THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Product Configuration with respect to Multiple Criteria—a Mathematical Programming Approach

Peter Lindroth

CHALMERS | UNIVERSITY OF GOTHENBURG

Volvo 3P
Chassis & Vehicle Dynamics
Chassis Strategies & Vehicle Analysis
SE-405 08 Göteborg, Sweden

and

Department of Mathematical Sciences, Chalmers University of Technology
Department of Mathematical Sciences, University of Gothenburg
SE-412 96 Göteborg, Sweden

Göteborg, 2008

This work was supported by:
Volvo 3P

Product Configuration with respect to Multiple Criteria—a Mathematical Programming Approach
Peter Lindroth

NO 2008:15ISSN 1652-9715
Department of Mathematical Sciences, Chalmers University of Technology
Department of Mathematical Sciences, University of Gothenburg
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 1000

Printed in Göteborg, Sweden 2008

Product Configuration with respect to Multiple Criteria—a Mathematical Programming Approach

Peter Lindroth

Volvo 3P
Chassis & Vehicle Dynamics
Chassis Strategies & Vehicle Analysis

and

Department of Mathematical Sciences, Chalmers University of Technology
Department of Mathematical Sciences, University of Gothenburg

Abstract

The trend in the market for trucks is that highly specialized configurations are made available for the customers and that only a few completely identical configurations are manufactured. One reason for this development is that the optimal truck configuration for a certain customer is very specific and depending on, e.g., the environment in which the truck is to be used and for what transport mission. To achieve reasonable cost levels the manufacturer must be able to produce a limited set of configurations in a cost-effective way by using the same parts in different combinations, leading to a relatively small number of parts but a large number of possible configurations.

This thesis presents an approach to the configuration problem by modeling it from a multi-objective optimization perspective. By assuming that a product is described by a number of quality measures which different customers appreciate differently, the interesting configurations consist of the configurations that lie in the Pareto optimal subset of the decision space.

For a large number of objectives, multi-objective optimization becomes cumbersome; therefore a first appended paper provides a method for problem reduction such that the representation of the Pareto optimal set is kept as good as possible.

A second paper considers a simplification of the configuration problem by assuming that the decision variables are continuous and box constrained. A problem, in which the objective is to find an optimal representation of the Pareto optimal set, while the number of chosen values of the decision variables is limited, is formulated and solved for a number of test instances.

The thesis has been written in close cooperation with the product development department of Volvo 3P.

Keywords: optimization; multiple objectives; configuration management

Appended papers:

Paper 1: *Approximating the Pareto Optimal Set using a Reduced Set of Objective Functions* , Presented at the 22nd European Conference on Operational Research (EURO XXII), 2007. (with Michael Patriksson and Ann-Brith Strömberg)

Paper 2: *Multi-Objective Design of a Combinatorial Structure* , To be presented at the SIAM Conference on Optimization (OP08), 2008. (with Michael Patriksson and Ann-Brith Strömberg)

Acknowledgements

The research presented in this thesis has been financially supported by Volvo 3P. This support is gratefully acknowledged.

My first personal thank goes to my boss and industrial advisor Stefan Edlund for his constant enthusiasm and encouragement. I would also like to thank my academic supervisors Ann-Brith Strömberg and Michael Patriksson. Ann-Brith for always taking the time for stimulating discussions and Michael for teaching me an interesting subject.

Peter Lindroth
Göteborg, April 2008

Contents

1 Introduction

By modeling the problem of creating an optimal set of truck configurations given a great variety of customer requirements, this thesis constructs a framework for applying multi-objective optimization to the product development of a population of products.

The problem is approached from a mathematical programming perspective with an assumption that there are functions measuring the different qualities of a truck. This is a critical assumption, and for the framework to be an effective tool, it is an important task for industry to learn what the customers seek, and hence how quality should be measured.

The thesis is an attempt to bring the whole product development problem to the surface. It models the problem in mathematical terms, and then focuses on some well-defined subparts, which are then studied more extensively in the two appended papers.

1.1 Background

The ultimate goal of a commercial company is to maximize the long-term dividends to the stakeholders. To enable this it is necessary to offer the market products that match the market demands in a cost-effective way.

Volvo 3P is a business unit within the Volvo group responsible for product planning, product development, and purchasing for the four brands Volvo Trucks, Mack Trucks, Renault Trucks and Nissan Diesel. The work presented in this thesis has been carried out within the product development of Volvo 3P, whose interest in this project comes from the belief that a structured and controlled approach to a process that today is very complicated and hard to grasp is important in order to maintain a long-term competitiveness in an increasingly complex and global truck market.

1.2 Differentiation and Pareto optimality

Volvo is a global company whose products are used in markets with very different characteristics concerning operating environments, legislations, and transport missions. This fact, together with a stiff competition, has led to a high degree of specialization and truck customization for individual customers. Thus, to be able to fulfil the demands of the customers, a great variety of truck configurations must be offered, illustrated in the Figures 1 and 2.

Around $200{,}000$ trucks are produced within the Volvo group each year. The average number of completely identical trucks is very small.

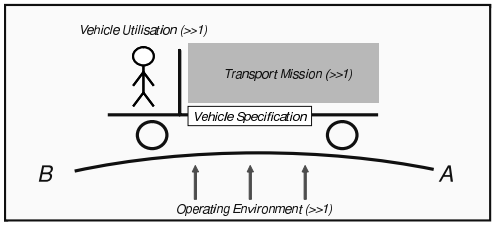


Figure 1: Many different operating environments, many different transport missions, and many different (driver dependent) vehicle utilizations require a wide range of differentiated vehicle configurations.

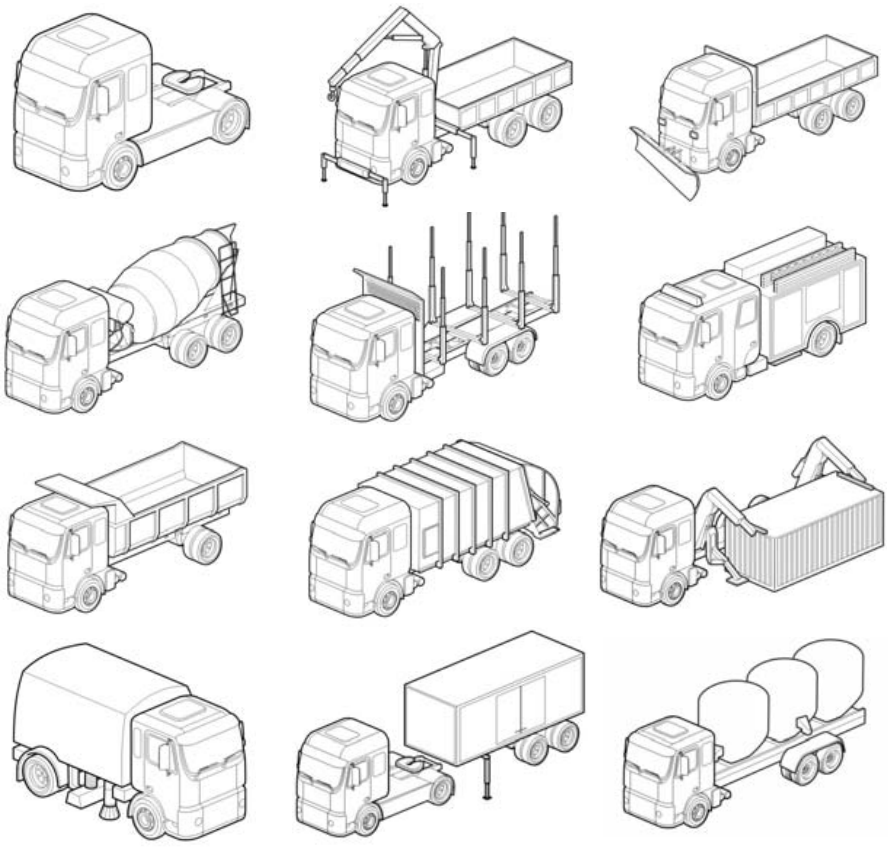


Figure 2: A number of different superstructures illustrating various transport missions.

Each individual configuration has a certain perceived quality for each specific customer. By assuming that quality can be divided into a number of components one can illustrate the quality of a configuration as a polygon in a diagram of the type drawn in Figure 3. (Note that the term "quality" is used for all objectives, it should not be confused with the classical "product quality").

Further, if we assume that each customer measures the different quality components principally equivalently, but that they possibly appreciate them differently depending on their operating environment, transport mission, financial strength, etcetera, then it becomes obvious that one is not interested in producing a truck that is worse than some other possible truck in *all* of the quality components.

Formally, this is denoted "Pareto optimality", a concept which is defined mathematically in Section 5. In Figure 3, a Pareto optimal solution corresponds to a polygon which is not entirely enclosed in any other polygon.

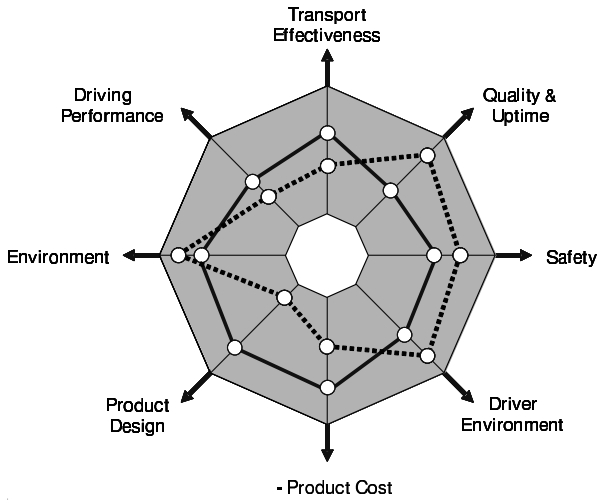


Figure 3: The quality of each configuration (in the figure there are two) can be represented as a polygon intersecting each axis at the numeric measure of the corresponding quality. The gray region represents possible values of the quality measures. Good values of a quality correspond to values far from the center of the diagram.

