The variables to be used in the optimization are chosen, as determined by preliminary experiments.
2. Surrogate model type(s) is selected.
3. Initial samples are selected from the original model/data.
4. From some or all the samples build/update surrogate model(s).
5. Using the surrogate model(s), choose a new sample of points and evaluate them on the original model.
6. Until some stopping criterion, return to 4.

Each of these steps is explained more profoundly in the following.

1) The first step, selecting the right aspects of the problem for the model, is a cumbersome problem itself. Since every new variable increases the computational cost, one must choose a subset of variables that can be managed computationally while still capturing all the necessary aspects of the original model. Forrester and Keane (2009) instruct that the selected subset should particularly contain enough variables to define the entire decision space.

In optimization, the selection of input and output variables is an essential part of formulating the optimization problem itself. When using a surrogate model, it is natural that the variables used for building the surrogate are the same variables that

will be used in the optimization also. This problem formulation and variable selection is an iterative process and closely related to the Design of Experiment (Anderson and McLean: 1974). The process of designing and conducting experiments has been of great interest for a long time and solutions for automating the process have been proposed, for example, by King et al. (2004). More about variable selection can be found in Guyon and Elisseeff (2003)

2) The selection of the surrogate type depends on the problem. The goal of building a surrogate for optimization is to form a model that produces accurate predictions of the original model, especially when close to optimum. Because all nor any surrogate type is suitable for all kind of problems, it is important to choose a surrogate type corresponding to the specific needs and features of the problem at hand.

The theoretical reasons for this are explained in the "No Free Lunch Theorems for Optimization" by Wolpert and Macready (1997). The theorems state that when averaged over all the possible optimization problems, all the optimization algorithms perform the same. From this it follows that if some algorithm is performing better than average on some optimization problems it has to be performing worse than average on some other type of optimization problems. From the surrogate based optimization perspective English (2000) has proven that learning is hard and optimization easy for typical functions. Because of this surrogate type and optimization methods have to carefully selected.

According to Knowles and Nakayama (2008) the main aspects in the selection of the surrogate type are the dimension of the decision space and the type of variables. The dimension of the decision space is important in the respect that some surrogate models handle high dimensions better than the others, while some models that perform well in high dimensions may fail to produce good results in low dimensions. Good examples given by Zhou et al. (2005) are Gaussian processes and polynomial regression models, from which the former performs well in high dimensions and the latter with fewer dimensions. The types of the function variables, i.e. whether they are discrete or continuous, also affect greatly the selection of the surrogate type. Knowles and Nakayama (2008) note that even though most surrogate models and methods

are developed for the continuous variables, some machine learning methods such as classification and regression trees, genetic programming and Bayes' regression may be suitable for discrete decision spaces also.

The other features of the original model affecting the selection of the surrogate, as listed by Forrester and Keane (2009), are mathematical size of the problem, expected complexity, the cost of the evaluations of the model and also whether and how it is possible to acquire new data during the optimization phase i.e. online or offline optimization. In addition other information about the structure of the problem can be exploited in the selection of the surrogate.

One possibility mentioned by Jin (2011) is to form multiple different surrogates for one single problem. These surrogates may be of different fidelity or type and they may be used for different phases of optimization or each surrogate is formed for different part of the decision or product space. These so called *surrogate ensembles* can also help in locating possible prediction errors: if different surrogates produce very different results for some parts of the problem, there might be some errors involved in the models. Different measures and criterion for the surrogates are presented, for example, by Jin, Husken, and Sendhoff (2003) and Husken, Jin, and Sendhoff (2005).

3) After choosing the decision variables and the surrogate type(s), the next step is initial sampling. When considering globally unknown decision spaces, it makes sense to use uniform sampling across all the decision variable axes. For that purpose there exists multiple different sampling techniques, such as the Latin hypercube sampling by Morris and Mitchell (1995).

Forrester and Keane (2009) name the chosen surrogate type and the way new data is acquired before the optimization as the main forces affecting the sampling. The surrogate type may define or constrain how the initial sampling can be done, and this naturally has to be addressed in sampling. The type of how new data acquired has already dictated the selection of the surrogate type, but it also affects how the initial sampling can be done, whatever the surrogate type is.

In the offline data-driven optimization no fresh data is acquired and so the sampling set is limited. Because all the data is available since the beginning of the building, it has to be decided whether to use all the available data points to build the surrogate or not. If the number of available data points is large, using all of them may be computationally too expensive or result to so called overfitting, which means that the surrogate is too detailed in order to simplify the problem. On the other hand, in online data-driven optimization the sampling scheme is less limited, except the constraints set by the problem domain or the problem formulation itself.

4) According to Knowles and Nakayama (2008) the surrogate model may be built or updated using all or just some of the selected new points, depending about the sample selection scheme. The actual way of building or updating the surrogate depends on the surrogate management strategy and type of the surrogate chosen. This is not a trivial issue and is not really discussed in the surrogate/meta-model literature, as Chugh, Sindhya, Hakanen, et al. (2017) note.

5) After building the initial surrogate model, new sampling points may be selected and applied to the surrogate iteratively until certain accuracy is reached. This can also be done during the optimization and according to the needs of the optimization process. The new samples can be chosen using only the current model, for example, in finding the optimum of the approximate model, or decided based on previously searched or considered points as is done in some EAs.

The choice of the new points is then dependent on the surrogate type and the surrogate management strategy. Again, the same way as during the initial sampling, these both are dependent on how new data is acquired during the optimization.

Knowles and Nakayama (2008) separate two possible criterion for selecting new sampling points: predicted value or estimated informativeness of the points. Choosing points based on their predicted values aims to choose points that are high-performing by themselves, whereas choosing based on the estimated informativeness tries to learn as much as possible about the features of the decision space. These two approaches

are naturally contradicting with each other, so selecting new samples based on these qualities is not trivial and some compromise between these has to be made.

6) Whether the surrogate is built entirely before optimization or updated during the optimization, the process is iterative by nature. During the iterative building process the surrogate model is improved on every iteration, but at some point this iterating process has to be ended and the model approved to be ready. The ending can be based on the desired accuracy of the built model, when good enough optimum for the surrogate model is acquired etc.

Knowles and Nakayama (2008) note that building a surrogate resembles classical regression models and machine learning. If one is going to make only one initial sampling for building the surrogate, the sample has to be carefully planned and this kind of surrogate building resembles a classical regression model building the most. If the surrogate model is to be updated during the modeling or the optimization, machine-learning methods are needed as the surrogate building must be combined with those phases in a more sophisticated manner.

3.2.2 Types of surrogates

There are multiple different surrogate types developed for machine learning and so used for optimization purposes. These include linear, nonlinear or polynomial regression models (Zhou et al.: 2005), Response Surface Method (RSM) (Myers and Montgomery: 1995), Radial Basis Function Networks (RBFN) (Broomhead and Lowe: 1988), Support Vector Machines (SVM) (Loshchilov, Schoenauer, and Sebag: 2010), Kriking or Gaussian processes (Zhou et al.: 2005; Cheng et al.: 2015) and Artificial Neural Networks (ANN) (Gaspar-Cunha and Vieira: 2005). More about different surrogates and their usage in optimization can be found in Tabatabaei et al. (2015) and Chugh, Sindhya, Hakanen, et al. (2017).

It is notable that a lot of work considering data-driven optimization and surrogates is strongly related to evolutionary algorithms, as surveys by Jin (2011) and Chugh, Sindhya, Hakanen, et al. (2017) reveal. The reason is that these algorithms are

quite well suited for the problems whose mathematical properties are not well known beforehand, as the most data-driven problems are. As EAs require a great number of function evaluations to optimize the problem, computationally cheaper evaluation methods, surrogates, are naturally of great interest. Wang, Jin, and Jansen (2016) also note that the literature of the field is mostly focused on online data-driven optimization with controlled incremental data.

Even though the surrogates are by far mostly used in evolutionary computations to reduce the amount of expensive evaluations, they are still not the only applications of such computationally cheaper models. While not replacing expensive evaluations as with EAs, the surrogates can also be used in the traditional linear and non-linear optimization in order to simplify large-scale mathematical programming problems.

3.2.3 Surrogates for multiple objectives

In the surrogate-based optimization the difference between multiobjective and single objective optimization is mostly in the surrogates. When there are multiple objectives Knowles and Nakayama (2008) suggest to build a surrogate for each objective function separately and using the surrogate models in the optimization. However, according to Chugh, Sindhya, Hakanen, et al. (2017), this far the most common approach for surrogate based multiobjective optimization has been using just one surrogate for all the objectives.

This kind of surrogate representing multiple or all the different objectives is called *mono surrogate*. In practice this can mean forming a surrogate for scalarizing functions as in ParEgo algorithm by Knowles (2006), surrogate presenting the entire Pareto front of the problem as in PAINT method of Hartikainen, Miettinen, and Wiecek (2012) or surrogate presenting just some parts of the front as in Pareto Navigator by Eskelinen et al. (2010).

Survey by Chugh, Sindhya, Hakanen, et al. (2017) points out that it is possible to use different surrogate models separately for all the objectives, for all the constraints or for different scalarizing functions. As can be deduced, it is then also possible to form

different combinations of these surrogate models i.e. surrogate for multiple but not all of the objectives, surrogates combining constraints or objectives etc. Because of this, the spectrum of different options is quite overwhelming, and unfortunately there does not exist any general guidelines for choosing the most suitable approach.

3.3 Challenges in data-driven optimization

Many challenges faced in the data-driven optimization and in the usage of surrogate models are already described in the previous sections, where they have been clearly connected to the topics covered. This section presents more general challenges that are not directly connected to any aspect presented before. The challenges described in this section are computational cost, data quantity, data quality, heterogeneity of data, prediction accuracy and constraint handling.

In the data-driven optimization there are multiple reasons for computational costs, as listed by Wang, Jin, and Jansen (2016). The used model can be based on heavy simulations or on computationally expensive functions, which make the sampling a computationally heavy task. In addition, the monetary value of samplings can be high, if they are such that real life experiments have to be arranged. The traditional car crash tests or chemical drug testings are examples of this kind of costs. The costs can also be related to gathering the data, which in the case of large and heterogeneous data can be time and resource consuming.

A sometimes neglected cause of computational costs in the surrogate-assisted optimization is training time requirements of the surrogate models as Chugh, Sindhya, Hakanen, et al. (2017) point out: The purpose of surrogate models being the simplification of the optimization, there is a risk that if the resource requirements for building the surrogates are not addressed, they may suddenly impair all the optimization benefits of the surrogate. Especially in the multiobjective optimization where one may have multiple different surrogates, all with different training requirements, the resource requirements can be surprising high, as Chugh, Sindhya, Hakanen, et al. (2017) note.

According to Wang, Jin, and Jansen (2016) the greatest difficulties concerning data quantity emerge when there is too much or too little data. If there are significantly great amounts of high dimensional data, the problem is how to choose relevant features for the surrogate forming and which data points to choose for sampling. The other side of that is the computational costs: the more features and data points are included in the modeling, the higher is the computational cost, but more accurate the surrogate model. Guo et al. (2016) also mentions that if the amount of data is small, the forming of accurate surrogate becomes extremely challenging.

In the data-driven optimization the quality of the data directly affects the quality of the surrogate. If the data used for building the surrogate is distorted or inaccurate, the surrogate built from the data is that also. When using that kind of surrogate for optimization, this naturally affects the optimization outcomes. The quality issues of data include uncertainty, ill-distribution, imbalance (Wang and Yao: 2013), incompleteness (Arbuckle: 1996) and contamination by noise (Wang et al.: 2016).

In the era of large scale and ubiquitously collected data, one great challenge remarked by Wang, Jin, and Jansen (2016) is heterogeneity of data. The data can be in many different formats and levels of reliability or accuracy. These aspects raise many questions about how to handle all the different formats with relation to one another, and all these questions have to be answered somehow. In the presence of heterogeneous data, the importance of preprocessing cannot be overemphasized.

When constructing a surrogate there is always some error between the model and the original data: otherwise the surrogate could not be any simpler than the original model. This aspect of constructing a surrogate has to be taken into account and controlled when needed. The existence of errors is inevitable, but in optimization they can be tolerated as long as they don't misguide the search as Jin, Husken, and Sendhoff (2003) point out. Lim et al. (2010) also remarks that in some surrogates the uncertainty of the surrogate can even be exploited. However, these issues have to be addressed consciously.

One often overlooked aspect of using surrogate models mentioned by Chugh, Sindhya, Hakanen, et al. (2017) is handling constraints. As with the number of objectives, the greatest problem in this is the fact that most surrogate based algorithms are simply not developed to handle any constraints. When solving (multiobjective) optimization problems having constraints they have to be addressed somehow.

This chapter has introduced foundations and issues of data-driven optimization and surrogates in general. The following Chapter 4 explains how a cluster based surrogate is constructed and the presented aspects are taken care of.

4 Cluster Based Interactive Multiobjective Optimization

Usually the surrogates are computational models such as polynomials or function networks, as described in Section 3.2. However, this is not the only way of defining a surrogate model. In addition to those models, a surrogate can be formed just by choosing a part of the data for estimating all the values instead of using all the data available. An example of that is presented by Wang, Jin, and Jansen (2016), who used a dynamic clustering technique to select only a subset of representative individuals to approximate the entire data set. This way the computational costs of the objective and constraint evaluations were reduced significantly without degrading the performance of solutions.

Based on the earlier presented remarks about possibilities for using surrogates also with other approaches in addition to EAs, and on already attained promising results of using clustering to reduce computational burden of optimization, this chapter introduces a cluster based decision space surrogate for large scale Integer Linear Problems (ILP). A cluster based surrogate is introduced in Section 4.1 and combining it with optimization to solve ILP problems in Section 4.2. Case study where a cluster based surrogate is used is described in Chapter 5. Discussion about the surrogate and method presented in this chapter can be found in Chapter 6.

4.1 Clustering as surrogate

This section presents the core of a cluster based surrogate which is the main contribution of this thesis. Clustering as general method is described in Section 4.1.1 as preliminary information. After that, clustering in the decision space for forming a cluster based surrogate is explained in Section 4.1.2.

4.1.1 Clustering

Arabie, Hubert, and Soete (1996) characterize clustering as:

> *"Those methods concerned in some way with the identification of homogeneous groups of objects, based on whatever data are available."*

This definition of clustering is quite broad but also reflects the difficulty of giving formal and accurate definition for clustering. Usually this definition is still extended by stating that objects in the same group, i.e. cluster, are supposed to be more similar to each other than to objects in different clusters, as Everitt, Landau, and Leese (1993) state. The similarity or distance between objects is dependent on the data used and what aspects are regarded important in the specific case.

The clustering itself can have hard or soft assigning, meaning that the objects belong to one and only one cluster, or each object belongs to each cluster to a certain degree. Depending on the application of the clustering, only either hard or soft clustering is chosen. For the purposes of a cluster based surrogate only hard clusterings are considered in this study later on.

Jain (2010) presents that clustering of data usually serves one or more of the following purposes: exposing the underlying structure, finding the natural classification, or compressing the data. Exposing the underlying structure of data is useful for gaining insights of data, generating hypotheses, detecting anomalies etc. The natural classification may enlighten similarities among individuals and their relations. The compression works for organizing and summarizing the data by generalizing it. In a cluster based surrogate the clustering is used mainly for compressing purposes.

For the actual clustering there exists plethora of different clustering algorithms in the literature. Fahad et al. (2014) have separated different algorithms into partitional, hierarchical, density-based, model-based and grid based algorithms. From these five, partitional algorithms are probably the most common, and so some of their characteristics are explained in the following.

As described in Jain (2010), partition based clustering algorithms are given the number of clusters beforehand and the algorithms find all the clusters from the data simultaneously, without any hierarchical structure. Fahad et al. (2014) enlightens this explaining how these algorithms divide data into a number of partitions, where each partition represents a cluster.

Determining automatically the number of clusters K is one of the greatest problems in clustering regardless of the class of algorithm. Especially in the partition based clustering algorithms the number of clusters is of great interest, as it has to be decided before the actual clustering. This problem is usually addressed by forming multiple different clusterings with different number of clusters; from these the best number is chosen afterwards based on a predefined criterion.

A criterion for deciding the number of clusters is for instance the L-method by Salvador and Chan (2004). This method aims to find the boundary between straight lines that most closely fit the curve of measuring the validity of a clustering. In practice this means finding the number of clusters after which the validity of clustering does not increase much with the increase in the number of clusters. This is better illustrated in Figure 3.

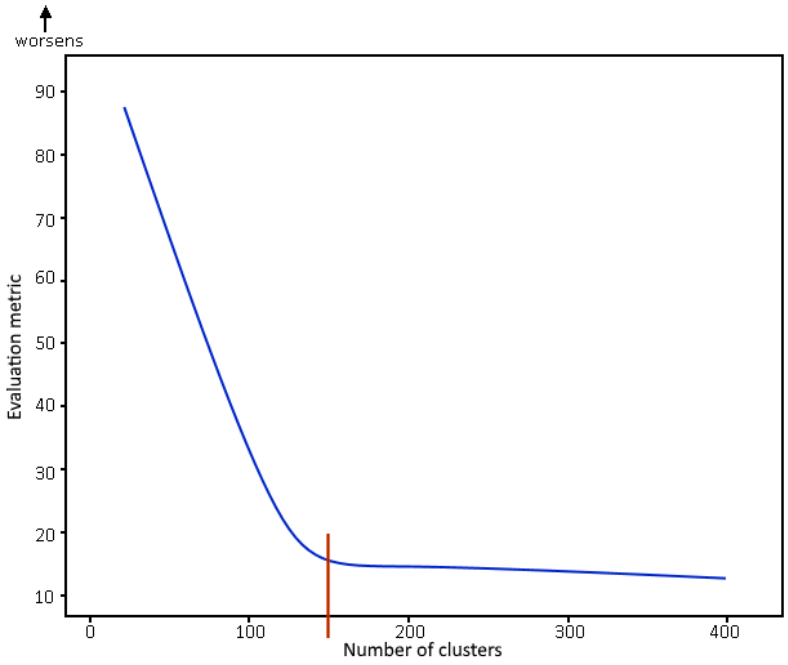

Figure 3: The concept of the L-method. Blue line marks the evaluation values of clusterings. Red straight line marks the "knee" at 150 clusters.

The most popular and one of the most simple partition based clustering algorithms is K-means. The algorithm aims to form a partition in which the squared error between the center of cluster and the points in clusters are minimized i.e. the distances between cluster center and the data points within the cluster are desired to be as small as possible. The steps in K-means algorithms according to Jain and Dubes (1988) are:

1. Select an initial partition with K clusters and calculate cluster centers; repeat steps 2 and 3 until memberships of clusters do not change anymore
2. Generate a new partition by assigning each point to the nearest cluster center
3. Calculate new cluster centers

These steps are better illustrated in Figure 4.

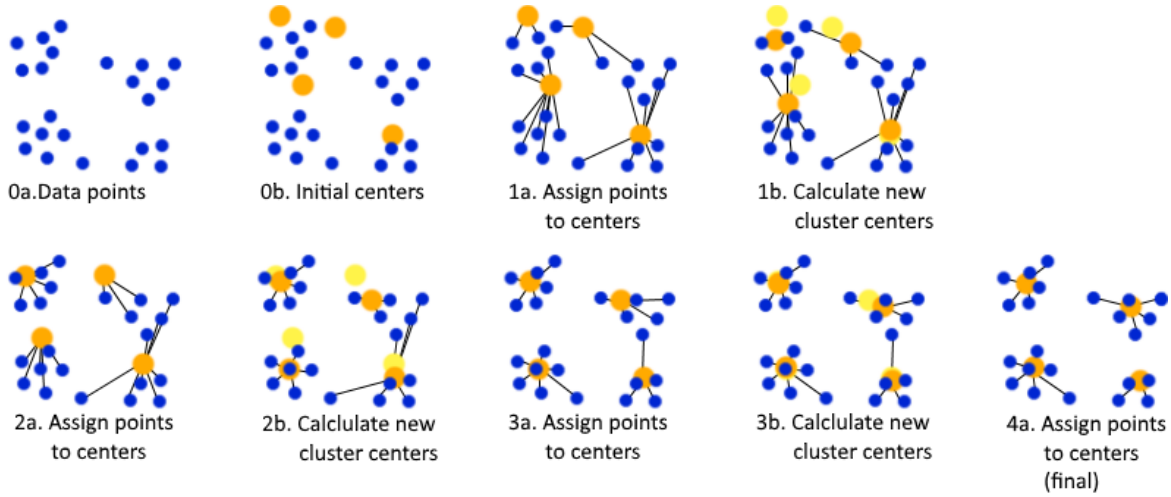

Figure 4: The steps of K-means algorithm.

The algorithm is greedy by its nature, which means that it will always converge to local minimum only. Because of this, the assigning of the first cluster centers and so the initial partition may affect the final clustering greatly. One way to tackle this challenge is running the algorithm many times with fixed K, but with different initializations. Afterwards the best clustering is chosen according to the smallest squared error or some other comparison metric, as e.g. Jain (2010) suggests.

When combining all the aforementioned challenges and features of clustering and K-means together, there are three parameters to choose for the K-means clustering:

similarity or distance metrics, number of clusters K and initial clustering. When forming actual clustering all theses issues have to be addressed.

There are multiple different variations for K-means clustering using different assigning metrics, median instead of mean or detecting differently shaped clusters. Also many efficient implementations especially for large data sets have been introduced, like Kanungo et al. (2002) and Cui et al. (2014)

Clustering, especially clustering of large and high dimensional data, is an art form by itself and cannot be explained here with the detail it would otherwise deserve. For more information about different algorithms and deeper understanding of clustering itself one may turn to Jain (2010) and Fahad et al. (2014)

Although clustering is usually aimed for any data points regardless of their interpretation, there exists literature and methods considering clustering of variables only. For example in Vigneau and Qannari (2003) and Vigneau et al. (2005) variables were clustered according to their mutual correlations and latent variables. These clusterings are however aimed for continuous variables so extending them to work with discrete ones and for optimization is out of the scope of this work. The next section explains how discrete variables are clustered in this study.

4.1.2 Clustering in decision space

Before constructing a cluster based surrogate, it is assumed that the data to be used in the optimization is selected and preprocessed. It is also required that the variables are discrete and are of similar type defined by some similarity measure, so that it is possible to cluster them in a meaningful sense. In addition to those, it is assumed that the number of variables is large, so that dimension reduction is desired for the benefiting the computations.

These requirements restrict this cluster based surrogate only for problems where the variables, i.e. data, is quite homogeneous. Large number of dimensions is not a requirement, but if that is not the case, the cluster based surrogate may not actually be needed.

The actual forming of a surrogate starts by clustering all of the variables using some hard clustering method, so that original n discrete variables are assigned to $K \leq n$ clusters according to their values. Because of the manner of how the cluster based surrogate is combined with the optimization later on, it is essential that all the variables assigned to same cluster have the same number of discrete value alternatives. This should be just a matter of preprocessing or doing an initial classification before the actual clustering process. Combining the surrogate with the optimization and the exact reason of doing this are enlightened later on.

More important than knowledge about the real life similarities of variables, which many times is not available, is the knowledge about suitable similarity measure for the clustering. Usually, and particularly in the case of hundreds or thousands of variables, it is practically impossible to cluster the data manually. In those cases, it is especially desirable to be able to do the clustering unsupervised but based on justified measures.

As always in clustering, the number of clusters K is important. However, because the ultimate goal of clustering as surrogate is not exploring or modeling real classes in the data, this aspect is not playing similar role than it would otherwise. As in this the clustering is only used for compressing the data it is desired that the number of clusters is rather too large than too small or even perfectly suitable for the data. The idea behind is that every cluster acts as a local approximation for the original data and so the more clusters there are the more accurate the entire approximation is.

The shapes of clusters are also one concern encountered in clustering. Because of the same reasons affecting the number of clusters, the requirements for the optimal shape of the clusters are not the same than in usual. When considering compression of data, the most suitable shape for clusters is rounded. Especially when combined with large number of clusters, the clusters as local approximations are the best if their shapes are all similar and rounded.

The choice of the actual clustering algorithm depends about the type of data and resources available. As the clustering is used only for compressing data via using clusters as local approximations, partitional clustering algorithms that find locally

optimal solutions should be good enough. Anyway, one major reason affecting the choice of the algorithm are the resources available and required for each algorithm. In the optimization where the clustering is used to ease the computational load, it does not make sense to use clustering and surrogate whose forming requires more time and computations than the actual optimization using data without clustering. However, if the problem is e.g. such that the surrogate needs to be formed only once and then optimized multiple times, then it can be acceptable to use significant amount of resources for forming the surrogate also.

After the n variables have been assigned into K clusters, the most "representative" variable x_k is selected from each cluster $k \in \{1, 2, ..., K\}$ as a proxy variable. This "representativity" depends on the similarity measure, shape of clusters and so on the method of clustering used. The one variable to be selected may then be the actual center of each cluster or the variable closest to the center of the cluster etc. If the shape of cluster is rounded as is recommended, it is also recommended to choose the variable in the center of the cluster.

In this point it becomes evident that all the variables in the same cluster have to have the same number of discrete value alternatives. While only one discrete variable is chosen among many others, its ability to represent all the others is greatly impaired if it has either more or less values than the others. Later on when combining the surrogate with optimization problem this aspect becomes even more essential.

The building of a cluster based surrogate is finalized with assigning each chosen variable x_k a weight w_k according to the proportion of variables in the cluster k. For example, if there are 158 variables in cluster k, the corresponding weight for the proxy variable x_k is $w_k = \frac{158}{n}$. These cluster representatives and their corresponding weights are then used in optimization as will be described in Section 4.2.

From the previous description it can be seen that the amount of data, i.e. variables, is not easily an issue for a cluster based surrogate. As the surrogate is mainly based on clustering and there exists powerful clustering algorithms for even billions of data

points in multiple dimensions, like the one presented by Cui et al. (2014), the inability of forming clustering is not going to be problem in most of the cases.

In order the chosen clustering algorithm to work correctly it is desired that the quality of data is good enough. If the data contains a lot of outliers or other undesired anomalies, the clustering may not work as expected even though the number of clusters would be large enough.

4.2 Combining Clustering to Integer Linear Problems

The basic Integer Linear Problem (ILP) formulation for discrete variables is of form:

$$\begin{aligned} maximize & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\ s.t. & \sum_{j=1}^{m} x_{ij} = 1 \\ & x_{ij} \in \{0,1\} \end{aligned} \tag{4.1}$$

where $i \in \{1,2,...,n\}$ denotes index for decision variable and $j \in \{1,2,...,m\}$ index for discrete value alternatives for each variable i. For variable i and value alternative j, x_{ij} has value 1, if value j is chosen for variable i and otherwise 0. With c_{ij} we denote the objective value of the i-th variable resulted from selecting the j-th discrete value alternative. In the data-driven optimization, the values c_{ij} are attained from the data.

For solving this kind of problems there exists many commercial and open source optimization softwares, including IBM ILOG CPLEX[1] and GLPK[2] mathematical optimization solvers.

This (4.1) is the one that needs to be modified in order to combine the cluster based surrogate with optimization. In the modification it is desired to keep the format of

1. https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer
2. https://www.gnu.org/software/glpk/

the problem as similar as possible and so to maintain it understandable for the same optimization solvers also. Keeping the problem as similar as possible aids also the human understandabality of the problem.

In (4.1) n denotes the dimension of the decision space, that is desired to be reduced via the surrogate. As explained in Section 4.1.2, the cluster based surrogate selects total of $K \leq n$ proxy variables, one from each cluster, to be used in the optimization and so the modified ILP formulation using the surrogate is:

$$\begin{aligned} maximize \;\; & n \sum_{k=1}^{K} \sum_{j=1}^{m} d_{kj} y_{kj} w_k \\ s.t. \; & \sum_{j=1}^{m} y_{kj} = 1 \\ & y_{kj} \in \{0,1\} \end{aligned} \tag{4.2}$$

where $k \in \{1,2,...,K\}$, $k \leq n$ denotes index for proxy variable, $j \in \{1,2,...,m\}$ index for discrete value alternatives for each proxy variable k and w_k the weighting coefficient for the proxy variable. Value d_{kj} denotes the objective value of the proxy k when the j-th discrete value alternative is chosen. For variable k and j, y_{kj} has value 1 if j is chosen as for proxy variable k, and otherwise 0.

As can be seen, if $k = n$, then $w_d = \frac{1}{n}$ for all $k \in \{1,2,...,K\}$ and this formulation is identical with the basic ILP formulation. For one part this feature also guarantees the validity of this approach for combining the surrogate and optimization.

Because of the inevitable approximation errors in the cluster based surrogate, optimization result based on the modified problem is not the same than that of the original problem. The bigger the number of clusters in the surrogate, the more accurate is the surrogate and so also the result of the surrogate based optimization. The exact differences and errors depend mostly on the original variables and how their natural features can be captured using this kind of surrogate.

4.3 Interactive Multiobjective Optimization using Cluster Based Surrogate

The following sections present how interactive multiobjective optimization is used in collaboration with the described cluster based surrogate. Section 4.3.1 presents how the surrogate is formed in the case of multiple objectives and Section 4.3.2 how the surrogate based multiobjective problem is solved interactively.

4.3.1 Forming Multiobjective Cluster Based Surrogate

In multiobjective optimization, different objectives can be of different type and so ILP based objective may be only one of them. Because of the dimension reduction in the cluster based surrogate and its complex effects on different type of objective functions, this section focuses on multiobjective optimization when all the objectives are similar linear integer problems having initially all the same variables.

When all the objectives are of the same form presented in (4.1) and the variables are the same for all the objectives, the surrogate for multiobjective case is constructed the same way than described earlier: All the variables are clustered using the desired metrics and each cluster is assigned the weight according to the proportion of variables. The formed surrogate is then combined with the multiobjective ILP formulation the same way than in the single objective case.

$$\begin{aligned} \text{maximize} \quad & \left\{ n \sum_{k=1}^{K} \sum_{j=1}^{m} w_k d_{kj}^1 y_{kj}, \ \ldots, \ n \sum_{k=1}^{K} \sum_{j=1}^{m} w_k d_{kj}^N y_{kj} \right\} \\ \text{s.t.} \quad & \sum_{j=1}^{m} y_{kj} = 1, \\ & y_{kj} \in \{0,1\}, \end{aligned} \tag{4.3}$$

where the superscripts $1,..,N$ denote the number of objective function in the multiobjective problem.

As all the data is used for forming the surrogate and the proxy variables are the same for all the objectives, this surrogate can be described as a mono surrogate. For the

used multiobjective ILP formulation no other type of surrogate could not even be applicable.

What is worth noticing is that different objectives may behave differently in the clustering i.e. approximation errors for different objectives may be different. This feature can already be seen while constructing the surrogate as the relative differences between actual discrete values of the variables and values of their proxies may be different. More than that, the differences become more apparent in the optimization if the real objective values are known and it is possible to compare values attained through the surrogate approach to the real objective values.

These relative errors may cause the surrogate based Pareto front to be shifted from the location of the real Pareto front. In addition to that, the surrogate may also change the shape of the front i.e. relationships between objectives. This is illustrated in Figure 5 using just two objectives.