1.3 Purpose

The vision of the project is to design a procedure for reducing the size of the assortment of variants used for configuring the population of trucks consistently and systematically in such a way that each customer is guided towards a configuration that is at least as good as the one that he/she would have chosen without using the procedure. The drive is essentially to satisfy as many customers (in selected market segments) as possible using the least number of technical solutions, and by this in-

crease the profitability for the company. The resulting set of configurations should be optimal with respect to a suitable measure. The expected goal for the company is a reduction of costs related to product development, production, maintenance, and sales of trucks. We envisage a computer environment in which an optimal set of variants of technical solutions together with a controlled configuration management guides the customer towards the most suitable specification in a set of available truck configurations. Another goal is to create an understanding of what a company can gain from the use of mathematical optimization and what it requires from the company in order to work (e.g., numerical measures of quality parameters). The expected goal for academia is an increased insight into complex product development, and how mathematical models and methods could be adapted to such an environment. Through feedback from the product planning process, academia will learn more about where the biggest differences lie between real, "dirty", industrial problems and academic, "clean", ones. Finally, the expected goal for the customers is guidance towards and within the set of available and verified appropriate vehicle configurations.

The full title of the project, in which this thesis is a part, is *"Product configuration with respect to multiple criteria in a heterogeneous environment within an extended enterprise"*. The "multiple criteria" is directly related to the different qualities of a truck, the "heterogeneous environment" represents the large number of different customers, and "extended enterprise" symbolizes that interdependence with other actors outside of the company, e.g., system suppliers, are of importance.

1.4 Outline

The theme of the thesis is to interpret the practical problem mathematically and to make some technical contributions to selected parts. In the view of modeling the problem in mathematical language, approaches are suggested on how to move towards the aims of the vision.

In the thesis, the scientific areas needed for illuminating the configuration problem are described. We intend to give a description as complete as possible of the configuration problem. It should, however, be pointed out that even though we describe the complete problem, we do not claim to be able to solve it without a major delimitation of the scope. There is too much knowledge missing about the product and the customers that has to be taken into consideration for the complete problem to be solved.

In Section 2 the current structure of the product development is described. This is our starting point, which also defines the framework that we have to adapt to. Section 3 takes a complex systems view on the configuration problem and Section 4 presents the different systems that have to be discretized or aggregated for an efficient configuration management. In Section 5, multi-objective optimization is reviewed with a special focus on the configuration problem. Then, in Section 6, the objective functions, the variables, and the constraints used to model the configuration problem as a multi-objective optimization problem are discussed. In Section 7 the issue of clustering the configurations is discussed. Finally, in Section 8, we summarize the main parts of the problem and collect and review the scientific areas needed

to describe and to solve it. We indicate where we—by interpreting the true problem and by lifting out subproblems, formulating these as general mathematical problem and treating them as such—have made contributions to the field of complex product development. The most critical limitations are also highlighted. Section 9 summarizes the appended papers, in which generalizations of selected parts of the complete problem are studied.

2 The current conditions at Volvo

2.1 The organization

There is a working product development process. If it was shown by a mathematical modeling of the actual problem that revolutionary changes should be carried through, then it is not clear that such changes should be performed directly. First, it would be naïve to believe that a mathematical model can capture the reality perfectly and find solutions that are truly optimal for the practical problem. It is probable that some characteristics of the true problem—handled by the current process—are missing in the model. Second, revolutionary changes hardly would be accepted by the organization. Third, it is a dynamic process and if changes are introduced, then so are the surroundings defining the problem.

The conclusion is that the mathematical model used should be formulated such that it, by construction, does not lead to totally different solutions from what are utilized in the current process. Changes from the current situation should be introduced gradually, making it possible for the organization to adapt and learn, and to take care of possible problems, not known in advance, as soon as possible.

2.2 The product structure: variant families, variants and restrictions

The managing of different configurations of trucks has led to an evolution of the product structure over time. An organized product structure can be utilized in many ways: It makes it possible for different departments to develop different parts of the truck. It enables the company to ensure that the right physical material is produced and assembled. It also makes it possible to secure that each truck fulfils the legislations that are valid at the market in which it is sold. At Volvo, a truck is described by its so called *variants*, each of which belonging to a *variant family*. The variant families describe a vast variety of entities, some of which represent physical choices such as engine type or frame width, while others describe, e.g., the type of roads that the truck is aimed for or in which market it is to be used. The *product type* is a very coarse division of the truck configurations, e.g., specifying the axle configuration, the type of cabin, and whether the truck is a tractor (to which a semi-trailer must be connected, cf. the upper-most configuration to the left in Figure 2) or a rigid (everything else). In principle, a certain truck configuration is completely defined by its variants, where in most cases exactly one variant has to be chosen from each variant family valid for that product type. Thus, the truck specification—a list of all the variants

chosen—can be regarded as a fingerprint of the truck. Figure 4 shows a part of such a list. A few of the entries are highlighted to give some examples of actual variants and variant families; they are described in Table 1. Figure 5 contains pictures of different rear spring types, each corresponding to a variant in one of the variant families.

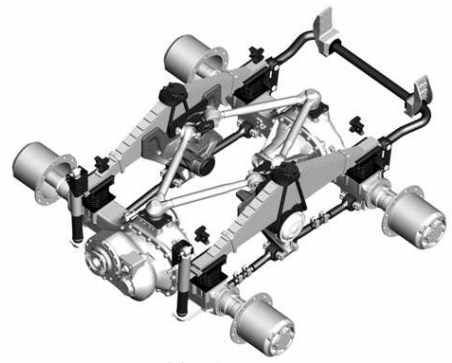
Vehicle Specification

04-TLV	FM 62R B	MAT-GREY	UXFUEL	ULADDER	SPOTP-R	UDECALTP
TMPF166	URUPP	UACMR	U1ACM	U2ACM	FFL1295	USPWCAD
UBULBKIT	ULAMPIN	UGAUGETP	UCHAINS	VERSION2	EMBL-TRP	WSTWD-S
UCROSSRE	RFE-T	UEGR	TYPE-TM	TP-DICON	ULOADIND	6*2
DASH-BLU	CABPT-R1	RIGID	RC-ROUGH	GCW60.0	CONC-BAS	UALARMB
ENG-VE13	VH4.4	WARNVEST	UBUNKBOX	UAIRBAG	EXHP-SAC	SUNV-SMO
APF-ENH	USIGNSWC	UCOMEQ	SUPH-TEX	SEATBRIA	AUXSW-6	WRITEPAD
USIGNS	TELECARD	UFUFIWH	STWPOS-L	RAA21T	DIMPLATE	UPAXLE
VW2600	MATT-PRE	FA-HIGH	FRAX-UNI	FAL9.0	FATYPE90	FAP3040
FAA10	TA-FIXED	RAL21	RADT-AR	RAP7340	LUGOP-BS	INST-MED
AUD-MED2	UADR	VERS-BAS	CHH-STD	BSR1365	URSTS	UCBRKTOW
UCBRKT	D13A440	VOLENG	ETOR2250	CHOSE-RU	EBR-VEB	BUGNET
RADI-NAR	VAL-BAS3	WTD-DUAL	UMRUP	UWCAP	CLUT-BAS	COOLC40
USPEEDDU	SPEED90	CRUISEC	FAN-VISE	UFUELFP	AIRIN-HI	FUELSEC2
FUELTFIL	UAUXFUEL	ULFUEL	RFUEL490	FUELTS71	TNK-SING	FTANK-AL
FCAP-L	MUF-HOR	EXD-VERR	EXSH-ST	EXSTH40	EAS-SCR	EM-EC06B
NR-80EC	ACL1ST	URPLCR	ENGPROT	UPSCOOL	24V	TRANSF-H
UTRACON	UTABS	UACC	PHONE	LOWB-DAY	UFLASHL	HL-ASYMR
DETECT-S	HORN-ELS	SWS-RP	HORN-JER	HBAT-CF	ELCE-CK	BBOX-L
BBOXC-BA	2BATT225	UHOOKSC	UFMS	UADTL	RAG-ADBL	TWEETER
URTOLL	EST-AID	TL-BAS	MAINSW-M	FOGL-WC	DRIVL2	FUP-X1
UHLADJ	HL-CLEAN	UIMMOBIL	UIDLAMP	MARKL-SR	UALTTR	DOL24S12
RLIGHT-W	UWPREBOC	UWLAMP	BEACONA2	SPG-KM	ESUNV2P	ASCOP85
UPSCAT	SBD-BLAC	SBP-BLAC	EPTT650	CIRCP-F	TEMP-AMB	TEMP-TRA
UGAUGEAM	24AL12OB	SI-GRAF	1DAY-STD	CALI-DIG	AIRRI-D	DIFL-SWF

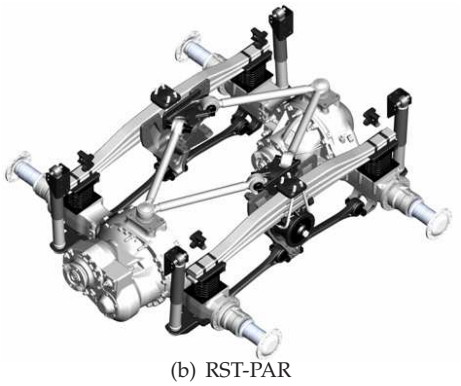
Figure 4: A part of a truck specification. Each entry is a code for a variant in some variant family.

Variant	Variant family	Description of the variant
RC-ROUGH	Road condition	Badly maintained road
6*2	Axle arrangement	6 wheels thereof 2 driving
RFUEL490	Fuel tank at the RHS	490 litre right side fuel tank

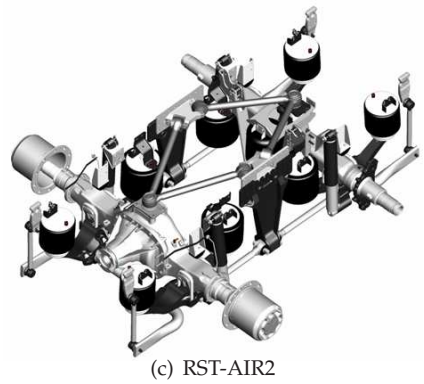
Table 1: Examples of items in the truck specification shown in Figure 4.



(a) RST-MUL



(b) RST-PAR



(c) RST-AIR2

Figure 5: An illustration of different rear spring types, defined by one of the variant families. a) has a multi-leaf spring, b) has a parabolic spring and c) has an air spring. The captions above are the names of the corresponding variants.

For a certain product type there are about 500 valid variant families, each containing two or more variants. Thus, the number of possible configurations—if we define a unique configuration by the selection of one variant from each variant family—is huge (much larger than $2^{500} \approx 10^{150}$). In reality, however, not all variants can be combined just anyhow, due to geometrical, physical, functional, and legislative reasons. In the product structure this is documented by so called *restrictions*, which represent disallowed combinations of two or more variants. This way of defining feasible and infeasible configurations is sometimes denoted as using a system of *positive variants*. The opposite, *negative variants*, presupposes that all configurations are infeasible except the ones that are explicitly defined as feasible. Which of the systems to use is a fundamental strategic decision for the company. The advantages of using a system of positive variants are that the flexibility increases and that in practice it leads to more configurations and thus more possibilities for customer adaptations. A serious disadvantage is, however, that positive variants impose a large complexity due to the large number of restrictions. In addition, the set of restrictions is dynamic in the sense that restrictions are added over time whenever infeasibilities are discovered, and it is a huge task to maintain the set of restrictions and to adapt it to newly developed variants and to changes in the legislations. The number of restrictions are in the order of $100{,}000$. Since each restriction represents an prohibited combination of variants, it might cut off a large number of configurations from the feasible set.

What drives different customers not preferring identical configurations is their objective and subjective preferences. The objective category comprises the transport mission and the geographical location (directly related to the operating environment and the legislation to adapt to). Subjective preferences are the feature profile wanted (wishes in the objective space Z, cf. Section 5) and opinions about the technical solutions (i.e., in the design space X, cf. Section 4.1). Intuitively, it seems like the only important thing is how the truck behaves and not how the actual technical solution is constructed (i.e., the location in the design space). However, there might be underlying requirements such as that the customer may wish to reduce the variety in his/her vehicle fleet, or that an old superstructure should be used also on a new truck. Furthermore, the customer might have strong feelings for a certain design, such as the size of the engine. A reasonable assumption is that the customer presets the transport mission of the truck (including what kind of superstructure that will be used), the cabin type, the engine (or driveline), and the axle configuration. This is in principle equivalent to defining the product type.

Figure 6 illustrates how the vehicle specification is defined. First, objective and subjective design requirements preset some of the main characteristics. Then the customer feature requirements yields the specification.

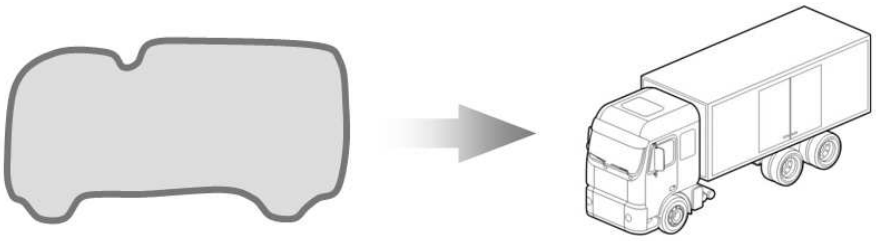


Figure 6: The gray silhouette to the left symbolizes the main characteristics of the truck that are preset by the customer requirements. Together with a preferred feature profile, the vehicle specification, illustrated to the right, should be defined.

3 The complexity of product development

A *complex system* can be defined as a collection of interacting parts whose collective behavior cannot be understood by studying the parts separately [5]. When incorporating the daily meaning of complexity as something being very large and incomprehensible into the definition, the product development of trucks can be viewed as a complex environment in at least three dimensions. These dimensions represent *functional*, *combinatorial*, and *dynamic* complexity; they are illustrated in Figure 9 on page 12.

3.1 Functional complexity

With functional complexity, we mean that the functional or feature requirements cannot be partitioned into independent measures, each of which could be controlled individually.

The feature structure of a truck is at Volvo divided into 8 *feature areas*. These are illustrated in Figure 3. The areas are broken down into 32 *customer features*, such as e.g., *durability*, *ride comfort*, *exterior noise* and *fuel economy*. The customer features are further divided and translated to *technical features* (and *technical subfeatures*), which should be quantitatively measurable. However, the measurable quantities are not independent of each other; this constitutes the functional complexity: the truck has a large number of functionalities that are to be controlled and steered against and the connections between the functionalities can be very complicated, see Figure 7. For this reason, it is hard to construct a configuration with a wanted feature profile by analyzing the features separately. This is no real issue when the problem is approached by optimization, however it might be hard when using traditional systems engineering techniques [36].

Let the features of a certain product be numbered from 1 to N^{feat} and define the symmetric matrix A of size $N^{\text{feat}} \times N^{\text{feat}}$ by $a_{ij} = 1$ if function i is connected to function j and 0 otherwise . The functional complexity of the product is related to

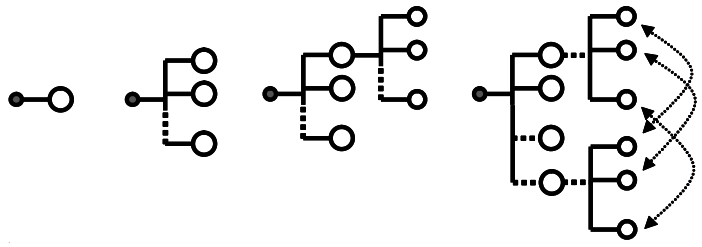


Figure 7: The features of a truck are connected and cannot be broken down into independent entities. The dotted arrowed symbolize connections between the subfeatures, where a design against one of subfeature inevitably also affects some of the others.

the density (the portion of 1:s) of A. A product with the lowest possible functional complexity corresponds to the unit matrix.

$$\begin{bmatrix} 1 & 0 & 1 & \cdots & 1 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 1 & \vdots \\ 0 & 1 & 0 & \cdots & \cdots & 1 \end{bmatrix}$$

Figure 8: The density of the feature-feature connection matrix is related to the functional complexity of a product.

In the thesis [33] an aspect of functional complexity is studied. Here, the top-level functionality retardation is investigated. For an efficient retardation system, a number of subsystems are interconnected, providing subfeatures that together build up the top-level function. (Optimization techniques are used, and it is shown that there is a trade-off between different objectives.)

3.2 Combinatorial complexity

As noted in Section 2.2, Volvo works with highly combinatorial products, where parts are used in different combinations such that a large number of configurations becomes possible. With the notation used in [5] this type of complexity is called *complication*, where the main issue is the incomprehensibly large number of parts of a system and not the unknown relations between them.

Since the configurations share building blocks, there is a permanent balancing between creating designs that are good for a single configuration (an individual cus-

tomer) and creating designs that are good for a collective of configurations (a population of customers).

At Volvo, all of the variant families, variants and restrictions are documented in databases. In addition, also *links* are documented, steering which physical parts that should be used for the manufacturing including which holes in the frame that should be drilled, depending on the variants chosen. The so called *bill of material*, given by the variants through the links, is like a recipe for constructing the truck[1]. However, both the sets of restrictions and links are dynamic, and it is a huge task to keep them up to date when changing the assortment of variants. In the organization, this problem is generally called the problem of handling the *documentation*.

3.3 Dynamic complexity

The truck market is dynamic with a constant demand for the development of new technical solutions, partly to satisfy new legal requirements, partly to meet the competition. The product development is a dynamic process, and it is important not to box into a corner, but to choose a way of action that is likely to be accessible also in the future. In principle, one would like to design a product structure and a development process that is robust, meaning that it can quickly adapt and provide the market its requirements at reasonable cost levels, and with respect to possible transport scenarios of the future.

The dynamic complexity concerns the time axis of the development process, where the scale is from slow evolutionary changes to drastic revolutionary changes. Evolutionary changes corresponds, e.g., to the tuning of current variants to increase the quality of the truck population, while revolutionary changes corresponds to topological changes in the form of new variants and variant families representing completely new technologies forced by legislation or innovations.

3.4 The complexity space

We define the *complexity space* as the three dimensional space spanned by the functional, combinatorial and dynamic dimensions. Figure 9 illustrates the complexity space, illustrating the high complexity of the product development of trucks, with its substantial contributions in all three components.

The complete problem is located deep along each axis in the complexity space. The thesis [18] deals with computer aided design (CAD) and management of geometrical data for combinatorial products, where common parts are to be used in a large number of different configurations. The thesis takes a large step along the combinatorial complexity axis. In this work, we focus on the combinatorial as well as the functional complexity dimensions. With a higher concentration on how the different transport scenarios would affect product development and with a larger focus on robustness in several meanings, the dynamic complexity dimension could also be

[1]Continuing the cooking metaphor, one of the beliefs in this project is that if the customers would specify the taste, the nutrients and the cost of a dish instead of the ingredients, then the customers would get tastier, healthier and cheaper dishes, and the company would not need as many different ingredients.

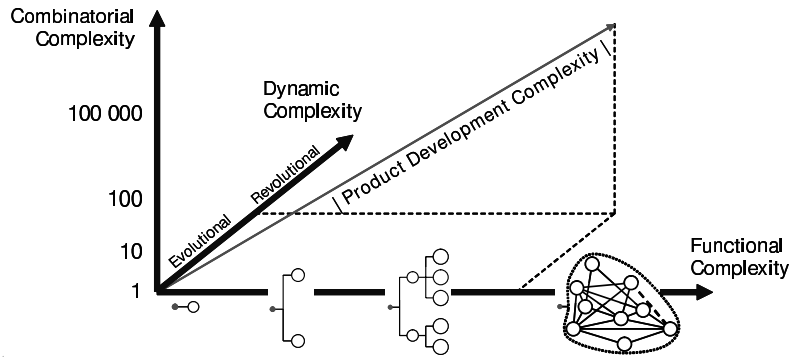


Figure 9: A complex environment. Product development of trucks comprises complexity components in all three dimensions (functional, combinatorial, and dynamic), thereby making the total product development complexity very high. (This thesis focuses on the functional and the combinatorial dimension.)

included in the analysis. Figure 10 roughly illustrates where other types of products are located in two components of the complexity space.