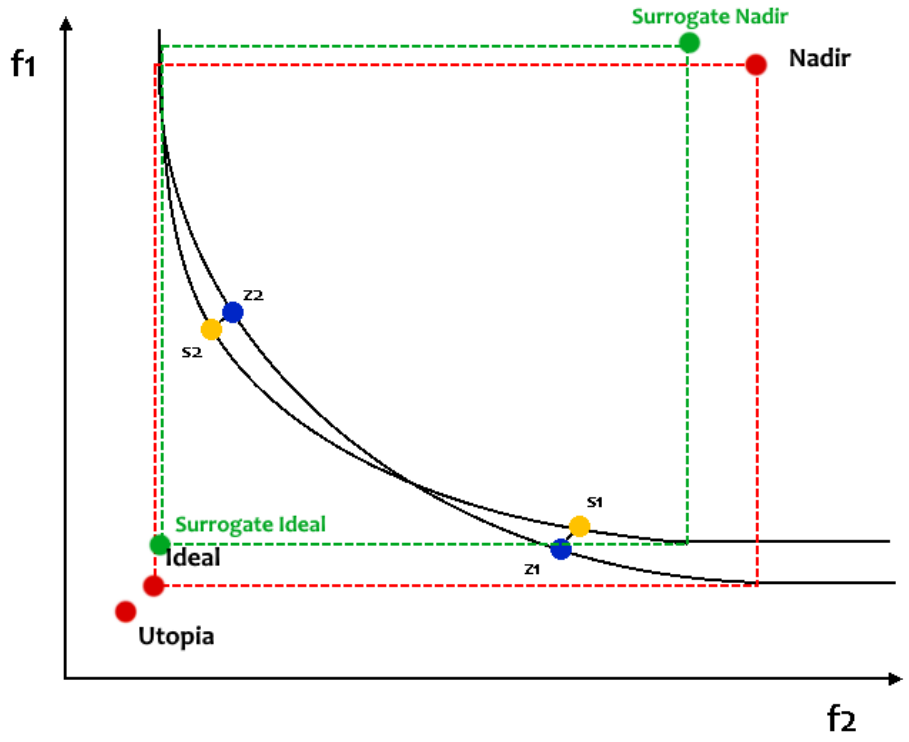

Figure 5: The change in Pareto front when using surrogate in the case of two objectives. Both objectives are to be minimized

In Figure 5 the first objective f_1 has its surrogate based nadir value greater than the real one and the ideal value a bit greater. The other objective f_2 has its nadir value smaller than the real one, and the ideal value a bit greater also. As can be see, the surrogate based Pareto front of this problem is not only shifted but also differently shaped when compared to the real Pareto front. In this example, the surrogate front is neither entirely above nor under the real front, but alternates on the both sides. Because of that we can see that given yellow solutions s_1 and s_2 in the surrogate based Pareto front are on different sides of the real Pareto front and so the corresponding blue real Pareto optimal solutions z_1 and z_2 cannot be similarly predicted from the surrogate based Pareto optimal solutions. This behavior cannot be easily verified, but should be kept in mind when verifying the results attained through the surrogate.

4.3.2 Solving Multiobjective Problem Interactively

The interactive multiobjective optimization methods are developed for finding the most preferred one from the Pareto optimal set. In this study the interactive method of choice is the synchronous NIMBUS method by Miettinen and Mäkelä (2006), which employs four different scalarizing functions. All of these are of non-linear form and because traditional MILP solvers are only capable of solving linear problems, the smooth formulations of these scalarizings, presented by Laukkanen et al. (2010), are used. Coupling these formulations with (4.3) is straightforward.

In the synchronous NIMBUS method it is important to calculate the ideal and nadir vectors, which was already discussed in Section 2.1. When using a cluster based surrogate it is better to calculate these values by using the original variables and problem instead of using the surrogate based values, if possible. Because clustering causes averaging for the objectives, the ideal and nadir values attained form the surrogate are not as far from each other as they really are. Even though the values of the cluster based surrogate are used with the scalarizing functions it is still better to use the original ideal and nadir values in order to attain more widespread solutions.

The interactive process itself does not alter from its original form while using a cluster based surrogate in optimizations: DM gives her/his preferences, explores different Pareto optimal solutions and finally chooses the preferred Pareto optimal solution. The main point in the usage of the surrogate is in reducing the computational burden and so enabling more seamless and less delayed interaction during the process.

In the interactive process it is notable that the objective functions may be nonlinear and non-continuous. This means that there may be some unattainable areas within the ranges of objective, i.e. it may not be possible to find certain objective function values at all. As the cluster based surrogate bundles together multiple variables this property is possibly emphasized even more. This feature is also one reason why the synchronous NIMBUS approach is chosen for solving multiobjective optimization problems when a cluster based surrogate is used. As there may be "holes" in the Pareto front, some single scalarizing function may not find any Pareto optimal solution corresponding to the given reference point. However, because the synchronous NIMBUS is utilizing four different scalarizing functions, it is much more likely to found at least some Pareto optimal solution corresponding to the preferences.

Once the decision maker has found her/his preferred Pareto optimal solution $s^\star$ by using a cluster based surrogate and an interactive method, the solution can be implemented as described in Section 4.2. After that there are two ways to continue the process: either the surrogate based Pareto optimal solutions is projected to the real Pareto front or the values of proxy variables in each cluster are mapped to all the variables in the cluster. These are both explained better in the following.

If it is possible to optimize the original problem without the surrogate, the preferred solution $s^\star$ can be "projected" to the set of real Pareto optimal solutions by using a scalarizing function and the original data without surrogate. If the selection of $s^\star$ was based only on relations of different objectives, the corresponding real Pareto optimal solution $z^\star$ can be calculated by using ASF and $s^\star$ as a reference point. If the decision involved some strict constraints for some objectives, NIMBUS classification can be used.

On the other hand, if it is not possible to optimize the original problem at all without the surrogate, the values of proxy variable y_{kj} can be mapped back to the original variables so that for every variable i in the cluster k, the value x_{ij} is assigned according to the value of the proxy variable y_{kj} i.e. all the variables in the same cluster are chosen the same value alternative. Inputting these variable values to the original problem produces a solution $z^{\star}$, that is more accurate and usually more close to the real optimum.

Independent of which approach of these is chosen, the $z^{\star}$ is presented to DM and her/his satisfaction is ensured. If $z^{\star}$ is not satisfactory certain phases since the construction of surrogate to the final projecting may be repeated, depending on the reason of DMs discontentedness. Consequently, interaction with DM has to be maintained during the entire process and until the very end.

5 Case study

In this chapter a data-driven multiobjective Forest management optimization problem is presented and solved by using a cluster based surrogate and the interactive synchronous NIMBUS approach.

The chapter is organized as follows: Section 5.1 describes the data used in the problem and how it has been used in earlier studies. The following Section 5.2 defines the single-objective optimization problem formulations used for the final multiobjective optimization problem and results of their optimization. The surrogate building process is documented in Section 5.3 and the single objective optimization results attained by using the surrogate are presented in Section 5.4. The final multiobjective optimization problem using surrogates is presented in Section 5.5 and remarks about the interactive solution process with the decision maker in Section 5.6. Section 5.7 describes some observations, analyses and discussion about the executed case study.

5.1 Data of the problem

The data used as an example in this thesis comes from the studies of the Biology Department of the University of Jyväskylä. It describes a total of 29 666 forest stands, i.e. groups of trees that are more or less homogeneous, with total area of 68 700 ha. The area of the forest stands is presented as a map picture in Figure 6. The forests belong to typical Finnish production forest landscape and most of the stands have been under active forest management for several decades. For more details about the forest data, one may turn to Mönkkönen et al. (2014), Peura et al. (2016), Triviño et al. (2015, 2017), and Pohjanmies et al. (2017).[1]

The decision making problem related to the data is about four different forestry objectives and how they are affected by seven different forest management strategies. From the practical perspective the problem is how each of the 29 666 forest stands

1. Data available: `https://dvn.jyu.fi/dvn/dv/Boreal_forest`

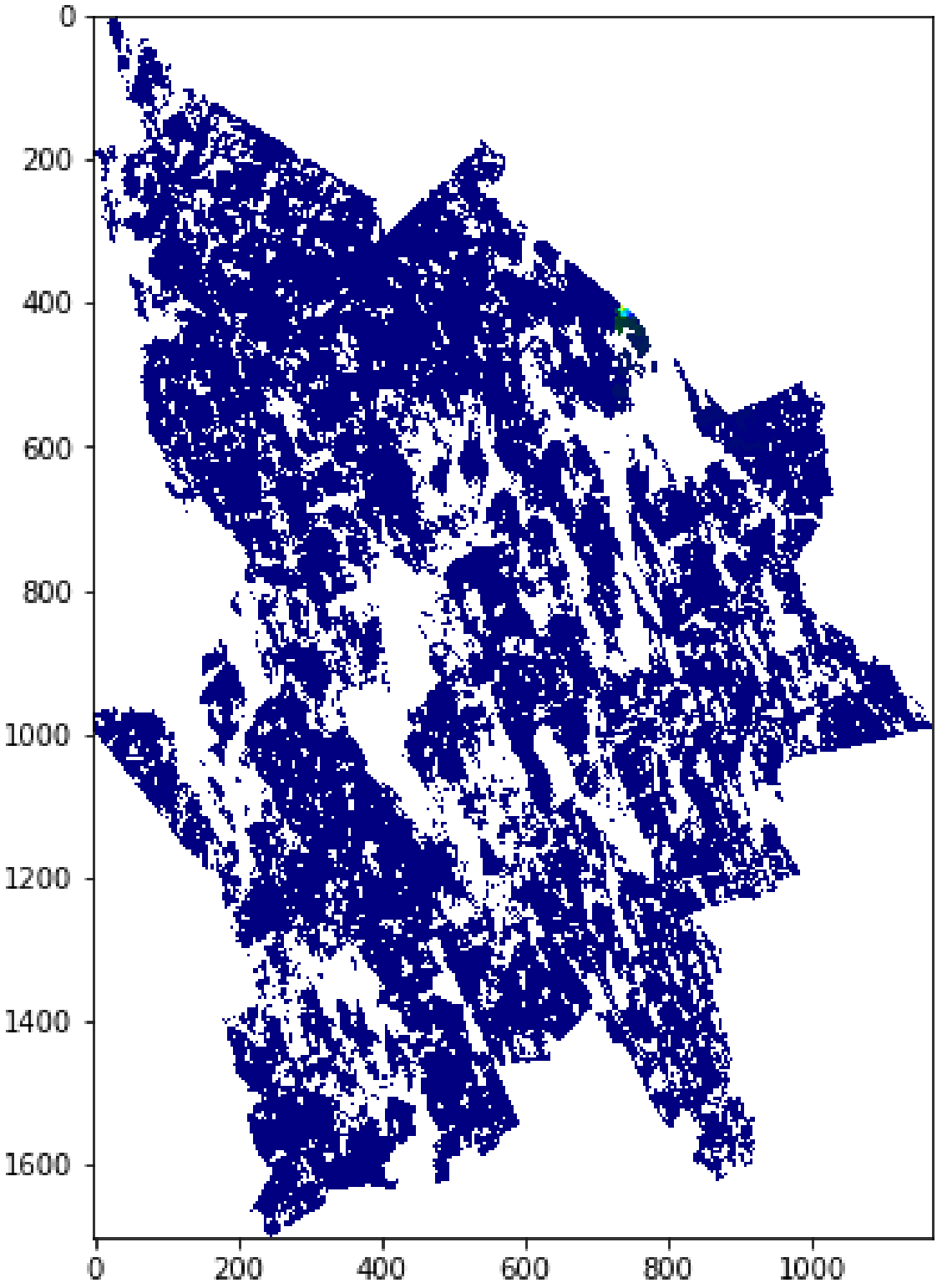

Figure 6: Map presenting the physical area of the forest stands. Forest areas in blue.

should be managed so that in the large scale all the forestry objectives related to financial revenues and biodiversity values would be as desired. (Triviño et al.: 2017)

The data set is based on real values from Finnish Forestry Center and extensive empirical simulation models of forest growth under different conditions. The actual values in the data set are results of computational simulations simulating 50-year growth of all the forest stands under different management regimes. For details and exact sources it is recommended to turn to Mönkkönen et al. (2014).

This forest management problem has already been mathematically formulated and solved by Triviño et al. (2017), so the single objective optimal values and some Pareto optimal solutions and their alternative optimal management strategies are already known. The purpose of using this data and problem in this is not to solve an yet unsolvable problem but to demonstrate and verify the new cluster based approach presented in Chapter 4. The objectives of the problem are described in the following Section 5.1.1 including the already attained optimal results and the different management strategies are shortly explained in Section 5.1.2. The actual format of data and its features are presented in Section 5.1.3.

5.1.1 Objectives

The four forestry objectives of the management problem are 1) financial revenue of extracted timber (*Revenue*), 2) amount of carbon stored into the forest stands (*Carbon*), 3) amount of deadwood in the forest stands (*Deadwood*), and 4) the suitability of habitat for different species in the stands (*HA*). All these objects are to be maximized and their potential maximum values can also be found in Triviño et al. (2017).

Timber revenue objective means the economic value of extracted timber, calculated using eight timber assortments. The values are calculated as net present value (NPV), and in the previous studies the maximum revenue of the entire area considered, i.e. ideal value, has been 250 M Eur, on average 5800 Eur/ha. For more details how revenue values have been estimated one may turn to Mönkkönen et al. (2014).

Carbon storage values imply the average amount of carbon stored in living wood, dead wood, extracted timber and the residuals left after harvesting. The potential maximum capacity for deadwood has been calculated to be $4459 * 10^3$ MgC, 103 MgC/ha on average. For more details see Triviño et al. (2015).

The amount of deadwood in the forest is an indicator of forest biodiversity. The values presented in the data are average values of 50 year period. The potential maximum capacity of deadwood calculated in the earlier studies is 218 150 m^3, on average 5.1 m^3/ha. More details can be found from Triviño et al. (2017).

Species habitat availability describes a combined habitat suitability index for six different vertebrate species. The details and description how the suitability values are calculated one may turn to Mönkkönen et al. (2014)). The maximum potential for this index is 20 211 (no unit), which on average means 0.47/ha.

5.1.2 Management regimes

The seven different available forest management regimes are: the recommended regime for Finnish private forestry (BAU), recommended regime with modified green tree retention (GTR30), recommended regime with postponed final harvesting (two different postponings, EXT10 & EXT30), recommended regime but without thinnings (two different modifications, NTLR & NTSR) and a set aside option (SA), where the forest is left without managing. All these are existing forest management strategies already used for different purposes, for example, mitigating biodiversity losses in otherwise commercially managed forests, as explained by Mönkkönen et al. (2014). The management regimes are presented in Table 1.

In the forest management simulations EXT10, EXT30 and GTR30 regimes were not allowed on forest stands that did not reach maturity during the 50 year simulation period. No-thinning regimes, NTLR and NTSR, were not allowed on stands that would not be thinned under the BAU regime either. Set aside (SA) regime was not allowed on stands that did not have any operations in the BAU regime. In practice all

Management regime	Acronym	Description
Business as usual	BAU	Recommended management: rotation length 80 years; site preparation, planting or seeding trees, 1-3 thinnings, final harvest with green tree retention level 5 trees/ha
Set aside	SA	No management
Extended rotation (10 yrs)	EXT10	BAU with postponed final harvesting by 10 yrs; rotation length 90 years
Extended rotation (30 yrs)	EXT30	BAU with postponed final harvesting by >30 yrs; rotation length 115 years
Green tree retention	GTR30	BAU with 30 green trees retained/ha at final harvest; rotation length 80 years
No thinnings (final harvest threshold values as in BAU	NTLR	Otherwise BAU regime but no thinnings applied; therefore forests grow more slowly and final harvest is delayed; rotation length 86 years
No thinnings (minimum final harvest threshold values)	NTSR	Otherwise BAU regime but no thinnings; final harvest criteria adjusted so that rotations do not prolong; rotation length 77 years

Table 1: The seven different forest management regimes, adapted from Mönkkönen et al. (2014)

these constraints meant cases when managing the forest stand according to these not allowed regimes would not differ at all when compared to BAU regime.

5.1.3 Data format

The Boreal forest data is available as four matrices, named *"Timber revenues"*, *"Carbon storage"*, *"Deadwood volume"* and *"Combined HA"*. Each of these presents values of one objective function. In each matrix, i-th row presents a single forest stand, and j-th column a management regime. This way each cell c_{ij} presents the objective function value of the forest stand i if management regime j is chosen for that stand. The data in all the matrices is homogeneous, the only difference being the numerical values and their magnitudes in different objectives.

As explained in Section 5.1.2 there were cases where some management regimes were not allowed for some stands. In matrices these cases are marked as Nan-values. It is also notable, that these Nan-values are always in the same stands i and management regimes j in all four matrices, so that the data can be said to be of good quality. In total 56.6% of lines contained Nan-values and 22.3% of all the values were Nan-values.

In total the data makes 29 666 variables – each forest stand being one variable. All of them have a maximum of 7 discrete values, i.e. management regime options, for four different objectives. Mathematically this means that the dimension of the decision space is 29 666 and of the product space 4. This size of the decision space is also the main cause of the computational cost in this problem. These characteristics, homogeneous large data and discrete variables, make this problem an appealing case study for the suggested cluster based surrogate.

5.2 Optimizing Single Objective

The objectives for the aforementioned Boreal forest management problem are:

maximize Timber Revenue,
maximize Carbon Storage,
maximize Deadwood value,
maximize Habitat Availability index.

All of these objectives are to be maximized and they can be formulated as:

$$\begin{aligned} maximize & \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \\ s.t. & \sum_{i=1}^{n} x_{ij} = 1 \\ & x_{ij} \in \{0,1\} \end{aligned} \tag{5.1}$$

where $i \in \{1,2,...,n\}$ denotes a forest stand (number of stands n = 29 666), and $j \in \{1,2,...,m\}$ denotes a management regime with seven alternatives (m = 7). For each stand i and management regime j, x_{ij} has value 1, if regime j is applied to stand i and otherwise 0. The basic formulation for all the single objective problems being the same, the differences between the problems are in the c_{ij} values.

All of these single objective optimization problems were first solved using all available data i.e. without surrogate. The solving was executed with Pyomo-optimization framework by Hart, Watson, and Woodruff (2011) and Hart et al. (2017), and by using Jupyter Notebooks [2] and CPLEX optimization solver [3]. Solving was done on four core Intel i7 CPU with 8 GB of RAM. The computer was running Ubuntu 16.04 LTS. [4]

2. https://jupyter.org/
3. https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer
4. Codes available in https://github.com/josejuhani/gradu-code

The single objective optimizations of Timber revenue, Carbon storage, Deadwood Volume and Combined HA took 2 min 52 sec, 2 min 25 sec, 2 min 22 sec and 2 min 14 sec of wall time, respectively. The results of these optimizations and comparison with the values presented in Triviño et al. (2017) are presented in Table 2

Objective	Unit	Value in article	Value attained
Timber Revenue	M Eur	250	250
Carbon Storage	MgC	4459×10^3	4449×10^3
Deadwood Volume	m^3	218 150	218 153
Habitat Availability index	(No unit)	20 211	20 225

Table 2: Values of single objective optimizations and values presented in Triviño et al. (2017)

As can be seen from Table 2, there are some minor differences between values presented in Triviño et al. (2017) and values attained by optimizing the same data. The differences are still relatively so small that we can assume them to be caused only by numerical approximation errors or being typos.

The real life interpretation of these results i.e. how different management regimes are chosen for each optimal result, are presented in Table 3.

Objective	BAU	SA	EXT10	EXT30	GTR30	NTSR	NTLR
Timber Revenue	12 501	29	2 919	97	2 192	9 220	2 708
Carbon Storage	2 541	22 101	174	168	77	32	4 573
Deadwood Volume	801	21 223	117	60	1 921	686	4 858
Habitat Availability	758	19 770	200	159	1 198	1 408	6 173

Table 3: Number of chosen management regimes for each single objective optimal solutions

From Table 3 it can be clearly seen that Business As Usual, the recommended forest management regime, is the best for maximizing financial revenue acquired from the forest. The other extreme is Set Aside option, which is useful in maximizing all the other objectives. Five other management regimes lie somewhere between these two.

These differences also imply how Timber revenue objective may pose the greatest trade-offs with the other objectives in the multiobjective optimization. It could be then assumed that when optimizing all these objectives simultaneously, there would be less BAU or SA managements and more of the alternative less extreme managing options chosen. In addition many different alternative combinations of managements could be considered for attaining the same desired result.

5.3 Building a Cluster Based Surrogate

The data at hand is small enough so that it actually could be possible to solve all the scalarizing problems of the Synchronous NIMBUS method within the resources of the given machinery: solving all the four scalarizing problems takes about 10 minutes. The usage of a surrogate is then not a necessity for this problem, but it can be used in order to reduce the computational burden of solving. Especially in an interactive process this also shortens the calculation and waiting times during the iterative decision making process and thus makes the interaction more seamless. However, if the data was significantly larger, the benefits of using this cluster based surrogate approach would become much greater.

5.3.1 Preprocessing the data

Before building the cluster based surrogate it was required to pre-process the data. In the data matrices there were many Nan-values representing cases where certain management regimes were not available for some forest stand. As described in Section 5.1.2, the reason for the management regimes being not available was that in reality their results did not differ from the BAU regime in reality. From these followed that it was safe to use the corresponding BAU values of each forest stand to replace its Nan-values.

The values for different objective functions were also on very different scales. The values of Timber Revenue objective ranged from -4231.84 to 175966.0 and of Combined Habitat from 0.0 to 11.3966. The ranges of the other two objectives lied somewhere

in between. Because of these differences, the values needed to be normalized for the clustering. All the cells in each matrix were scaled between 0 and 1 i.e. the greatest value in matrix was assigned to be 1, the smallest 0 and all the other values scaled to be relative to these.

5.3.2 Choosing the similarity metric

In this Boreal Forest data each variable i had 7 different values c_{ij} for each of the four objectives. In total there were then 28 numerical values for each variable, that could be used for clustering the variables. The clustering was then supposed to be done for 29 666 variables in a space which dimension was 28.

As the goal of clustering was to group similar type of variables to same cluster, the similarity had to be defined: Each variable being a forest stand, we could define two of them being similar when they produce similar kind of results with the same management regimes. However, because the forest stands are all of different sizes, the similarity of results was better to be measured by using the relations between different management regimens. If that was measured using the absolute similarity of values, there may not be any two forest stands resembling on another enough. The relation-based similarity of variables was then captured the best by using cosine distance as similarity metric. The cosine distance metric caused the clusters to be of elongated shape, as can be seen in Figure 7.

5.3.3 Choosing the clustering method

In addition to being able to choose the similarity metric, we did not know what kind of clustering method would be naturally the most suitable for the data. Still, as the purpose of clustering in this was just producing a good enough surrogate model, the simple and traditional K-means algorithm was chosen.

An algorithm almost equal to K-means would have been K-medoids by Kaufmann and Rousseeuw (1987). While being similarly iterative than K-means, the centers for the clusters are assigned differently in K-medoids algorithm. When in K-means the centers

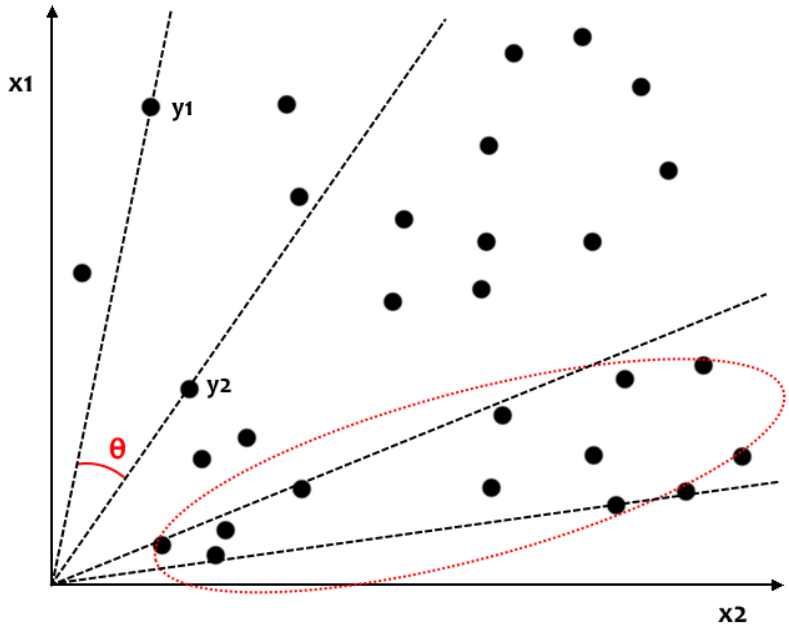

Figure 7: Cosine distance θ between points y1 and y2 in two dimensions. The shape of cluster caused by the cosine metric marked on red.

of clusters are just abstract points in the exact center of the cluster , in the K-medoids each center has to be an existing data point. This naturally affects also how the other points are assigned to clusters in each iteration.

Usually K-medoids is used instead of K-means for discrete data, as abstract center points would not make any sense with discrete data. However, because cosine distance was decided to be used as a similarity metric in the clustering, the centers generated by K-medoids would not represent each cluster in the way it would be desired. As the cosine metric assigns data points into clusters according to the “angle” between the points, the K-medoids center of each cluster would not be presenting the average sized data point, as is needed for the optimization phase. This problem is further elaborated in Figure 8.

Because of this it was decided to use K-means algorithm instead of K-medoids, and selecting the variable closest to the euclidean average of the cluster as the most representative one. This is also shown in Figure 8.

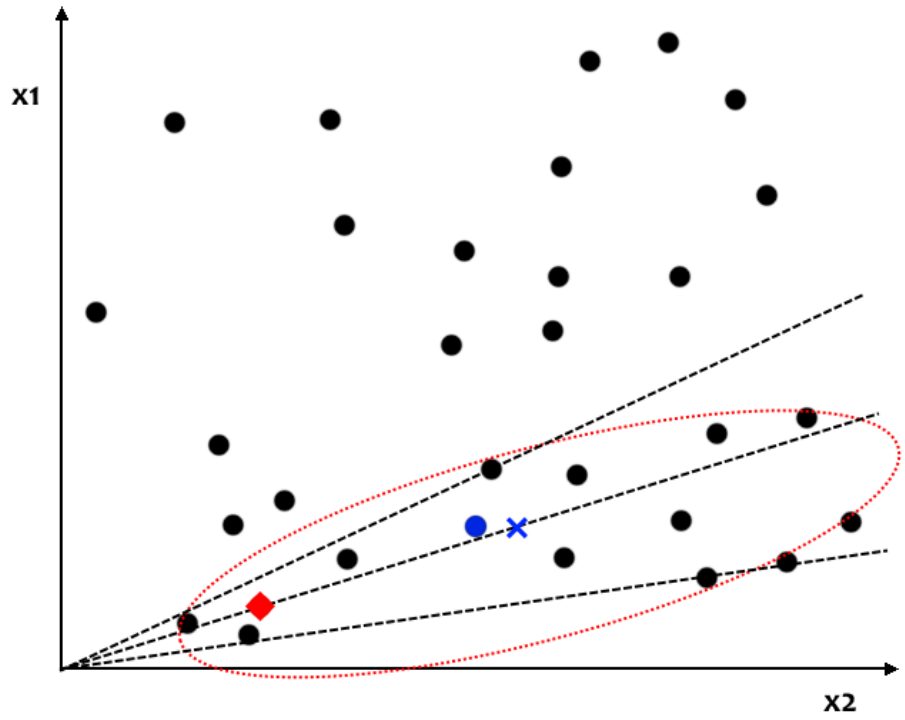

Figure 8: An imaginary cluster marked with red dotted ellipse. If the cluster had been formed using cosine distance and K-medoids algorithm the center would be the point marked with red diamond. The euclidean center of the cluster is marked with blue cross and the point closest to it with blue. In this case the center point of K-medoids does not represent the entire cluster as desired.

The K-means algorithm requires the initial cluster centers to be used as starting points for the formed clusters. As described in Section 4.1.1 these initial centers affect greatly the clustering results. From the surrogate forming point of view, these initial cluster centers may also be seen as the initial sampling: the third step in the surrogate forming process (Section 3.2.1). In this study this selection of the initial cluster centers was chosen to be done by taking random samples from the data.

5.3.4 Choosing the number of clusters

For K-means algorithm the number of clusters is needed to be decided beforehand. In this case we could not guess what that number really should be due to the lack of domain knowledge. In Section 4.1.2, it was stated that the more clusters there are the

closer the optimization result are to the real optimum. In this case it was possible to generate the number of different clusterings and compare them empirically. K-means algorithm being sensitive for the initial clustering of the data, all the clusterings with different number of clusters were repeated ten times with different random initial cluster centers.

Because training a clusterings model to a data set is also a computationally expensive task, it was not sensible to form clusterings for all possible numbers of clusters. It was decided to consider only clusterings for which the forming and solving the scalarizing problems would not take longer than what it would take to solve single interactive iteration without surrogate. As one iteration took about 10 minutes, the upper limit was set empirically to 8300 clusters, for which the time requirement was approximately the given 10 minutes, and the lower limit to 50, for which it took about 2 seconds. The clusterings were then formed with the number of clusters ranging from 50 to 8300, with interval of 50 clusters.

These clusterings were compared by using the sum of intra cluster cosine distances as an evaluation metric. This metric aims to evaluate the validity of the clustering by showing how similar the objects within the same cluster are. The evaluation results are shown in Figure 9.

As can be seen from Figure 9, the sums of intra cluster distances decreased exponentially with the increase in the number of clusters. This decrease in distances meant that when there was more clusters, the features within clusters were more close to each other i.e. more similar.

In the graph there is no clear “knee points” visible, that would imply remarkable change in the validity of clustering. This also meant that the above-mentioned L-method by Salvador and Chan (2004) was not a valid approach for this. Because of this, the only remark we were able to make from the graph was that the more clusters we had the more valid the clustering was. The proper number of clusters, either for traditional clustering or for cluster based surrogate, could not then be decided based on this metric.

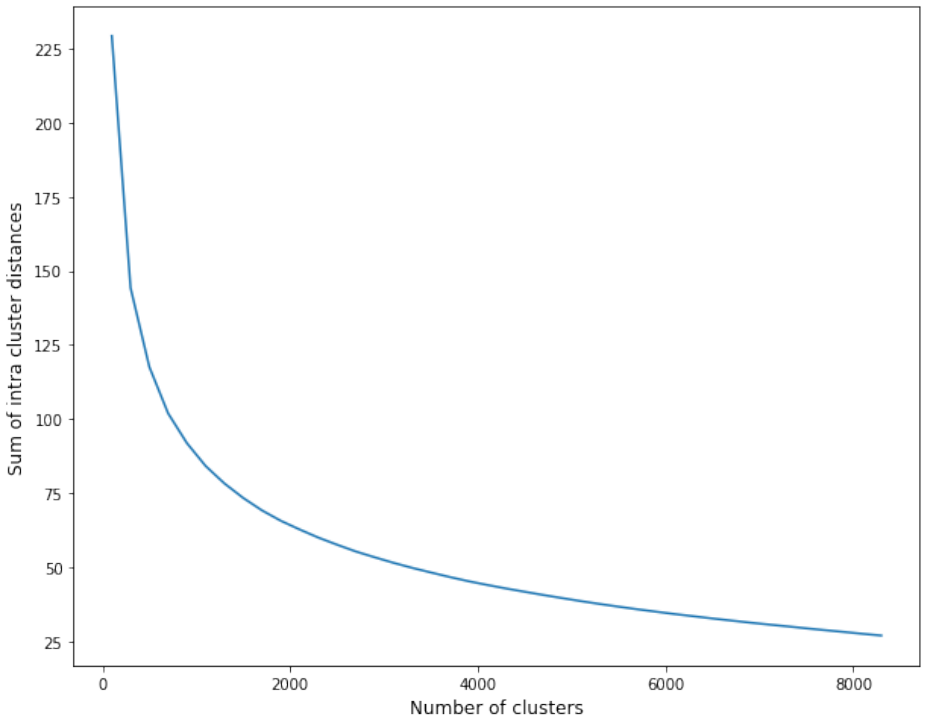

Figure 9: Relationship between the numbers of clusters and average sum of intra cluster distances. Calculated with interval of 200 clusters.