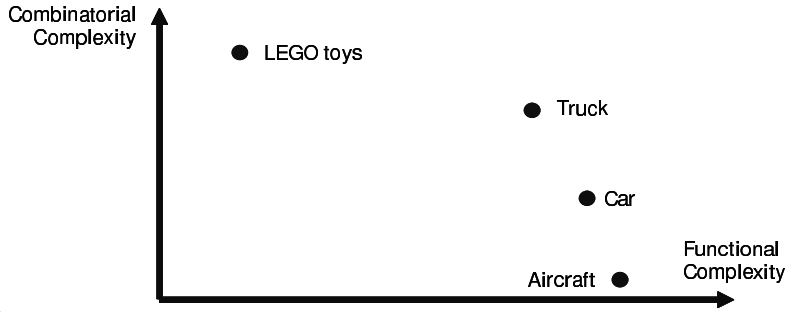


Figure 10: An illustration of the location of some products of different types in the complexity space projected onto the functional and combinatorial dimensions.

The notations of *functional* and *geometrical building blocks*, respectively, are introduced as being different descriptions of the parts building up the complex truck structure. The truck itself is a complex system, in which it is not possible to determine the overall behavior by studying the individual building blocks separately. The correspondence between the functional and geometrical building blocks varies depending on the scale on which the truck is viewed. In the geometrical description of the parts, the building blocks are the physical parts, e.g., screws and bolts on a fine scale, and axle installations and engines on a larger scale. In the functional description, the building blocks are, e.g., different signals and movements on a fine

scale, turning and braking on a large scale. On a very fine scale, the functional and the geometrical descriptions coincide, with each physical part having its own functionality. On an intermediate scale, the geometrical building blocks are connected to many functionalities, and similarly the functional building blocks depend on many geometrical parts. On a very large scale, where the truck is one physical entity with the function of moving from A to B, the descriptions coincide again.

Figure 11 illustrates how different building blocks in the truck (and in neighboring systems) interact. (Another feature of complex systems is that it is hard to define the system boundaries.)

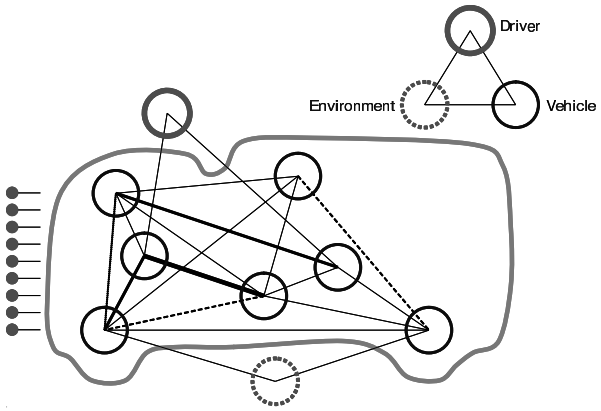


Figure 11: The truck is a complex system in itself. It is constituted by interacting building blocks determining the overall behavior. Different types of edges symbolize different strengths of connections.

It is not easy to say how the functional as well as the geometrical building blocks are connected to each other. What is known, at least partly, is one aspect on how variants are connected, namely through the restrictions. A small study of the distribution of restrictions over the variants has been performed. We define a graph $G = (V, E)$, where the set of nodes, V, represents the valid variant families for a certain product type, and where the set of edges, E, represents the pairs of nodes representing variants present in the same restriction. One interesting observation is that the graph is almost connected with only a few exceptions. Hence, any configuration optimization must be performed over a large non-separable feasible set.

4 Model discretization/aggregation

In the global market, the number of customer types is very large. Also, the number of possible configurations is huge. The company is not the only actor profiting from a limitation of the number of configurations. The cost of a truck is one of the quality

measures of a configuration and a coarser discretization (to fewer elements) of the available set of configurations can imply that a large number of customers will get better (in all objectives) configurations at a fixed price level. This is because a coarser discretization leads to an increased cost-effectiveness and, in the end, to lower product prices. In Appendix A we present an example illustrating this phenomenon. We claim that it is of advantage both for the company and for its customers to discretize the sets of customers and available configurations by lumping them together into (a suitable number of) representative groups.

What is in it for the company?

- Economics of scale

  An increased part volume leads to lower costs for, e.g., purchasing. A decreased number of variants leads to reduced costs not only for the total development, but also for, e.g., spare parts handling. By synergy effects, cost-effectiveness is created. If the company, also with fewer variants, manages to supply the market with technical solutions corresponding to the market requirements, the company will be able to offer competitive products in all quality segments.

- Better control of the market

  By discretization of the customers it is easier to control whether the different market segments are covered by the available configurations.

What is in it for the customers?

- Economics of scale

  In the long term, the economics-of-scale advantage for the company will lead to products at the same quality levels, but with lower product prices or, equivalently, products at the same price levels but with increased qualities.

- Better quality control

  With a larger number of variants and configurations, the product verification (the control of whether a product fulfils its requirements) will naturally be worse than with a small number. Computer simulations, rig tests, and proving ground tests are made on a relatively small number of distinct configurations, and the measured results are "extrapolated" to similar configurations. With fewer variants and configurations, the control will be better.

Discretization/aggregation should take place in three structures: the *product*, the *features* and the *operating environment*. Each is described in the following sections.

4.1 Discretization of the product and the decision spaces for the optimization problem

In Section 2.2 the current product structure is discussed. The structure is already now discretized; the variant families with their variants constitute a discretized version of the space of all possible trucks.

Let us now disregard the actual product structure at Volvo for a while and return to our configuration problem. In the space of all possible truck configurations, the company wish to produce the ones that are at least close to Pareto optimal. In addition, the company wants to use common parts for the configurations that are constructed. To enable a well-functioning product development process and to facilitate an efficient specification of the product itself, the product must be partitioned into groups or subsystems. This is what the present product structure describes, although it is not clear that the present structure is the best one. Therefore, we identify the different groups with the different variant families and let X^i be the set of all technically possible variants in variant family $i,\ i = 1, \ldots, m$. Further, we let the actual variants of today define the sets $\bar{X}^i \subseteq X^i,\ i = 1, \ldots, m$.

Based on the actual conditions, we define three alternative decision spaces, in which the configurations are located. The *configuration space*, or $\bar{X}$, consists of all configurations (feasible or not) that can be defined using the current variants, i.e., $\bar{X} = \bar{X}^1 \times \ldots \times \bar{X}^n$. By allowing new variants in each variant family we define the *design space*, or just X, as the Cartesian product of all possible variants (feasible or not) in the current variant families, i.e., $X = X^1 \times \ldots \times X^n$. Also, we define the set of all feasible trucks as the *full design space*, or X^{tot}. This set consists of just anything that can be called a truck, or even that can perform certain transport missions; only the imagination puts boundaries on X^{tot} (e.g., all of the competitors' trucks are members of this set.)

It is clear that $\bar{X} \subseteq X$ and that both $\bar{X}$ and X can be partitioned into feasible and infeasible parts, i.e.,

$$\bar{X} = \bar{X}_{\text{feas}} \cup \bar{X}_{\text{infeas}}, \quad \bar{X}_{\text{feas}} \cap \bar{X}_{\text{infeas}} = \emptyset$$

and

$$X = X_{\text{feas}} \cup X_{\text{infeas}}, \quad X_{\text{feas}} \cap X_{\text{infeas}} = \emptyset.$$

Also, the relations $\bar{X}_{\text{feas}} \subseteq X_{\text{feas}} \subseteq X^{\text{tot}}$ hold.

The principle behind the *product development strategy* used is to, instead of searching in X^{tot} when developing, search in X. Though, the basis of X, i.e., the variant families, is now and then altered such that the, for the moment, interesting part of X^{tot} becomes reachable. When using X as the decision space, the need for complicating restrictions (cf. Section 2.2, 6.3) is introduced. This is, however, necessary in order to obtain a manageable mass-market product. If the company's mission instead was to create a moon-lander—a one-time-product without combinatorial complexity—it would be natural to search within the corresponding set X^{tot} of all spaceships.

An important point is that in reality the development work lies somewhere in between the configuration space, the design space, and the full design space. There is a continuous product development, mostly of variants and now and then introduction of new variant families, but also many carry-over-effects where old technical solutions and concepts for different reasons are to be used also in the future. Two well-defined isolated formulations of the configuration problem can be defined, one in which the decision space is the configuration space $\bar{X}$ and one in which the decision space is the design space X. We denote the former problem as the *mode 1* problem, and the latter as the *mode 2* problem. The mode 1 problem is more of an operational type while mode 2 is more of a strategic or development type. One obvious difference between the two modes is that the restrictions/constraints are only formulated for mode 1 (they are unknown for mode 2). Furthermore, the mode 1 formulation is completely discrete whereas the mode 2 formulation contains continuous portions.

If we could start from an empty sheet, how should the product be structured? It seems natural to use a similar product structure as is used today, however with some alterations. One alternative would be to separate variant families corresponding to physical design alternatives from those corresponding to the operating environment. In addition, the number of redundant variant families (i.e., containing the same information as—or being a direct consequence of—another variant family) should be kept as low as possible. Further, the product structure should be (without specifying how) constructed in a way that it makes the objective functions as well-behaving as possible in the resulting decision variables (cf. Section 6.1 for a discussion about the quantification of the objectives).