However, the performance of a cluster based surrogate is not necessarily dependent on the quality of the clustering used. In some cases these two may even be in contradiction. The performance of the cluster based surrogate and the effect of the number of clusters could then be measured not by using only clustering indices but by measuring its ability to approximate the data. In all its simplicity, this meant solving the actual optimization problems using the cluster based surrogates. As described earlier in Section 4.1.2, one proxy variable from each cluster was selected to be used in the optimization. In this case the variable closest to the euclidean center of cluster was selected.

The single objective optimizations of all the objectives were done using the same clusterings constructed, presented and evaluated in the previous chapter. As the time required for each optimization run increased greatly with the number of clusters used for the surrogate, the optimizations were executed only for up to 1500 clusters, as the

variations in the results stabilized approximately before that point. These results are presented in Figure 10.

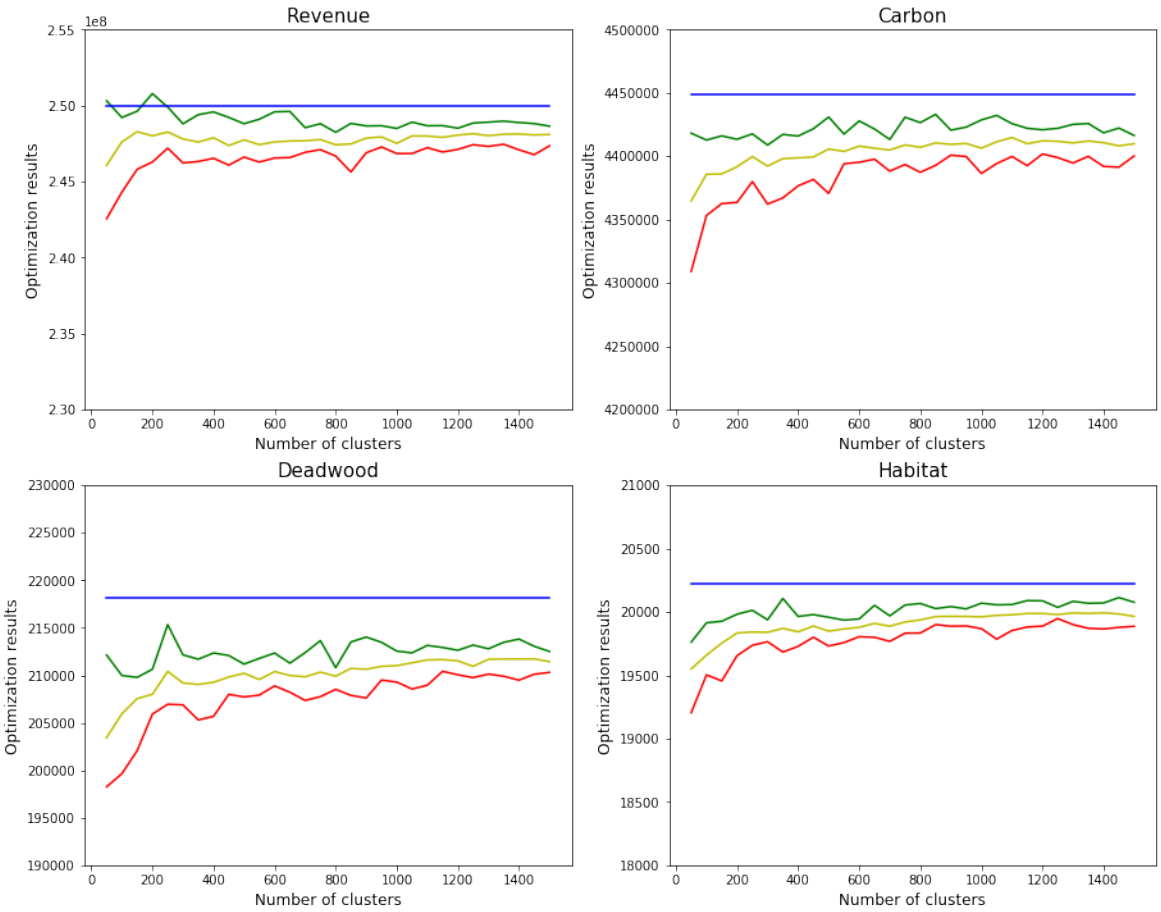

Figure 10: Optimization results for four objectives when using values from the surrogate. Maximum (green), mean (yellow) and minimum (red) values of ten different clusterings with same number of clusters. Blue lines marks the real optimal values for each objective. Calculated with interval of 50 clusters. Scales focused near the optimum of each objective

As can be seen in Figure 10, the optimization results in general improved with the increase in the number of clusters. For the Deadwood Volume objective there are a bit greater variations in the optimization results, but still they were not remarkably worse than the others.

The optimization results were behaving as expected and presented in Section 4.1.2, i.e. the results improved along with the increase in the number of clusters. This also

mean that we were free to choose any clustering we regarded to be accurate enough, or one that would keep the time of an interactive iteration under certain time limit.

In this case, we wanted to choose a clustering that kept the interactive iteration delay times under ten seconds as Miller (1968) suggests for this kind of man-computer interaction. When empirically tested, the number of clusters filling that requirement was 600 clusters: For 600 clusters it took about 2.5 seconds to solve each of the four scalarizing functions, and about ten seconds in total. The time required for forming the surrogate with 600 clusters was about 25 seconds.

As there were ten differently initialized clusterings for each number of clusters, the ten different clusterings with 600 clusters had to be compared to one another in order to choose the actual one to be used. As the different objectives had relatively different optimization results with different clustering initializations, all of them had to be regarded somehow in choosing the right clustering. As the only way to measure the goodness of a single objective optimization result was its relation to the real optimum, this same measure was chosen for measuring the clusterings also. Thus the sum of relative differences of cluster based optimization results to the real optima was selected as the measure of goodness, as proposed in Jin, Husken, and Sendhoff (2003). For each clustering its goodness measure was calculated by:

$$\sum_{i=1}^{4} \frac{z_i - z_i^{\star\star}}{z_i^{\star\star}} \tag{5.2}$$

where i means the index of the objective and z_i the optimal value of the objective i attained by surrogate based optimization. The comparisons of different objectives based on this metric are presented in Figure 11

From Figure 11 it was seen, that the clustering with random initialization number six was the best one according to the measure used. This clustering was selected to be used in the interactive optimizations. The single objective properties of this clustering are presented in the following Section 5.4.

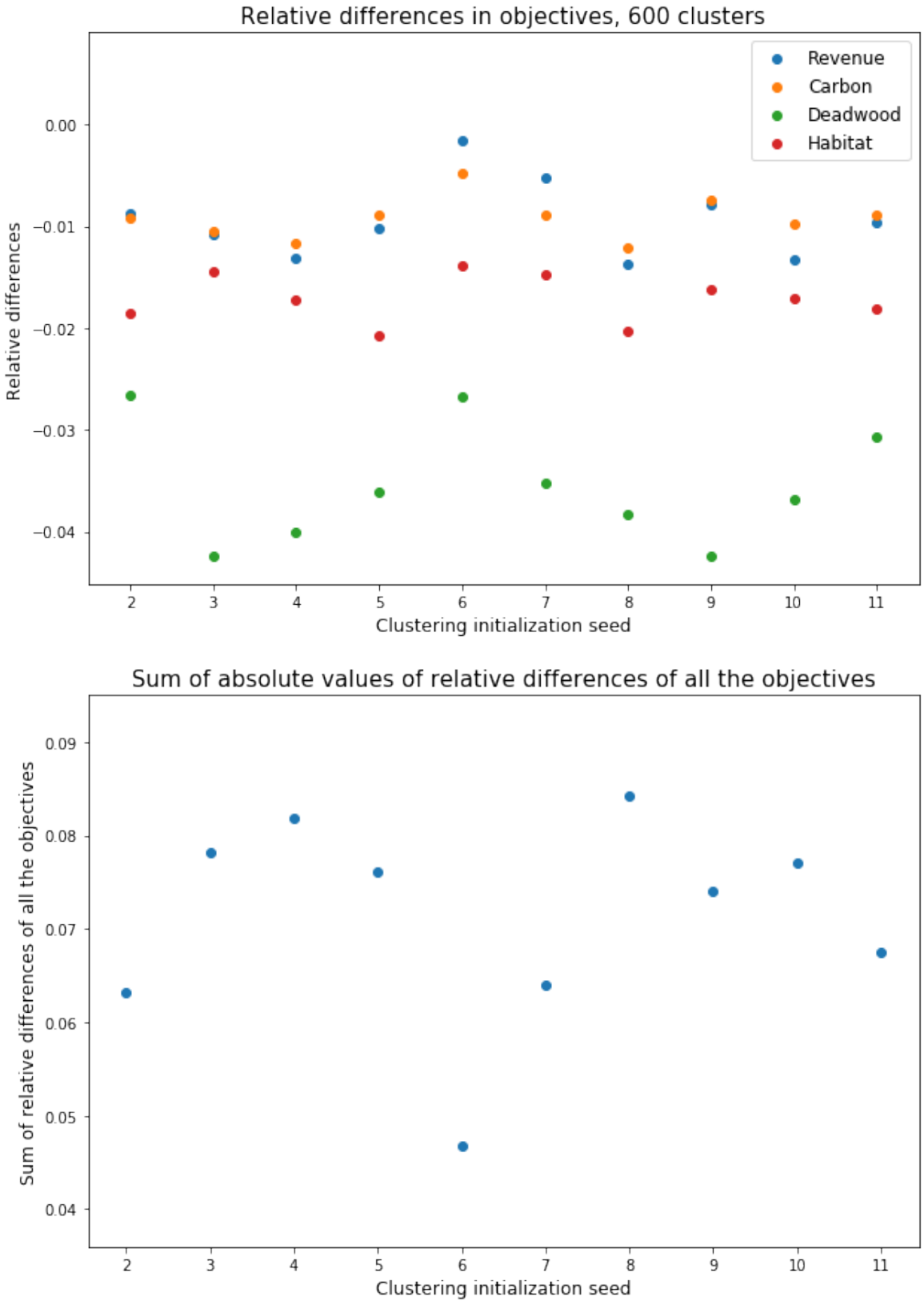

Figure 11: Upper picture: The relative differences of all the objectives and different cluster initializations having 600 clusters. Lower picture: The sums of relative differences for all the four objectives.

5.4 Optimizing Single Objective using Surrogates

For the chosen clustering, that consisted of 600 clusters, the ideal and nadir values according to the proxy variables are compared to the real ideal and nadir values in Figure 12. The results are presented in parallel coordinates, so that all the objectives are scaled between 0 to 1, where 0:s are the real nadirs of the objectives and 1:s the real ideals.

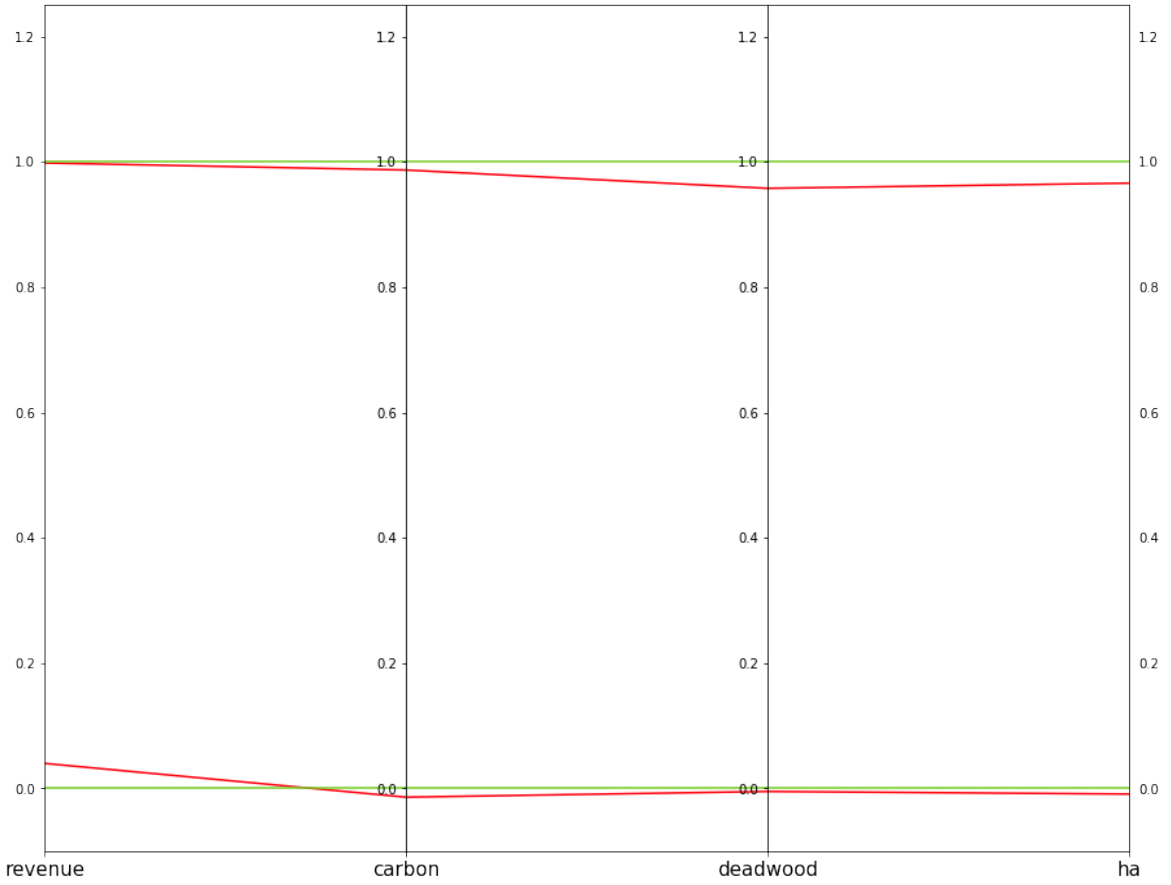

Figure 12: The ideal and nadir values of cluster based surrogate with 600 cluster. The surrogate based results are in red and the real ideal and nadir values in green.

The surrogate based ideal values for Timber revenue, Carbon storage, Deadwood volume and Habitat suitability were 249 596 432, 4 427 946, 212 327, 19 945, and measured by the relative difference to the real ideal values these are -0.1%, -0.5%, -2.7% and -1.3%. The nadirs of the corresponding objectives were 39 679 664, 2 808

943, 79 554 and 11 915, and the differences to real nadirs 3.5%, -0.5%, -0.3% and -0.4%. These can be seen as measures for the approximation errors inevitably related to the use of surrogates and in this case these errors were not significantly large.

Figure 12 presents clearly how objectives behave differently after the clustering. The differences in ideal values are not very large for any of the objectives, the Deadwood Volume being the greatest anomaly. The same notion can be said from the nadirs also, but because the calculations of nadirs are based on payoff tables, and may not be very accurate even with the original data, comparisons between them has to be taken only as rough approximations.