There is also a balance between the number of variant families used and how well the interesting (near-Pareto optimal, cf. Section 5) region of the full design space X^{tot} is approximated by solutions in X, see Figure 12. One example could be if a company produces rectangular boxes with the objectives to maximize the volume and to minimize the surface area. Then only cubes are interesting (the Pareto optimal set consists of all cubes). Instead of using a structure with one variant family each for describing the length, the depth and the height of the box, respectively, a structure where the box is required to be a cube, and where only the diagonal of the cube is specified can be used to represent all Pareto optimal designs.

4.2 Aggregation of features

It is natural to utilize a number of different quality measures when evaluating a product. Thus, the quality of the truck must be partitioned into a number of "objective functions". An important question is how many objectives that are needed. When each customer chooses between different configurations, he/she might only use three or four objectives. For the whole collection of customers, however, many more objectives are needed.

It is important how the customers evaluate the quality of a configuration. If a customer cannot experience the difference between two technical solutions, then there is no point in separating these as different in the technical description. Harshly speaking, the cheapest solution within a group evaluated as incomparable should be the

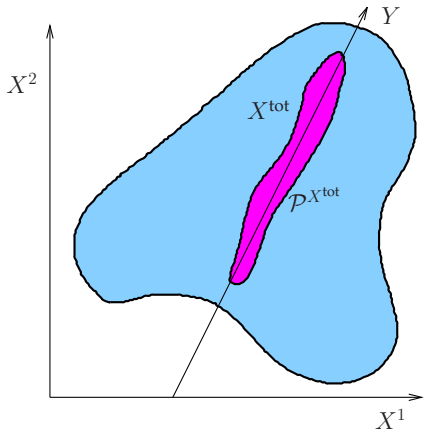


Figure 12: A rough illustration of a situation where by using one single variant family Y instead of two variant families X^1 and X^2, the Pareto optimal set of all possible solutions, $\mathcal{P}^{X^{\text{tot}}}$, can still be reasonably good represented.

one used. We denote the values of the quality measures that are recognized as different by the customers by *perceived feature steps*. These are however hard to model since they can vary between different customers.

The quality of a truck must be discretized or aggregated for a number of reasons. One is to get well-defined measures such that the global organization can compare results of different analyses. Another is to get a descriptive language to be used in the sales process. Moreover, management of the product range, making it match the different markets and transport missions, is much more straightforward if there are well-defined subqualities. Finally, a well-defined set of objectives is an essential requirement for being able to apply optimization to the configuration problem.

4.3 Discretization of environments

The components defining the operating environment are continuous sets, the elements of which are hard to measure. However, previous work has been done to create well-defined subsets of these sets in order to discretize the market. The reason for this work is the emergent possibility of differentiated products. Instead of designing against the worst-case solutions (e.g., such that the truck will hold even when driving on the worst roads imaginable), one creates an opportunity to design solutions adapted to the actual customers.

The parameters that determine in which subset a certain customer is located are called the GTA (Global Truck Application) parameters [14]. These parameters classify the transport mission, the operating environment, and the vehicle utilization, all split down into more well-measured parameters.

The resolution of the discretization should not be finer than which enables a classification of the customers without sources of errors dominating their differences. Neither is there any point in discretizing finer than the resolution in the feature levels in the technical solutions being perceivable by the customers.

The thesis [30] deals with topics related to discretizations of the operating environment. Parametric models are constructed describing the lateral loads acting on a truck for different types of customers in different operating environments. The aim is partly to create input signals for computer simulations for fatigue analyses adapted to the specific use of the truck.

Concluding this section, it is important to make discretizations to enable an efficiently solvable optimization formulation of the configuration problem. The discretizations of the features and of the environments can be seen as inputs to the configuration problem. Given these, we wish to find the best discretization of the product. The discretization of the product is divided into two steps, where in the first step the "dimensions of the grid", i.e., the variant families are defined, and then, when solving the problem, the elements in these dimensions, i.e., the variants, are found. Discretization is one of the main components of the configuration problem. Another one is *Pareto optimality*, which is treated in the next section.

5 Pareto optimality

A (single-objective) optimization problem is traditionally written as to

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}), && \text{(1a)}\\ &\text{subject to} \quad \mathbf{x} \in X, && \text{(1b)} \end{aligned}$$

where $X \subseteq \Re^n$ and $f : \Re^n \to \Re$. Note that the set X in this section is a general set of decision variables, not necessarily the same as the defined design space X in Section 4. The goal is to find an $\mathbf{x} \in X$ (the feasible set of points) that minimizes the objective function f over X. In multi-objective optimization there are, instead of just one objective, a vector $\mathbf{f} = \{f_1, \ldots, f_k\}$ of objective functions, $\mathbf{f} : \Re^n \to \Re^k$, that are to be minimized simultaneously. This problem is not well-defined in the common sense if there is a conflict between the objectives, i.e., if there exists no $\mathbf{x} \in X$ minimizing all $f_i,\ i = 1, \ldots, k$, over X. The reason for this is that vectors are not totally ordered, e.g., $(1, 1) < (2, 2)$ but how does one order the vectors $(1, 2)$ and $(2, 1)$? The goal in multi-objective optimization is to find the Pareto optimal subset $\mathcal{P} \subseteq X$, which is defined according to the following.

Definition 5.1 *Given a set X of feasible vectors and a set $\{f_1, \ldots, f_k\}$ of objective functions to be minimized, a vector $\mathbf{x}^* \in X$ is defined as* Pareto optimal *if there exists no vector $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \{1, \ldots, k\}$. An objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$ is called* Pareto optimal *if the corresponding vector $\mathbf{x}^*$ is Pareto optimal. The set of all Pareto optimal vectors $\mathbf{x}^* \in X$ is denoted $\mathcal{P} \subseteq X$.*

We define the *Pareto operator* $P: \Re^n \times \Omega \to \Re^n$ by $P(X, \mathbf{f}) = \mathcal{P}$, where Ω denotes the collection of all functions $\mathbf{f}: \Re^n \to \Re^k$. (For the special case when $\mathbf{f}$ is scalar-valued, i.e., when $\mathbf{f} = f$, $P(X, f)$ extracts all global optima in X given the objective function f.) We adopt the convention of extending the minimization operator by allowing it to apply to vectors and write a multi-objective optimization problem (MOOP) as that to

$$\begin{aligned} &\text{minimize} && \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\}, && \text{(2a)} \\ &\text{subject to} && \mathbf{x} \in X. && \text{(2b)} \end{aligned}$$

An illustration of Pareto optimality for a MOOP with $k = 2$ is shown in Figure 13. This picture motivates that the Pareto optimal set, because of the geometrical interpretation in the objective space, sometimes is called the *Pareto front*.

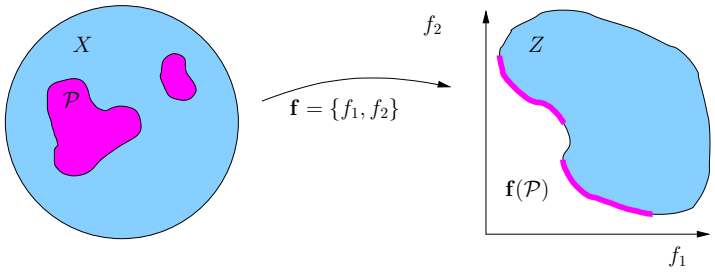


Figure 13: The decision space X with the Pareto optimal set $\mathcal{P} \subseteq X$, the objective functions $\mathbf{f} = \{f_1, f_2\}$, the objective space Z, and the image, $\mathbf{f}(\mathcal{P})$, of $\mathcal{P}$.

The Pareto optimal set for a problem with k objective functions is in general (if $n \geq k$), for a (continuous) MOOP a collection of hypersurfaces of dimension $k - 1$. The more objective functions there are, the computationally more intense it is to find $\mathcal{P}$. In the appended Paper 1—*Approximating the Pareto Optimal Set using a Reduced Set of Objective Functions*—a procedure is given for reducing the number of objectives such that the precision lost when approximating the set $\mathcal{P}$ in a certain sense is kept at a minimum. We refer to Paper 1 for more ideas on and references to problem reduction.

5.1 Classifications of multi-objective optimization problems

The purpose of this subsection is to describe different classes of multi-objective optimization problems and to locate the multi-objective problem we study within these.

Consider the multi-objective optimization problem (2). If X is a polyhedron and all objective functions $\{f_1, \ldots, f_k\}$ are linear then (2) is called a *multi-objective linear*

optimization problem, or MOLP in brief. If any of the objectives is non-linear or if X is not a polyhedron then (2) is called a *non-linear multi-objective optimization problem*, or MONP. The special characteristics of MOLP problems are studied in, e.g., [43]. The more general problem MONP is treated in the textbooks [15] and [35].

As well as for the single-objective optimization problem (1), convexity is a crucial concept for problems with multiple criteria. The problem (2) is *convex* if all $f_i,\ i = 1, \ldots, k$, are convex and if X is a convex set. Otherwise, it is a non-convex problem. Convex problems possess many nice characteristics. One of them is exploited in [32] where linear interpolations are made between Pareto optimal design points. Due to convexity, using corresponding interpolations of the objective points is guaranteed to underestimate the true objective values. This has made it possible to construct a useful tool for navigation within the Pareto optimal set.

If integrality constraints are part of the specification of X then (2) is called a *discrete* multi-objective optimization problem. A large part of the literature on discrete multi-objective problems focuses on the subset of multi-objective combinatorial problems (MOCP) which are multi-objective generalizations of traditional single-objective combinatorial optimization problems, e.g., traveling salesman, spanning tree, and knapsack problems. An extensive survey of MOCP is given in [16]. Often, the focus is on problems with only a few (often two) objectives (cf., e.g., [23]) and/or with an evident structure of the decision space X (cf., e.g., [27, 17]). One special feature of problems with two objectives (bi-criteria problems) that can be exploited in solution algorithms is that a total ordering of the Pareto optimal set can be defined.

One special type of discrete variables is the *categorical* type (cf. Section 6.2), representing variables that have no natural ordering (in the decision space). By considering this type of variables, a distinction between numerical and categorical problems is defined.