The cluster based surrogate bundles together multiple variables, in this case multiple forest stands. After the optimization the results can be interpreted so that for each cluster the same forest management regime is implemented for every forest stand in the cluster. The only exceptions are forest stands in the cluster for which it is not possible to use the chosen management regime. In those cases the "Business As Usual" regime can be chosen as it certainly is available for every forest stand. These results of the surrogate based single objective optimizations and comparisons to original optimal solutions are presented in Table 4.

	Objective							
	Timber Revenue		Carbon storage		Deadwood Volume		Habitat suitability	
BAU	13 933	+1 432	2 147	-394	555	-246	715	-43
SA	17	-12	21 739	-362	21 512	+289	19 225	-545
EXT10	2 177	-742	0	-174	0	-117	22	-178
EXT30	11	-86	2 372	+2 204	2 283	+2 223	2 258	+2 099
GTR30	1 135	-1 057	0	-77	1 410	-511	1 087	-111
NTSR	10 082	+862	0	-32	298	-388	1 297	-111
NTLR	2 311	-397	3 408	-1 165	3 608	-1 250	5 061	-1 112

Table 4: Number of chosen management regimes for each single objective optimal solution when using cluster based surrogate and difference to the real optimal solutions presented in Table 3

From Table 4 it can be seen that the surrogate seems to lead emphasizing EXT30 management regime. The exception to that is optimizing Timber Revenue, in which case the BAU regime is chosen even more frequently than originally. As already suggested in Section 5.4, the EXT30 regime, that is a compromise between BAU and SA regimes, became chosen more often than the extreme regimes.

5.5 Interactive Multiobjective Optimization using Surrogates

5.5.1 Implementation

The implementation of any cluster based surrogate and the interactive optimization method are dependent of the actual case at hand. In Section 5.3 the required similarity metrics and optimization algorithms were justified, and the actual implementation of these is described in this section.

The clustering of the data was implemented and verified using Python libraries and Jupyter Notebooks. For solving the multiobjective problem IND-NIMBUS by Ojalehto, Miettinen, and Laukkanen (2014), an implementation of the synchronous NIMBUS method, was used. The single objective sub-problems were modeled using PYOMO (Hart, Watson, and Woodruff: 2011; Hart et al.: 2017) and solved using CPLEX optimizer. [5]

From the implementation point of view, the most essential part in the synchronous NIMBUS approach are the four scalarizing functions: ASF, STOM, GUESS and NIMBUS. They and their exact mathematical formulations have already been presented in Section 2.3, and implementing them with Pyomo framework was quite straightforward. The only challenge with the formulations was that the framework is only capable of handling linear problems, so for instance the maximum function could not be

5. Codes available in `https://github.com/josejuhani/gradu-code`

used. Because of this the original formulation (2.2) had to be replaced with a smooth formulation by Laukkanen et al. (2010), introducing an additional variable y:

$$\begin{aligned} \textit{minimize} \;\; & y + \rho \sum_{i=1}^{k} \frac{f_i(x)}{z_i^{nad} - z_i^{\star\star}} \\ s.t \;\; & x \in S \\ & y \geq \frac{f_i(x) - \bar{z}_i}{z_i^{nad} - z_i^{\star\star}} \;\; i = 1..k \end{aligned} \tag{5.3}$$

Where the variables are as in (2.2). As can be seen, the smooth formulation transforms the maximum function into a set of new constraints. In this case that meant generating four new constraints, one for each objective function. In the original article presenting the smooth formulation of the scalarizing functions, the augmentation term was missing, but is presented here as it was used in this implementation.

From the four scalarizing functions of the synchronous NIMBUS, the GUESS scalarizing function needed to be modified when it was used with classification based reference point. As Miettinen and Mäkelä (2006) describe, for $I^{\diamond}$ objectives the denominator must be replaced with $z_i^{nad} - z_i^{\star\star}$. Also the min-max term considers only the objectives other than possible free class $I^{\diamond}$. The modified GUESS function is then of form:

$$\begin{aligned} \text{minimize} \;\; & \max_{i \notin I^{\diamond}} \left[\frac{f_i(x) - z_i^{nad}}{z_i^{nad} - \bar{z}_i} \right] + \rho \sum_{i=1}^{k} \frac{f_i(x)}{z_i^{nad} - \bar{z}_i} \\ \text{subject to} \;\; & x \in S \end{aligned} \tag{5.4}$$

The surrogate based ideal and nadir values were presented already in Section 5.4 and while the differences between the real and surrogate based ranges were not significantly large, there still came up some differences in the scalarizing results in empirical tests. Because of this the real ideal and nadir values of the problem were chosen to be used in the implementation, as was recommended in Section 4.3.2.

The graphical user interface of IND-NIMBUS is presented in Figure 13. The current Pareto optimal solution is on the left presented as a bar chart, where each bar presents

a single objective function and its end points the nadir and ideal values. In this case, the objectives are to be maximized and so the bars are aligned to right and the shorter the colored bar the better the values. The decision maker can indicate her/his preferences by clicking the bars. Clicking the non-colored part implies that the chosen objective is allowed to impair. Respectively, clicking colored part means that the objective is desired to improve. On the right all the calculated Pareto optimal solutions are presented in the upper and the most interesting ones in the lower panel.

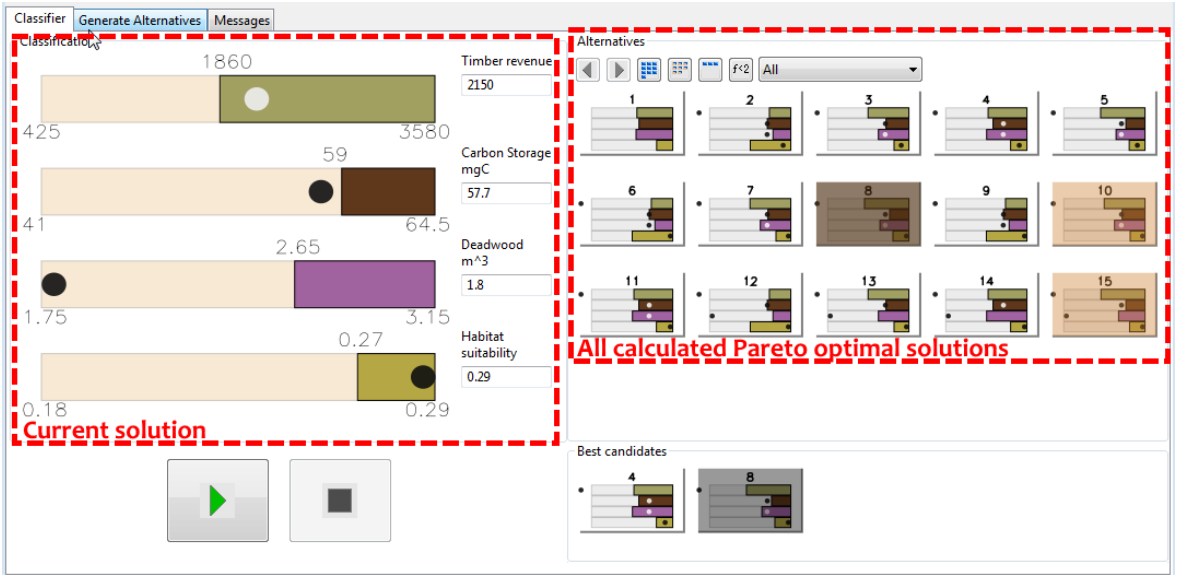

Figure 13: A screenshot of the graphical user interface of the IND-NIMBUS showing interaction with the decision maker.

5.5.2 Relationships of the objectives

In this Boreal Forest optimization case, the shifting of Pareto front that was presented in Section 4.3.1 and in Figure 5 is possible for instance between Deadwood Volume as f_1 and Timber Revenue as f_2 objectives, as their ideal and nadir values are related the same way as presented there. One way to verify this kind of behavior is calculating results for all four different scalarizing of the synchronous NIMBUS using different reference points. The results of the scalarizings when one objective at a time was classified to be improved and the others were classified to be free are presented in Table 5.

	Bounds of the surrogate			
	Revenue	Carbon	Deadwood	Habitat
Ideal	249 596 433	4 427 946	212 328	19 946
Nadir	39 679 664	2 808 943	79 554	11 915

	Objective to optimize			
	Timber Revenue			
	Revenue	Carbon	Deadwood	Habitat
ASF	249 578 305	2 828 548	79 973	11 974
STOM	249 596 433	2 808 943	79 554	11 915
GUESS	249 415 335	2 784 933	72 219	11 182
NIMBUS	249 596 433	2 808 943	79 554	11 915
	Carbon Storage			
	Revenue	Carbon	Deadwood	Habitat
ASF	28 036 750	4 427 788	200 868	18 108
STOM	8 129 204	4 427 946	202 350	18 221
GUESS	8 152 708	4 427 918	201 158	18 215
NIMBUS	88 837 788	4 427 788	202 044	18 124
	Deadwood Volume			
	Revenue	Carbon	Deadwood	Habitat
ASF	30 184 020	4 214 348	212 317	18 915
STOM	31 658 632	4 207 590	212 328	19 036
GUESS	30 073 050	4 214 416	212 317	18 911
NIMBUS	113 414 218	4 191 144	212 317	19 011
	Habitat Suitability			
	Revenue	Carbon	Deadwood	Habitat
ASF	69 059 478	3 921 094	204 333	19 945
STOM	67 378 315	3 931 616	206 008	19 946
GUESS	68 882 190	3 921 091	204 333	19 945
NIMBUS	144 157 068	3 913 622	205 682	19 945

Table 5: Results of different scalarizing functions when the reference point set to ideal value of one objective and the other objectives are free. The object wise best values are highlighted with red and the worst ones with blue.

From Table 5 it can be seen that it is possible to attain all the ideal values of the single objective problems and nadir values even smaller that were estimated before. As the nadir values are based only on pay-off table estimations we can safely assume that the scalarizing functions are working properly and that it is possible to attain results from any part of the Pareto front. It can also be noticed that the smallest values of all the other objectives are attained while optimizing the Timber Revenue. On the other hand, the values of Timber Revenue are significantly smaller when optimizing any other objective. This indicates that the Revenue objective has the greatest conflict with the other three objectives.

However, Table 5 does not reveal anything about the behavior of the surrogate based Pareto front; All the objectives are behaving as expected and there are no indications about possible "twists" in the front. In order to explore the behavior of the surrogate based Pareto front, some surrogate based Pareto optimal solutions were used as reference points to ASF in the original data and problem. Since any results from Table 5 would have been used, the NIMBUS results of the table were chosen for this purpose. These points and the corresponding results in the original product space are presented in Table 6. For comparison purposes there are also presented the results of the surrogate based optimization if decision values y_{kj} of the proxy variables are mapped to all the variables in each cluster k, as described in Section 4.3.2.

	Objective to optimize			
	Timber Revenue			
	Revenue	Carbon	Deadwood	Habitat
Surrogate front	249 578 348	2 823 113	81 765	12 043
Mapped front	245 986 463	2 838 939	85 283	11 996
Real front	249 956 786	2 825 912	82 020	12 057
	Carbon Storage			
	Revenue	Carbon	Deadwood	Habitat
Surrogate front	87 386 441	4 427 793	202 829	18 178
Mapped front	87 704 387	4 399 341	204 091	18 104
Real front	89 982 468	4 446 963	204 467	18 276
	Deadwood Volume			
	Revenue	Carbon	Deadwood	Habitat
Surrogate front	113 414 218	4 191 144	212 317	19 011
Mapped front	112 105 435	4 185 685	213 158	18 956
Real front	121 167 672	4 248 398	217 199	19 303
	Habitat Suitability			
	Revenue	Carbon	Deadwood	Habitat
Surrogate front	144 157 068	3 913 622	205 682	19 945
Mapped front	141 649 379	3 942 044	208 390	19 811
Real front	150 256 353	3 958 661	209 523	20 175

Table 6: Surrogate-based and mapping-based extreme solutions and corresponding really Pareto optimal solutions. Objective wise greatest values in red and smallest in blue.

From Table 6 it can be seen that the all the Pareto optimal values of the surrogate are less than those of the original Pareto front. This leads to assume that the surrogate based Pareto front is located below the original front i.e. the entire surrogate front is just shifted “down” from the position of the original Pareto front. Naturally there still may be some areas where the front is twisted as in Figure 5, but this would have

required exploring the entire Pareto front extensively and that could not be afforded in this study.

The presented mapping based results are neither systematically greater nor less than the surrogate based solutions. As the cluster based surrogate is based on averaging values of multiple variables this is not a surprise. In general these different solutions are resembling one another quite a lot and so they also could be used as implementable preferred solutions, as suggested in Section 4.3.2.

When the surrogate based solutions could be too high i.e. better than the real Pareto optimal solutions, the mapping based results could never be. This is because the mapping is using all the real variables, and so it is simply not possible to reach unattainable solutions. However, in the Timber Revenue solution of Table 6 the mapping based solution has some greater values than the original Pareto optimal solution and so they are non-dominating. This indicates that the solution attained using the mapping may also be actually Pareto optimal, but on different part of the Pareto front.

5.6 Solving with the Decision Maker

The interactive solving process of the multiobjective problem was done using the parameters and software implementation described in the previous sections. The decision maker of the process was a person that had real experience from researching and implementing forest management regimes. From the wish of the decision maker, all the objective values during the process were scaled to indicate values per hectare. As the total area of the forest stands was 68 700 hectares, all the objective values were divided by that number.

To start the solution process the decision maker was shown a neutral compromise solution with values [2710.0, 58.3, 2.76, 0.26] i.e. a solution where all the objectives were balanced. That solution was obtained by using the midpoints between ideal and nadir values as a reference point.

Iter	Issue	Timber revenue eur	Carbon Storage mgC	Deadwood volume m^3	Habitat suitability
	Ideal	3640.0	64.8	3.18	0.29
	Nadir	450.0	41.2	1.16	0.17
1	Init. Sol.	2710.0	58.3	2.76	0.26
2	Cur. Sol.	2710.0	58.3	2.76	0.26
	Classif	$I^{\geq=2070.0}$	$I^{\leq=59.2}$	$I^{\geq=2.19}$	$I^{\leq=0.28}$
		2070.0	60.4	3.02	0.28
		2180.0	60.0	2.92	0.28
		2250.0	59.9	2.92	0.28
		2150.0	60.1	2.91	0.28
3	Cur. Sol.	2070.0	60.4	3.02	0.28
	Classif	$I^{\leq=2500.0}$	$I^{\geq=59.9}$	$I^{\geq=2.19}$	$I^{\geq=0.28}$
		2280.0	59.9	2.99	0.28
		2420.0	59.3	2.83	0.27
4	Cur. Sol.	2420.0	59.3	2.83	0.27
	Classif	$I^{\geq=2400.0}$	$I^{\leq=59.5}$	$I^{\geq=2.81}$	$I^{\leq=0.28}$
		2380.0	59.4	2.87	0.28
5	Cur. Sol.	2380.0	59.4	2.87	0.28
	Classif	$I^{<}$	$I^{\diamond}$	$I^{\diamond}$	$I^{\diamond}$
		3630.0	41.2	1.16	0.17
		3630.0	41.8	1.53	0.19
	Final Sol.	**2380.0**	**59.4**	**2.87**	**0.28**

Table 7: The interactive solution process with the decision maker illustrated. Values presented with the accuracy of three significant figures.

Starting from that [2710.0, 58.3, 2.76, 0.26] solution DM wanted to see in the second iteration how the solutions would change if Timber Revenue and Deadwood Volume were allowed to impair until 2070.0 and 2.19, and the Carbon Storage and Habitat Suitability were desired to be improved until 59.2 and 0.28. Based on those preferences four new alternative solutions were produced as shown in Table 7.

From the generated four solutions the decision maker chose the second one [2070.0, 60.4, 3.02, 0.28] for further examination. Using this as the current solution for the third iteration he wanted to increase Timber Revenue to 2500.0. For all the other objectives the preference levels were kept the same than on the previous iteration. This time DM wanted to see only two new solutions, so only ASF and NIMBUS scalarizings were selected for this iteration.

The third iteration produced two new solution alternatives, as was desired. From these two DM was a bit more satisfied with the latter solution [2420.0 59.3 2.83 0.27], because it generated more Timber Revenue than the former one. DM also stated that he is not really interested about Deadwood Volume and Habitat Suitability values, because differences in their objective values were quite small in this case. For the fourth iteration the decision maker wanted to attain one more solution that would be a compromise between those two. Starting from the latter solution [2420.0 59.3 2.83 0.27] he set all the references to be somewhere between the two solution alternatives generated during the previous iteration: Timber Revenue was set to impair until 2400.0 and Deadwood Volume until 2.81. Carbon storage was set to be improved until 59.5 and Habitat Suitability until 0.28.

With the generated solution [2380.0 59.4 2.87 0.28], DM was satisfied. To his mind, Timber Revenue was nicely above the average, Carbon Storage was quite high and the ecological aspects were good enough. At this point the decision maker chose this solution to be the best one.

Finally, the decision maker wanted to see what happens if only Timber Revenue is maximized and the other objectives are allowed to change freely. For this purpose only ASF and NIMBUS scalarizing were selected. As expected, the Revenue values

almost attained their ideal value and the others impaired all the way to their nadir values. It was also noticed that while the two scalarizings produced almost similar outcomes there were some minor differences in Timber Revenue that caused clear differences in other objectives. The actual Timber Revenue values of the two final solutions were 3632.9 and 3626.4, and because the values in Table 7 are presented with accuracy of three significant figures the differences cannot be seen in it.

After the interactive solution process the chosen preferred solution [2380.0, 59.4, 2.87, 0.28] was projected to the real Pareto front, as described in Section 4.3.2. The projection was done using ASF scalarizing function and it took 12 minutes to solve. The attained real Pareto optimal solution was [2410.98, 59.58, 2.92, 0.28], so every objective became improved when compared to the surrogate based solution. The relative differences between these two solutions to the real solutions were -1.3%, -0.3%, -1.7% and 0.0%. The decision maker was satisfied with these results also, as all the changes from the surrogate to the real results were positive and relatively small.

The decision maker of the interactive process was satisfied with the smoothness of the interactive process and how there were only short waiting times, especially if compared to solving without the surrogate. He was also happy with the results of the process, both the surrogate based results and the final really Pareto optimal ones and how the relative errors were only minor.

5.7 Analysis of the Case Study

The presented Boreal Forest case study included a data set which was well suited for the cluster based surrogate approach. The characteristics of the data – discreteness, large number of similar variables and an integer linear programming problem – proved that with these requirements this approach is able to provide meaningful multiobjective optimization results. In short, the cluster based surrogate approach was proved to be functional.

As the original results of the Boreal Forest problem were already known, it was possible to compare the results attained through the cluster based surrogate to the original results. This enabled us to prove some hypotheses presented in Chapter 4, such as that the accuracy of the surrogate increases with the increase in the number of clusters. We were also able to verify that the finally chosen clustering with 600 clusters caused the four single objective optimization results to differentiate from the real results only by -0.1%, -0.5%, -2.7% and -1.3%. After the final interactive optimization process and projecting its results to the original Pareto front the differences were -1.3%, -0.3%, -1.7% and 0.0% for that point. Any of these differences was not very large and if larger number of clusters had been chosen, the differences would have been even smaller but the interactive iteration times longer.

Based on the executed case study, the cluster based surrogate approach is especially suitable for forest management problems of the given type. As there is a great amount of forests and also forest management planning in Finland, the developed surrogate and method may prove useful in practice, even though its applicability would otherwise be quite narrow.

5.7.1 Extensions to the Boreal Forest problem

As already noticed in Section 5.2 the Business As Usual and Set Aside managing options are quite opposite to one another, and the other five regimes produce results somewhere between these two. Based on this, it would be interesting to study what kind of trade-offs would be required for creating different combinations of

management regimes for almost similar optimization results. Such comparisons would reveal even more desirable management options than what is attainable with only direct optimization procedure.

At its current set-up the Boreal Forest problem considers an area of the size of about one municipality. Thus the size of the problem could be easily extended by taking into account larger forest areas. This would also be a desirable research question for the future, since the advantages of cluster based surrogate are exposed the best when the sizes of problems are larger. Solving a similar but larger problem could also reveal possible issues related to the usage of cluster based surrogate.

From the practical point of view the Boreal Forest problem could also be extended with a few more objectives. For instance, forest owners could be interested about having all of their forest stands handled using the same management regime or have certain proportion of forest stands handled using certain management regime. The total number of different management regimes could also be set as an objective.

In the clustering phase of the Boreal Forest data, the spatial distances between different forest stands could be taken to account. This way it would be possible to handle stands close to one another with the same management regime. From the practical perspective this would simplify actual forest management and monitoring.

6 Discussion

In this chapter different aspects of the cluster based surrogate are discussed. The cluster based surrogate presented in this thesis can be compared both to dimension reduction techniques and to other surrogate models. In both comparisons, the most important aspect to consider is the discreteness of the decision space, which is the main strength of the cluster based surrogate. These comparisons to dimension reduction are presented in Section 6.1 and to traditional surrogate models in Section 6.2. After these, the possibility of using different weighting methods is discussed in Section 6.3 and using other types of clusterings in Section 6.4. Guidelines for choosing the number of clusters are discussed in Section 6.5. Differences between using real ideal and nadir values or surrogate based ones are enlightened in Section 6.6. The means for estimating approximation errors in different objectives are briefly presented in Section 6.7. As the presented cluster based surrogate was studied only with the linear integer problem extending it to linear mixed integer problems is discussed in Section 6.8.

6.1 Comparison to dimension reduction

Traditional statistical dimension reduction techniques have primarily been developed for measurement and real valued data and not for discrete decision spaces. Earlier, techniques like Principal component analysis (PCA) have been used in optimization for reducing the number of objectives, for example by Pozo et al. (2012), but not in the decision space. There also exist PCA for discrete data by Buntine and Jakulin (2004), but combining this kind of techniques with optimization and in decision space has not been tried, or to our knowledge not even discussed earlier. Otherwise, much more sophisticated techniques for reducing discrete spaces do exist , but for combining discrete dimension reduction with optimization this study proposes a simple solution.

In addition to knowing the objective function values of the optimal solution it is of great importance to know every decision value of the solution. If using some PCA

type of dimension reduction that “mixes” together multiple variables without clear borders, mapping the optimal decision variables of the reduced space back to the original decision variables may be challenging. The PCA like dimension reduction may produce optimization results that have good and accurate objective values, but using them in order to know the values for each decision variable may produce challenges.

As a competitor to other dimension reduction techniques clustering is quite a simple and intuitively more understandable option. As one variable is representing many variables of similar type, the reduction of dimension is in a way more controlled. This is the case especially if the clustering or parts of it are done manually or otherwise exploiting domain knowledge by some clearly understandable mean. This controllability also leads to more interpretable optimization results.

When compared to other dimension reduction approaches, the cluster based method enables one to use existing variables for presenting subset of original variables. These variables are then more easily coupled with optimization problem formulation, because no new and completely different type of variables are presented.

6.2 Comparison to surrogates

As can be seen from the classification by Jin (2005) (introduced in Section 3.2), the surrogates are usually built for problems or functions in the product space. The cluster based surrogate is however built in the decision space instead. Because the purpose of the cluster based surrogate is also to model the original problem and to reduce the computational load, this approach should be classified as a surrogate also. The list presented by Jin (2005) is not then a comprehensive one. This classification having three product space surrogate types – problem approximations, functional approximations and objective approximations – should be extended with *variable approximations*.

If the original three-class classification of the surrogates is still to be used, the cluster based surrogate could be seen as a function approximation or an objective approximation: On the one hand the clustering, and so the surrogate, is formed in the decision

space to reduce the number of variables, and in the mathematical programming this then also requires and leads to modifying of the objective functions, as presented in Section 4.2. On the other hand, in a cluster based surrogate one variable is used for approximating other variables that are closely related to it, which is similar to how objective approximation works.

An advantage of the cluster based surrogate when compared to many other surrogate models is that it does not really require variable selection. As the cluster based surrogate uses all the existing variables for clustering and then automatically selects the most representative ones, manual choosing is not needed at all. Especially in the cases involving a large number of variables this feature of the surrogate becomes useful.

Unfortunately, the cluster based surrogate is not updatable during the optimization phase and is aimed only for linear (mixed) integer optimization problems. At the moment there is also no way to handle additional constraints in the optimization. Handling the constraints with the surrogate should also be studied in the future.

6.3 Possibilities of using different weighting method

In the presented cluster based surrogate method each proxy variable is assigned a weighting coefficient according to the proportion of variables it is presenting. This choice is based on assumption that the relative importance of each proxy variable is equal with the proportion of the variables it is representing in the modified multiobjective formulation (4.2). When the clusters are approximately equal in their shape and size this assumption should be valid.

However, different similarity metrics, clustering algorithms and numbers of clusters may produce clusters of very different shapes and sizes. With certain data sets this may be inevitable independently of the parameters used, or the clustering may be given ready so that there is not any possibility for impacting its characteristics. In these cases choosing the proxy variable may be challenging, but more than that the

relative importance of proxy variables may not correspond to the proportion of the variables they are representing.

Because of this, a more sophisticated approach for defining the significance and representativity of each proxy could be considered. The possible alternative approaches could consider the relative hypervolume of each cluster, the variance of variable values in each cluster or some metrics for assessing the impact of different shapes, for instance. A different weighting approach would also require different approach for mapping the proxy decision variables to the original decision variables. Exploring and comparing different weighting methods and mapping approaches could be a good research question in the future.

6.4 Possibilities of using alternative clustering approaches

Because of the way each proxy variable and its weighting coefficient are representing clusters, the shapes of clusters may have great impact on the accuracy of the cluster based surrogate. This issue is the main reason why a valid clustering by the machine learning point of view may not be the best possible one for a cluster based surrogate. The main application of clustering being machine learning and classifications that are meaningful in the information extraction, it would be worth considering how the cluster based surrogate could be modified to be more applicable with this kind of needs also. If these issues could be handled, also more “realistic” clusterings could be used for forming the cluster based surrogate for optimization.

One possible approach for solving the problem of different clusterings needed for different purposes would be multiobjective clustering by Handl and Knowles (2007). As clustering itself is a cumbersome task with simultaneous goals of maximizing the similarity of objects within clusters and minimizing it between different clusters, multiobjective clustering could serve the traditional clustering task even without the surrogate building aspect. The objectives of cluster based surrogate being somewhat similar with the traditional clustering but having some specified needs in addition, multiobjective clustering could offer a tool for finding satisfying compromise for both

modeling the classes of the phenomenon and giving accurate enough surrogate for optimization. The applicability of the multiobjective clustering for the surrogate building purposes could be studied in the future also.

6.5 Possibilities for choosing the number of clusters

One of the main issues of the cluster based surrogate – and clustering in general – is the number of clusters. As was seen in Section 5.3.4 the single objective optimization results using the surrogate were quite good even with a small number of clusters and the results improved linearly with the increase in the number of clusters. In this case, the real optimal values were known and it was easy to compare different clusterings.

The cluster based surrogate uses the average values of multiple variables to approximate all those variables so it can be assumed that the optimization results attained via surrogates are more close to average results than to the optimal ones. This means that when maximizing using the surrogate the values are probably less than the real maximum (and greater if minimizing). Based on this hypothesis one could always choose the clustering which gives the maximal optimization results (or minimal if minimizing). If the results only get better with the increase in the number of clusters without starting to degrade after some point, the number of clusters could probably be chosen based solely on the time requirements of the optimization and forming the clustering.

Even though the surrogate based optimization results usually are more average than the real ones, there still exists possibility that the optimization results attained via the surrogate are better than what really is possible to attain. In the cases when single objective optimizations using the original variables are not possible due to the size of the data, this cannot be verified and choosing the number of clusters based on the best attained value causes bias. In those cases it is still possible to use the original variables so that the decision values of the proxy variables are mapped into the original variables and so “new” optimal objective function values are produced. Because these values are based on the original values and original problem formulation, they can

never exceed the real ideal values, and can be used for verifying the optimization results of the cluster based surrogate. However, some other more accurate verification methods and metrics should be found or developed in the future.

In addition to these, previously mentioned multiobjective clustering of Handl and Knowles (2007) could be utilized for choosing the proper number of clusters also. If the applicability of multiobjective clustering for cluster based surrogate is studied later on, this aspect should also be included into the observations.

6.6 Effects of using different Ideal and Nadir values

In ILP optimization using cluster based surrogate all the values and reference points appear to the optimization algorithm through the values of the surrogate. This means that also the scalarizings of the synchronous NIMBUS method, for instance, are executed in the reduced decision space. Because of this, it would seem justified and harmless to use the ideal and nadir values attained from the surrogate model also. There are some issues with that approach, however.

The surrogates being only approximations of the original variables or data it is very likely that the ideal and nadir values attained using the surrogate are not the same than the original ones, but more averaged as explained in previous Section 6.5. As can be seen in Figure 2 the ideal and nadir vectors are used as projecting directions for different scalarizing functions. If these points are not on their correct places the scalarizing results of the given reference point become twisted. At worst, this makes some areas of Pareto front unobtainable via scalarizings.

Based on empirical tests, using the surrogate based ideal and nadir values produce more average results than if using original ideal and nadir values. Fortunately, interactive multiobjective optimization and using the scalarizing functions is an iterative process, during which the decision space is explored and inspected. Because quite seldom in the multiobjective optimization the decision maker ends up selecting a solution very close to ideal or nadir value of one objective, the inability to reach these points is possibly not crucial. It is still important to note this during the process.

6.7 Approximation errors in multiobjective optimization

In cases when the real ideal values of objectives are known without the surrogate i.e. surrogate is only used for lightening the optimization process, the differences of the surrogate based ideals and real ones could be used for estimating the modeling errors involved in the surrogate. That approach was tried in Section 5.5.2, but in that case it was not possible to discover any differences or errors. There might be some potential in that kind of approach and this should be studied more thoroughly in the future.

Naturally, if the original single objective optimal values are not known, this approach cannot be employed at all. For these cases, some measurement methods and metrics should be researched. This is also closely related to questions proposed in Section 6.5.

6.8 Extending Cluster Based Surrogate to Mixed Integers

In the presented cluster based surrogate approach and in the Boreal Forest case study the variables were only integer valued. This makes the problem only a integer linear programming problem. As the mixed integer linear problems (MILP) are quite a lot like the integer linear problems (ILP) it would be possible to extend the presented approach to include mixed integers also with quite a little effort.

One possible addition to the method if used with mixed integer variables would be changing the selection of proxy variable from clusters. Because of the discreteness of all the variables, the variable closest to the abstract center of the cluster was chosen. If there still was some continuous variables it could be considered to cluster them separately and use their real cluster centers as the proxy variables. With real valued variables this still may not be trivial or unambiguous. Easier approach would be leaving the real valued variables completely out from the surrogate forming and only form cluster based surrogate from the discrete ones.

In addition, some guidelines and specific instructions for using cluster based surrogate with MILP problems should be developed and tested in the future.

7 Conclusions

The amount of data has increased significantly during the last years, and it is also desired to be used for making decision. Even though available processing power has also increased, that growth has not been large enough and new approaches for handling large data-driven optimization problems are required. Especially when cognitively effective interactive decision making methods are used, some ways to reduce the computation times of iterations are needed. For this purpose, a computationally cheaper cluster based surrogate model was developed and tested in this thesis.

The developed cluster based surrogate was proved effective for compressing discrete decision space and for reducing computational load of large scale integer linear problems. In addition, the inevitable approximation errors involved in the cluster based surrogate model were moderate. When the cluster based surrogate was used in collaboration with an interactive multiobjective optimization method, the delay times were greatly shortened and the decision maker was satisfied with much smoother decision making process.