One can also separate between problems possessing simulation-based objective functions and problems in which the objective functions have analytical expressions.

When solving MOOP's the aim is not always the same. Often one wish is to find a single solution $\mathbf{x}^* \in X$ that is optimal for a certain decision maker (DM). Thus it is really a (hidden) single-objective problem, with the DM's utility (or value) function $u : \ \Re^n \rightarrow \Re$ measuring his/her overall preference of the design points as the single objective function[2]. However, the utility function is often hard to specify, and a common belief is that it is easier to decide on the components of the utility and to formulate a multi-objective problem using these components as objective functions. In other cases one searches for the whole Pareto optimal set $\mathcal{P}$.

The underlying multi-objective optimization problem in the configuration problem studied in this thesis is non-linear with simulation-based objective functions that are not convex in general. Some of the variables are discrete and many of the discrete variables are categorical. There is no obvious structure of the decision space. It is a true multi-objective problem, where we are interested in the entire $\mathcal{P}$.

[2]The utility function is sometimes defined as a function $u : \ \Re^k \rightarrow \Re$ of the objective components $f_i(\mathbf{x}),\ i = 1, \ldots, k$. We let it be a function of the design itself to handle the problem that otherwise arises if the assumption of identical objectives is invalid.

5.2 Solution methods for the configuration problem

The suitable solution method for a MONP is of course heavily depending on the aim of solving the problem at hand.

If the aim is to search for *one* optimal vector $\mathbf{x}^* \in X$, the solution methods for MONP can be divided into four major classes (originally presented in [25]):

1) *no-preference* methods

2) *a priori* methods

3) *a posteriori* methods

4) *interactive* methods

For all of these four classes [35] gives many examples of solution methods. Briefly, in class 1 the utility function of the DM is roughly guessed and a single objective problem is solved. The optimal solution is presented for the DM who can either accept or reject (whence a new updated guess for the utility function is made). The a priori methods of class 2 are similar to those in class 1; the difference is that the DM is explicitly asked to give an accurate mathematical expression of his/her utility function. The a posteriori methods of class 3 and the interactive methods of class 4 are fundamentally different from the former ones. Here, the fact that it is hard (or even impossible) for the DM to express his/her utility function is accepted. Thus, in the a posteriori methods the goal is to produce the whole Pareto optimal set, or a sufficiently good approximation of it, and then present it for the DM who then selects his/her most preferred solution. A posteriori methods might be very time consuming and it is possible that a lot of time is spent on describing a part of $\mathcal{P}$ that is not interesting for the DM. This is the reason for developing interactive methods. Here, the DM is participating in the whole solution process, making it possible for him/her to specify and correct his/her preferences along the process when he/she gets a feeling for the opportunities and the limitations of the problem. Many examples of interactive methods can be found in [35]. A positive aspect of the interactive methods is that focus is only on the interesting part of the (a priori unknown) set $\mathcal{P}$. A negative one is, however, that interactive methods might be very time consuming for the DM if the objective functions are expensive to compute.

Concluding the classes of methods above, it is only the a posteriori methods that search the entire $\mathcal{P}$, and thus is applicable to our problem. However, parts of the ideas used in interactive methods could be exploited for a problem reduction, such as using engineering expertise and experience to decide on which solutions that cannot be interesting to any customers.

Examples of a posteriori methods are the traditional weighting method (where a weighted sum of the objectives is minimized), the ϵ-constraint method (where one of the objectives is minimized while the other objectives are constrained from above), and the method of weighted metrics, or compromise programming (minimization of a weighted L^p-distance from the solution vector to a reference point). Another method in the a posteriori class is the normal-boundary intersection method [10].

One class of solution methods that is sometimes overlooked is the class consisting of evolutionary methods. These are of heuristic nature and therefore often lack guarantee measures on optimality. However, they are easy to implement, they do not require many assumptions on the problem (e.g., convexity, differentiability, or continuity), and they have shown success in many real-world applications. A thorough overview of different evolutionary methods can be found in the textbook [11]. Two of the evolutionary methods most used are the Neighborhood cultivation algorithm (NCGA) [45] and the Nondominated sorting genetic algorithm II (NSGA-II) [12]. This type of algorithms also belongs to class 3 above.

One could think that the *weighting method*—the first intuitive method to use when dealing with a MONP—is suitable for the configuration problem. In the weighting method, a sequence of single-objective problems of the form

$$\text{minimize} \quad \sum_{i=1}^{k} w_i^\ell f_i(\mathbf{x}), \tag{3a}$$

$$\text{subject to} \quad \mathbf{x} \in X, \tag{3b}$$

is solved with different weight vectors $\mathbf{w}^\ell \in W = \{\mathbf{w} \in \Re^k | \ \sum_{i=1}^k w_i = 1, \ w_i \geq 0, \ i = 1, \ldots, k\}$, using a single-objective optimization method. Denote the set of solutions to (3) for all $\mathbf{w} \in W$ by $\mathcal{P}^S$, the *supported* Pareto optimal set. Using the Pareto operator (defined on page 19, and here applied to scalar objectives) we can write $\mathcal{P}^S = \{\mathbf{x}^* \in \Re^n \mid \mathbf{x}^* = P(X, \sum_{i=1}^k w_i^\ell f_i(\mathbf{x})), \mathbf{w}^\ell \in W\}$. It is easily shown that $\mathcal{P}^S \subseteq \mathcal{P}$. One major drawback of the weighting method is, however, that it can be guaranteed that $\mathcal{P}^S = \mathcal{P}$ only when the MONP is convex. Figure 14 illustrates the supported and the non-supported parts of the Pareto optimal set of a non-convex MONP.

Even if one would argue that solutions to (3) only for $\mathbf{w} \in W$ are the interesting solutions to (2)[3] or if we really have a convex problem at hand, whence $\mathcal{P}^S \subseteq \mathcal{P}$ and the solution sets coincide, the weighting method still have drawbacks. One is that the interpretation of the weights used is unclear (the problem of comparing apples and oranges). Another is that the mapping from the weights to the solutions to (3), i.e., $\Re^k \ni \mathbf{w} \mapsto P(X, \sum_{i=1}^k w_i f_i(\mathbf{x})) \in \Re^n$, is non-linear and strongly depending on the properties of the actual functions involved. This is nicely shown by an example in [9] where (3) is solved with equidistant weight vectors leading to a highly non-uniform distribution of points in $\mathcal{P}^S$. A third disadvantage comes from the fact that it might be hard (at least for engineering applications) to assign absolute numbers to the objectives (qualities) for different design solutions. That is, for each objective, it might be possible to order the different solutions but impossible to state how much better one is than another and what the underlying functions look like. Two collections of

[3]The most preferred solution of a MOOP for a certain DM is the decision point maximizing his/her utility function, i.e., $\mathbf{x}^* = \arg\max_{\mathbf{x} \in X} u(\mathbf{x})$. All objective points that are evaluated as equally desirable by the decision maker are said to be situated on the same *indifference curve*. These curves represent the marginal rate of substitution, or indifference trade-off, between the objectives locally around the most preferred point. If the slopes of the indifference curves, for all DM's, are constant in the whole of the objective space for every pair of objectives, then only supported Pareto optimal solutions can be preferred.

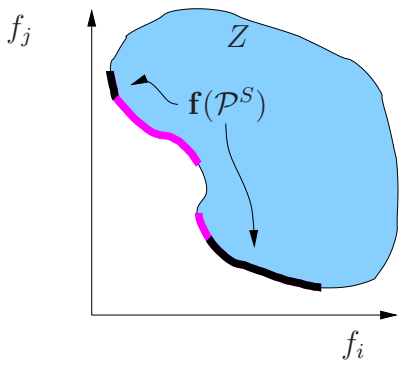


Figure 14: The image of the supported Pareto optimal set $\mathcal{P}^S$ is marked in black. The dark gray region is the image of the non-supported Pareto optimal set, i.e., $\mathcal{P} \setminus \mathcal{P}^S$.

functions ordering X identically (i.e., collections $\{f_1, \ldots, f_k\}$ and $\{g_1, \ldots, g_k\}$ such that $f_i(\mathbf{x}^1) < f_i(\mathbf{x}^2) \Leftrightarrow g_i(\mathbf{x}^1) < g_i(\mathbf{x}^2), \forall i \in \{1, \ldots, k\}, \forall \mathbf{x}^1, \mathbf{x}^2 \in X$) lead to the same Pareto optimal set, but there might be great differences in the corresponding supported sets $\mathcal{P}^S$. The following example illustrates this behavior.

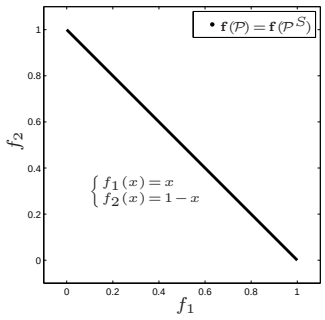
Example 5.2 *Assume that our original MONP is defined as that to*

$$\begin{aligned} &\text{minimize} && \{x, 1-x\}, && \text{(4a)} \\ &\text{subject to} && x \in X = [0,1]. && \text{(4b)} \end{aligned}$$

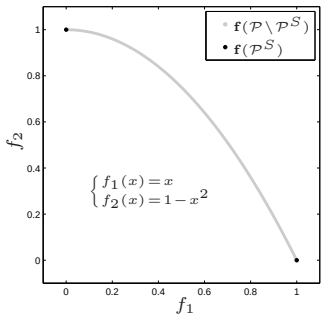
The corresponding Pareto optimal set is $\mathcal{P}_1 = X$. Consider next the problem to

$$\begin{aligned} &\text{minimize} && \left\{x, 1-x^2\right\}, && \text{(5a)} \\ &\text{subject to} && x \in X = [0,1]. && \text{(5b)} \end{aligned}$$

Clearly, the objectives in (5) individually order the decision space as in the problem (4). Since $\mathcal{P}_2 = X$, the problems (4) and (5) are equivalent in the sense that $\mathcal{P}_1 = \mathcal{P}_2$. However, (due to the non-convexity of the second objective in (5)) the supported subsets of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, are very different: $\mathcal{P}_1^S = \mathcal{P}_1 = [0,1]$ but $\mathcal{P}_2^S = \{0,1\}$, as illustrated in Figure 15.