Different aspects, strengths and weaknesses of the developed surrogate were discussed in Chapter 6 along with many new research ideas. These suggested research topics were:

- How to better handle constraints with the clusters based surrogate?
- What other weighting methods could be used with the cluster based surrogate?
- If and how multiobjective clustering could be used for forming cluster based surrogate?
- How to verify the accuracy of the surrogate if the original single objective optimization results are not available?
- How to approximate the modeling differences between different objectives before the multiobjective optimization? Is it possible to adjust the surrogate accordingly based on these differences?

- How to extend the cluster based surrogate for different kind of mixed integer linear problems?

The major drawback of the cluster based surrogate method is its narrow applicability. The requirement for a large discrete decision space with similar kind of decision variables greatly cuts down the number of applicable problems. Even if this requirement is fulfilled, clustering the data may not produce desirable results and the cluster based surrogate approach may fail to provide accurate outcomes. Because of this, the developed surrogate and method should be tested for larger and completely different problems also.

As a whole, this study proves that reducing the computational requirements of multi-objective integer linear optimization problems is possible by using relatively simple clustering approach. The presented method is far from being generally applicable for even all the linear integer problems, but especially for forest management problems it has proved to be effective.

Bibliography

Anderson, C. 2008. “The End of Theory: The Data Deluge Makes the Scientific Method Obsolete.” June 23. Accessed February 6, 2018. https://www.wired.com/2008/06/pb-theory/.

Anderson, V. L., and R. A. McLean. 1974. *Design of experiments: a realistic approach*. Vol. 5. CRC Press.

Arabie, P., L. J. Hubert, and G. De Soete. 1996. *Clustering and classification*. World Scientific.

Arbuckle, J. L. 1996. “Full information estimation in the presence of incomplete data.” *Advanced structural equation modeling: Issues and techniques* 243:277.

Broomhead, D. S., and D. Lowe. 1988. *Radial basis functions, multi-variable functional interpolation and adaptive networks*. Royal Signals / Radar Establishment Malvern (United Kingdom).

Buchanan, J. T. 1997. “A naive approach for solving MCDM problems: The GUESS method.” *Journal of the Operational Research Society* 48 (2): 202–206.

Buntine, W., and A. Jakulin. 2004. “Applying discrete PCA in data analysis.” In *Proceedings of the 20th conference on Uncertainty in artificial intelligence,* 59–66. AUAI Press.

Cheng, R., Y. Jin, K. Narukawa, and B. Sendhoff. 2015. “A Multiobjective Evolutionary Algorithm Using Gaussian Process-Based Inverse Modeling.” *IEEE Transactions on Evolutionary Computation* 19 (6): 838–856. ISSN: 1089-778X.

Chugh, T., N. Chakraborti, K. Sindhya, and Y. Jin. 2017. “A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem.” *Materials and Manufacturing Processes* 32 (10): 1172–1178. ISSN: 1042-6914.

Chugh, T., K. Sindhya, J. Hakanen, and K. Miettinen. 2017. “A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms.” *Soft Computing:* 1–30.

Chugh, T., K. Sindhya, K. Miettinen, Yaochu Jin, T. Kratky, and P. Makkonen. 2017. “Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system.” In *2017 IEEE Congress on Evolutionary Computation (CEC),* 1541–1548.

Cui, X., P. Zhu, X. Yang, K. Li, and C. Ji. 2014. “Optimized big data K-means clustering using MapReduce.” *The Journal of Supercomputing* 70 (3): 1249–1259. ISSN: 1573-0484.

Dai, W., T. Chai, and S. X. Yang. 2015. “Data-Driven Optimization Control for Safety Operation of Hematite Grinding Process.” *IEEE Transactions on Industrial Electronics* 62 (5): 2930–2941. ISSN: 0278-0046.

Deb, K. 2012. *Optimization for engineering design: Algorithms and examples.* PHI Learning Pvt. Ltd.

English, T. M. 2000. “Optimization is easy and learning is hard in the typical function.” In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512),* vol. 2, 931 vol.2.

Eskelinen, P., K. Miettinen, K. Klamroth, and J. Hakanen. 2010. “Pareto navigator for interactive nonlinear multiobjective optimization.” *OR spectrum* 32 (1): 211–227.

Everitt, B. S., S. Landau, and M. Leese. 1993. “Cluster analysis.” *Edward Arnold and Halsted Press,*

Fahad, A., N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras. 2014. “A survey of clustering algorithms for big data: Taxonomy and empirical analysis.” *IEEE transactions on emerging topics in computing* 2 (3): 267–279.

Forrester, A. I. J., and A. J. Keane. 2009. “Recent advances in surrogate-based optimization.” *Progress in Aerospace Sciences* 45 (1): 50–79. ISSN: 0376-0421.

Gaspar-Cunha, A., and A. Vieira. 2005. “A Multi-Objective Evolutionary Algorithm Using Neural Networks to Approximate Fitness Evaluations.” *Int.J.Comput.Syst.Signal* 6 (1): 18–36.

Giannakoglou, K. C. 2002. *Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence.* ID: 271425.

Grave, K. De, J. Ramon, and L. De Raedt. 2008. “Active learning for high throughput screening.” In *International Conference on Discovery Science,* 185–196. Springer.

Guo, D., T. Chai, J. Ding, and Y. Jin. 2016. “Small data driven evolutionary multi-objective optimization of fused magnesium furnaces.” In *2016 IEEE Symposium Series on Computational Intelligence (SSCI),* 1–8.

Guyon, I., and A. Elisseeff. 2003. “An introduction to variable and feature selection.” *Journal of machine learning research* 3 (Mar): 1157–1182.

Handl, J., and J. Knowles. 2007. “An evolutionary approach to multiobjective clustering.” *IEEE transactions on Evolutionary Computation* 11 (1): 56–76.

Hart, W. E., C. D. Laird, J-P Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola. 2017. *Pyomo–optimization modeling in python.* Second. Vol. 67. Springer Science & Business Media.

Hart, W. E., J-P Watson, and D. L. Woodruff. 2011. “Pyomo: modeling and solving mathematical programs in Python.” *Mathematical Programming Computation* 3 (3): 219–260.

Hartikainen, M., K. Miettinen, and M. M. Wiecek. 2012. “PAINT: Pareto front interpolation for nonlinear multiobjective optimization.” *Computational optimization and applications* 52 (3): 845–867.

Husken, M., Y. Jin, and B. Sendhoff. 2005. “Structure optimization of neural networks for evolutionary design optimization.” *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 9 (1): 21–28.

Hwang, C-L, and A. Masud. 1979. *Multiple Objective Decision Making, Methods and Applications: A State-of-the-art Survey.* Springer-verlag.

Jain, A. K. 2010. “Data clustering: 50 years beyond K-means.” *Pattern Recognition Letters* 31 (8): 651–666. ISSN: 0167-8655.

Jain, A. K., and R. C. Dubes. 1988. *Algorithms for clustering data.* Prentice-Hall, Inc.

Jin, Y. 2005. “A comprehensive survey of fitness approximation in evolutionary computation.” *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 9 (1): 3–12.

———. 2011. *Surrogate-assisted evolutionary computation: Recent advances and future challenges.*

Jin, Y., M. Husken, and B. Sendhoff. 2003. “Quality measures for approximate models in evolutionary computation.” In *GECCO,* 170–173.

Kanungo, T., D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. 2002. “An efficient k-means clustering algorithm: Analysis and implementation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7): 881–892.

Kaufmann, L., and P. Rousseeuw. 1987. “Clustering by Means of Medoids” (January): 405–416.

King, R. D., K. E. Whelan, F. M. Jones, P. GK Reiser, C. H. Bryant, S. H. Muggleton, D. B. Kell, and S. G. Oliver. 2004. “Functional genomic hypothesis generation and experimentation by a robot scientist.” *Nature* 427 (6971): 247–252.

Knowles, J. 2006. “ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems.” *IEEE Transactions on Evolutionary Computation* 10 (1): 50–66. ISSN: 1089-778X.

Knowles, J., and H. Nakayama. 2008. “Meta-modeling in multiobjective optimization.” *Multiobjective optimization:* 245–284.

Larichev, O. I. 1992. “Cognitive validity in design of decision-aiding techniques.” *Journal of Multi-Criteria Decision Analysis* 1 (3): 127–138.

Laukkanen, T., T-M Tveit, V. Ojalehto, K. Miettinen, and C-J Fogelholm. 2010. *An interactive multi-objective approach to heat exchanger network synthesis.*

Lim, D., Y. Jin, Y. S. Ong, and B. Sendhoff. 2010. “Generalizing Surrogate-Assisted Evolutionary Computation.” *IEEE Transactions on Evolutionary Computation* 14 (3): 329–355. ISSN: 1089-778X.

Loshchilov, I., M. Schoenauer, and M. Sebag. 2010. “Comparison-based optimizers need comparison-based surrogates.” *Parallel Problem Solving from Nature, PPSN XI:* 364–373.

Maney, K. 2014. “DATA IS GROWING FASTER THAN COMPUTING POWER.” August 15. Accessed February 6, 2018. `http://www.newsweek.com/2014/08/15/computers-need-be-more-human-brains-262504.html`.

Miettinen, K. 1999. *Nonlinear multiobjective optimization.* Vol. 12. Springer Science & Business Media.

Miettinen, K., J. Hakanen, and D. Podkopaev. 2016. “Interactive nonlinear multiobjective optimization methods,” 927–976. Multiple Criteria Decision Analysis. Springer.

Miettinen, K., and M. M. Mäkelä. 1995. “Interactive bundle-based method for nondifferentiable multiobjeective optimization: nimbus §.” *Optimization* 34 (3): 231–246.

———. 2000. “Interactive multiobjective optimization system WWW-NIMBUS on the Internet.” *Computers & Operations Research* 27 (7): 709–723.

———. 2002. “On scalarizing functions in multiobjective optimization.” *OR Spectrum* 24 (2): 193–213. ISSN: 1436-6304.

———. 2006. “Synchronous approach in interactive multiobjective optimization.” *European Journal of Operational Research* 170 (3): 909–922. ISSN: 0377-2217.

Miettinen, K., F. Ruiz, and A. P. Wierzbicki. 2008. “Introduction to multiobjective optimization: interactive approaches,” 27–57. Multiobjective Optimization. Springer.

Miller, R. B. 1968. “Response time in man-computer conversational transactions.” In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I,* 267–277. ACM.

Mönkkönen, M., A. Juutinen, A. Mazziotta, K. Miettinen, D. Podkopaev, P. Reunanen, H. Salminen, and O. Tikkanen. 2014. *Spatially dynamic forest management to sustain biodiversity and economic returns.*

Morris, M. D., and T. J. Mitchell. 1995. "Exploratory designs for computational experiments." *Journal of statistical planning and inference* 43 (3): 381–402.

Myers, R.H., and D.C. Montgomery. 1995. *Response Surface Methodology: Process and Product Optimization using Designed Experiments.* Wiley, Chichester.

Nakayama, H., and Y. Sawaragi. 1984. "Satisficing trade-off method for multiobjective programming." *Interactive Decision Analysis:* 113–122.

Ojalehto, V., K. Miettinen, and T. Laukkanen. 2014. "Implementation aspects of interactive multiobjective optimization for modeling environments: the case of GAMS-NIMBUS." *Computational Optimization and Applications* 58 (3): 757–779.

Peura, M., M. Triviño, A. Mazziotta, D. Podkopaev, A. Juutinen, and M. Mönkkönen. 2016. "Managing boreal forests for the simultaneous production of collectable goods and timber revenues." *Silva Fennica* 50 (5). ISSN: 1044-5005.

Pohjanmies, T., M. Triviño, E. Le Tortorec, H. Salminen, and M. Mönkkönen. 2017. "Conflicting objectives in production forests pose a challenge for forest management." *Ecosystem Services.* ISSN: 2212-0416.

Pozo, C., R. Ruíz-Femenia, J. Caballero, G. Guillén-Gosálbez, and L. Jiménez. 2012. *On the use of Principal Component Analysis for reducing the number of environmental objectives in multi-objective optimization: Application to the design of chemical supply chains.*

Press, G. 2013. "A Very Short History Of Big Data." March 9. Accessed February 6, 2018. `https://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/2/`.

Salvador, S., and P. Chan. 2004. "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms." In *16th IEEE International Conference on Tools with Artificial Intelligence,* 576–584.

Sindhya, K., A. Manninen, K. Miettinen, and J. Pippuri. 2017. *Design of a Permanent Magnet Synchronous Generator using Interactive Multiobjective Optimization*.

Sindhya, K., V. Ojalehto, J. Savolainen, H. Niemistö, J. Hakanen, and K. Miettinen. 2014. "Coupling dynamic simulation and interactive multiobjective optimization for complex problems: An APROS-NIMBUS case study." *Expert Systems with Applications* 41 (5): 2546–2558.

Tabatabaei, M., J. Hakanen, M. Hartikainen, K. Miettinen, and K. Sindhya. 2015. "A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods." *Structural and Multidisciplinary Optimization* 52 (1): 1–25.

Triviño, M., A. Juutinen, A. Mazziotta, K. Miettinen, D. Podkopaev, P. Reunanen, and M. Mönkkönen. 2015. "Managing a boreal forest landscape for providing timber, storing and sequestering carbon." *Ecosystem Services* 14 (Supplement C): 179–189. ISSN: 2212-0416.

Triviño, M., T. Pohjanmies, A. Mazziotta, A. Juutinen, D. Podkopaev, E. Le Tortorec, and M. Mönkkönen. 2017. "Optimizing management to enhance multifunctionality in a boreal forest landscape." *Journal of Applied Ecology* 54 (1): 61–70. ISSN: 1365-2664.

Vanian, J. 2016. "Why Data Is The New Oil." July 11. Accessed January 29, 2018. `http://fortune.com/2016/07/11/data-oil-brainstorm-tech/`.

Vigneau, E., K. Sahmer, E. M. Qannari, and D. Bertrand. 2005. "Clustering of variables to analyze spectral data." *Journal of Chemometrics* 19 (3): 122–128.

Vigneau, Evelyne, and E. M. Qannari. 2003. "Clustering of variables around latent components." *Communications in Statistics-Simulation and Computation* 32 (4): 1131–1150.

Wang, H., Y. Jin, and J. O. Jansen. 2016. "Data-Driven Surrogate-Assisted Multiobjective Evolutionary Optimization of a Trauma System." *IEEE Transactions on Evolutionary Computation* 20 (6): 939–952. ISSN: 1089-778X.

Wang, H., Q. Zhang, L. Jiao, and X. Yao. 2016. “Regularity Model for Noisy Multiobjective Optimization.” *IEEE Transactions on Cybernetics* 46 (9): 1997–2009. ISSN: 2168-2267.

Wang, S., and X. Yao. 2013. “Using class imbalance learning for software defect prediction.” *IEEE Transactions on Reliability* 62 (2): 434–443.

Wierzbicki, A. P. 1982. *A mathematical basis for satisficing decision making.*

———. 1986. “On the completeness and constructiveness of parametric characterizations to vector optimization problems.” *OR spectrum* 8 (2): 73–87.

———. 1997. “On the role of intuition in decision making and some ways of multicriteria aid of intuition.” *Journal of Multi-Criteria Decision Analysis* 6 (2): 65–76.

Wolpert, D. H., and W. G. Macready. 1997. “No free lunch theorems for optimization.” *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82. ISSN: 1089-778X.

Zhou, Z., YS Ong, MH Nguyen, and D Lim. 2005. “A study on polynomial regression and Gaussian process global surrogate model in hierarchical surrogate-assisted evolutionary algorithm.” In *2005 IEEE Congress on Evolutionary Computation,* vol. 3, 2839 Vol. 3.