(a) The supported subset of the Pareto optimal set is the whole Pareto optimal set



(b) The supported subset of the Pareto set consists of two single points.

Figure 15: The supported set $\mathcal{P}^S \subseteq \mathcal{P}$ is not only defined by the ordering of the decision space by each objective function, but also on the absolute function values.

Within the configuration problem there are instances of (2) of strongly varying nature. The reason for this is that the configuration problem can be viewed on different scales, with decision variables of totally different types. On the largest scale, the decision variables represent the variants in the variant families. On the smallest scale—if one can find (relatively) isolated subsystems of the truck (such that the design parameters do not significantly affect the rest of the truck)—the decision variables can represent continuous design parameters such as, e.g., spring constants or lengths.

There are no methods or algorithms that are efficient for all types of multi-objective optimization problems; it is important to design solution algorithms for the actual problems at hand. On the large scale, evolutionary algorithms in combination with metamodeling (where approximate models are constructed and used instead of the true models, cf. Section 8.1) is one option. Another is to solve sequences of single-objective optimization problems using pattern search algorithms [3]. This, however, requires a definition of local neighborhoods in the decision space, which in turn requires a well-defined product structure. On the small scale, the optimization problems are numeric and it should be possible to use some of the a posteriori algorithms, such as the ϵ-constraint or the compromise programming methods, mentioned above.

5.3 Approximations of the Pareto optimal set

In practical applications usually one cannot expect to find the whole Pareto optimal set $\mathcal{P}$. For general non-linear non-convex problems it neither possible to find $\mathcal{P}$ in finite time without some further assumptions on the problem (such as Lipschitz continuity with known constants). Approximating methods, i.e., methods for creating good approximations of $\mathcal{P}$, are developed for this reason. A survey of such methods is contained in [40]. In principle, the methods are from the a posteriori class, where

the different parameters of the algorithms are controlled systematically. When an approximating method is used, it is important to get some sort of quality measure for the resulting set of solutions. Note that the issue of measurement is simple for single-objective problems, where the objective values for two solutions can be compared directly. From the literature, it is clear that there is no standard measure for evaluating an approximate Pareto optimal set. The subject is discussed and propositions are given in, e.g., [38, 8, 44, 41, 6, 46].

When discussing approximate sets, there is need for two kinds of measures. First, a measure describing the distance between two points is needed; the *point measure*. This measure is then utilized in a second measure, the *set measure*, describing the distance between two sets of points.

The measures should ideally be defined both in the decision space and in the objective space. Since the objective space contains only numeric vectors, it is easier to define the measures in that space. However, from a practical point of view the distances in the decision space are also important. For the current situation at Volvo, the decision space defined by the product structure is heterogeneous with variants and variant families representing totally different entities. Moreover, there is no clear understanding of the hierarchies in the product structure which makes it hard to define a suitable measure in the decision space.

In the objective space we notice that if the objectives measure completely different entities (such as, e.g., safety and cost, or stress and volume) then it is not obvious how to scale the objectives such that the units (e.g., if measuring cost in Euros or cents) does not affect the result. One way to handle this [35] is to normalize the objectives such that their values vary between 0 and 1 over $\mathcal{P}$.

For the application we study, the point measure should reasonably be defined asymmetrically. One natural choice for comparing a point $\mathbf{x}^2 \in X$ with another point $\mathbf{x}^1 \in X$ is to use some norm of the components in $\mathbf{f}(\mathbf{x}^2)$ being larger than the corresponding components in $\mathbf{f}(\mathbf{x}^1)$, i.e.,

$$c(\mathbf{x}^1, \mathbf{x}^2) = \rho(\max\left\{\mathbf{f}(\mathbf{x}^2) - \mathbf{f}(\mathbf{x}^1), 0\right\}), \tag{6}$$

where the $\max\{\cdot,\cdot\}$ is defined element-wise and $\rho(\cdot)$ is some suitable norm. The reason for only selecting the components being larger is that we do not want to punish objectives that are improved. The Figure 16 illustrates the indifference curves, i.e., the sets of points that are evaluated as equally good, induced by the point measure (6) where $\rho(\cdot)$ is chosen as the L^2-norm defined by $||\mathbf{u}||_2 = \sqrt{u_1^2 + \ldots + u_n^2}$. The distance measure (6) is open to certain objections. As can be seen in the figure, one does not earn anything if an objective value is improved if it already is at least as good as for the reference point. The distance measure should be developed further such that it better adapts to real customer behavior.

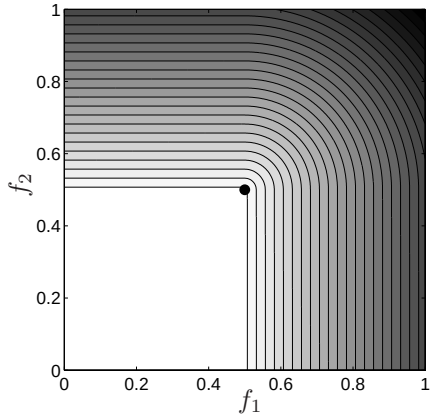


Figure 16: Indifference curves when comparing $\mathbf{f}(\mathbf{x}^1) = (0.5, 0.5)$ to all points $\mathbf{f}(\mathbf{x}) \in Z = [0,1]^2$ induced by the point measure $c(\mathbf{x}^1, \mathbf{x}) = \rho(\max\{\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}^1), 0\})$ with $\rho(\cdot)$ being the L^2-norm. A darker color means a larger distance. White color corresponds to zero distance.

The set measure can be either symmetrically or asymmetrically defined depending on the aim of the comparison. If we want to compare the Pareto optimal sets for two problems (e.g., if one problem is a computationally easier but an approximate version of the other), then a symmetric measure might be the right choice. Such a measure is defined and used in the appended Paper 1, *Approximating the Pareto Optimal Set using a Reduced Set of Objective Functions*. Here, we measure the distance between the sets as the largest (point measure) distance between any of the points in each of the sets to their respective nearest point in the other set.

If instead we want to evaluate how good representation of the Pareto optimal set is achieved by an approximate method, then an asymmetric measure might be a better choice. This is common, e.g., when evaluating new evolutionary algorithms for MONP's. Such an asymmetric measure is also used in the appended Paper 2, *Multiobjective Design of a Combinatorial Structure*. Here, a good solution set to an underlying MONP is searched under the constraint that the solutions have certain similarities in the decision space. Quality measures of the solution set, comparing it with the Pareto optimal set for the corresponding problem where these constraints are relaxed, are used. When searching for a good representation $\mathcal{R}$ of a Pareto optimal set $\mathcal{P}$ one wants the points in $\mathcal{R}$ to be (at least) near-Pareto optimal and also well distributed along the Pareto optimal front. One (non-unique) measure capturing both these wishes is, with $c(\cdot,\cdot)$ being the point measure,

$$d(\mathcal{R}, \mathcal{P}) = \max_{\mathbf{x}^1 \in \mathcal{P}} \min_{\mathbf{x}^2 \in \mathcal{R}} c(\mathbf{x}^1, \mathbf{x}^2), \tag{7}$$

illustrated in Figure 17. This is the *Dist2* measure from [8].

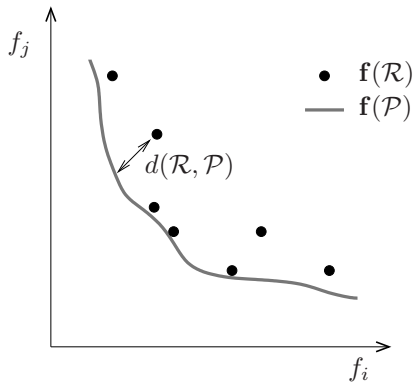


Figure 17: The distance between $\mathcal{R}$ and $\mathcal{P}$ defined by (7) is illustrated with an arrow.

Furthermore, for our application in which we search for the entire set $\mathcal{P}$ but are content with a discrete subset of $\mathcal{P}$, it seems reasonable to use a point measure that depends on the region of Z where the points are. Sales volumes and/or other strategical issues could be incorporated in the measures, e.g., by using region-dependent weights. By using such, it would be possible to steer against approximations that are focusing on a certain set of designs, e.g. trucks with a high profitability or trucks whose features are validated carefully with simulations and tests.

6 Quantification of the problem

Optimization in industry is an iterative process where, in each iteration, a model of the true problem is formulated and solved. Then the outcome should be evaluated by experts, using experience and "implicit" knowledge (which is known but cannot be explained with rational statements) with the probable conclusion that something is missing in the model. The model is then adjusted and a new iteration takes place. The major benefits of using optimization for industrial problems are:

- that it yields a better understanding of what is affecting the properties of a design or a system[4], and
- that it provides good results in short time that an expert may modify locally (i.e., change to solutions that lie near in the decision space).

That is, multi-objective optimization in industry has two applications; to learn the problem at hand, and to create basic data for decision making.

We now return to the configuration problem that Volvo is heading and regard this problem as a multi-objective optimization problem. This section quantifies the

[4]For non-linear problems there can be intricate dependencies within the design that are hard to get a clear understanding of generally. Optimization together with sensitivity studies around near-optimal points can help to illuminate the dependencies in interesting regions.

problem and places it into an optimization context. The important ingredients of an optimization problems are the *objective(s)* which measure(s) the quality(ies) of a solution and is (are) to be minimized (or maximized), the *variables* which are to be selected within some given sets, and the *constraints* which corresponds to design or feature requirements that has to be fulfilled for a solution to be acceptable. The aim of this section is to describe these three components for the configuration problem.

6.1 The objective functions

Engineers often use a combination of experience (or, rules-of-thumb), computer simulations of different levels-of-detail together with rig testing, proving ground testing, and actual field verification to decide on which designs are good and which are not. For an optimization approach to be valid, we must assume that there are real-valued functions (possibly outputs from black-box simulations) assigning quality measures to each single truck configuration. This requirement poses a number of practical problems. First, it is hard to gather all the knowledge and transfer it into functions. Another problem is that it is hard to put scalar values on output from, e.g., computer simulations or proving ground tests. For example, it is difficult to assign a fair scalar quality measure to a spectrum (representing, e.g., the acceleration at some point in the cabin over a frequency interval). The author of the thesis [20] devotes his whole work to investigate how drivers perceives (different combinations of) sounds in truck cabins. It is hardly an easy task to define an objective to be minimized measuring the driver's sound experience as a function of the design variables.

Further still, it is difficult to keep the total computer work at a reasonable level, managing the different levels-of-detail of the simulations. Large-scale problems can seldom be solved exactly, and to get a good solution in the end, the accuracy of each model must be weighed against the simulation time required. It is not easy to decide on a suitable balance. Of importance is that the final decisions based on the models developed in this work are good enough compared to if the models described the problem in reality perfectly and, furthermore, if they were solved exactly. Important issues for simulation-based optimization are described in Section 8.1.

The Figures 18 and 19 illustrate models with different levels-of-detail. Figure 20 shows output of an actual simulation, measuring vibrations in the cabin induced by the engine. To incorporate such a simulation into an optimization framework, it is required to set a scalar value (or possibly a number of scalar values) on the simulation outcome, which is not a straightforward task.

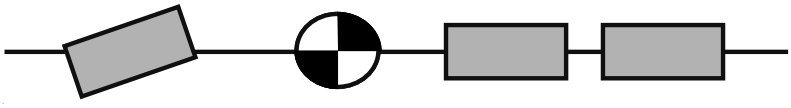


Figure 18: A simple model of a truck.

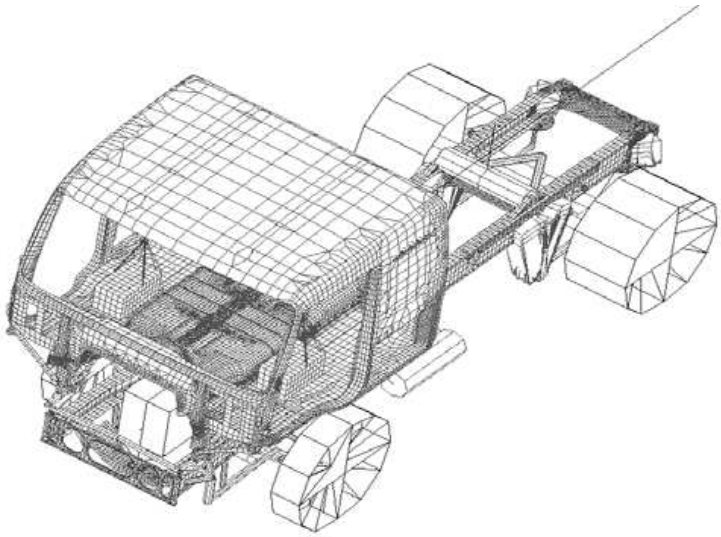


Figure 19: A detailed model of a truck.

Within our approach to reach the ultimate goal of making the best set of configurations (i.e., configurations matching the customer demands and designed in a cost-efficient way) available we create a model of the real problem and solve this model (as well as possible). In our case, we do exactly as described in Section 1.2, that is assume that the quality perceived by each customer can be separated into a number of objective functions (see also Section 4.2) which are identical for all customers, but whose combination can be appreciated differently. Consequently, we need functions measuring the concepts on the axes of Figure 3. For some of the concepts there exist mature measures; e.g., for *Driver environment* there are standard ways of measuring the ride comfort in the cabin based on accelerations in certain frequencies. For other measures, such as *Product design*, there is no know suitable function. Ideally, there should be functions available for all possible quality measures of the truck to be chosen as $\{f_1, \ldots, f_k\}$. In reality though, the process must be to start with a small set of reasonable functions, and then modify and complement this set based on the result (the Pareto optimal set of configurations) from using these functions. That is, to use engineering experience to decide on a good problem formulation, in the same way as experience is used to decide which simulations that should be used for the evaluation of new designs.

It is important to keep the true purpose of the actual problem in mind such that the consequences of the explicit or implicit (by the approach) assumptions made are not overlooked. The validity of the multi-objective optimization approach—that all customers measure quality in the same way—is in no way obvious. Furthermore, the objective functions used will be more or less inexact, both their modeling and the

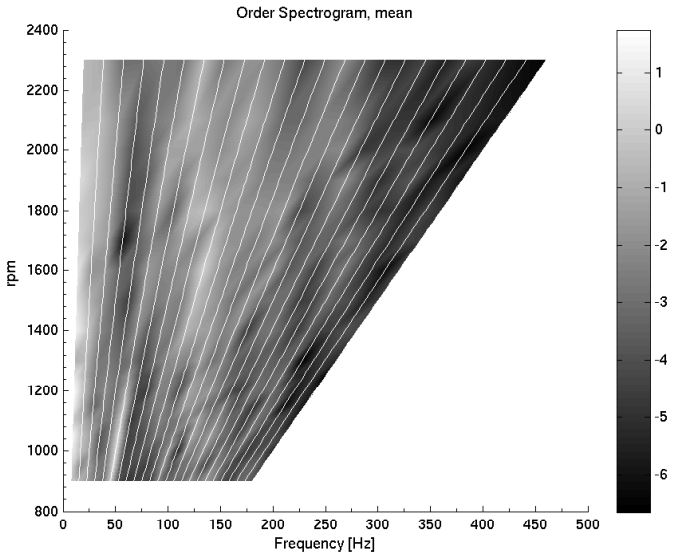


Figure 20: An outcome of an actual simulation regarding engine induced vibrations in the cabin over a frequency interval for different rotation velocities in the engine. It is not straightforward to translate this result to a (number of) scalar value(s).

computing simulations. We use metamodeling to get reasonable computation times. The metamodels will be constructed in such a way that the decisions they lead to are as similar as possible to those that would have been the result of using the original functions. It is not yet clear how to translate this target into analytical terms. To sum up, we wish to create a robust system, where *robust* means that by using the system the customers will choose configurations that are (near-)optimal to them, and that these configurations are constructed as cost-efficiently as possible, even if there are uncertainties and/or approximations in the model and the computations. Also, we wish to use a robust design approach in its traditional meaning [37] where elements such as manufacturing variations, component deterioration, and environmental variation over time are considered.

6.2 Variables in the configuration problem

Many optimization models contain only continuous decision variables, i.e., variables whose values lie in intervals on the real line. Other models, e.g., for problems involving fixed costs or different routing problems, require the use of discrete or integer variables, representing on/off decisions or indivisible quantities. Large engineering design optimization problems often involve a certain type of discrete decision variables that cannot be naturally ordered. These variables are denoted *categorical* [3].

These variables typically represent choices from some unordered list, such as variables corresponding to totally different technical solutions or to various materials with different properties. This is the case also in our configuration problem, where many of the variants in the variant families represent technical solutions that are incomparable, e.g., the family representing whether air springs, conventional leaf springs, or multi-leaf springs are used in the rear axle installation (cf. Figure 5). When an optimization problem contains such variables, the standard methodology for solving discrete problems using relaxation of the integrality constraints is no longer usable.

In [3], a method for solving problems modeled with categorical variables is presented. Here, the notion of local optimality is adapted to the categorical variables by the construction of a distance measure between design solutions and by the definition of a user-specified local neighborhood based on this measure. Then a pattern search algorithm is used leading to a local minimum with respect to the neighborhood defined. Another approach to solve mixed-variable problems, i.e., problems containing categorical variables, is to use evolutionary algorithms [42, 1]. Also here, a measure is required to specify the distances between solution points.

An observation to be made is that the scale on which the configuration problem is viewed affects the distribution of the types of variables that appears in the resulting optimization problem. By looking at the "clean" problem in the configuration space $\bar{X}$ (see Section 4.1) almost every variable is categorical, since almost every variant affects more than one physical parameter which in turn affects the vehicle behavior. If we instead look at the problem in the design space X, then by looking at a finer scale more of the variables become continuous (or at least orderable discrete). Looking at the finest possible scale, letting all parameters in the simulations for the objective values be decision variables, the problem will not be mixed. However, a large number of constraints would have to be added to make sure that the feasible solutions correspond to trucks that can be defined using the variant families.

6.3 Formulation of the restrictions/constraints

One difficulty of viewing the problem comprehensively on a large scale, where the variables represent variants in the variant families, is that many of the variables are categorical, that is, not well-ordered. A natural way of formulating the restrictions (cf. Section 2.2) is to use binary variables, assuming that the number of possible variants is finite:

$$x_{j\ell} = \begin{cases} 1, & \text{if variant } \ell \text{ in family } j \text{ is chosen,} \\ 0, & \text{otherwise.} \end{cases}$$

By letting R denote a list of variants in variant families that are not allowed to be used all of them together, the packing type constraint (8) excludes all designs not fulfilling the restriction, since it is not possible to have $x_{j\ell} = 1,\ \forall (j,\ell) \in R$.

$$\sum_{(j,\ell)\in R} x_{j\ell} \leq |R| - 1. \tag{8}$$

However, due to the categorical variables, the objective functions do not possess any nice properties with respect to the variables when applying this formulation. The reason for this is that when the scale is increased to a level where categorical variables must be used, then a physical interpretation of the variables is not longer possible.

7 Clustering the configurations

The scientific area of *clustering*, or *data clustering*, concerns the classification of a set of objects into more or less homogeneous groups, such that the objects assigned to the same group are similar according to some suitable distance measure. The area is widespread with applications within a large number of domains, such as pattern recognition, image analysis, marketing, and machine learning. The textbook [19] gives an overview of the subject.

It is clear from Section 4 that our application fits into a clustering context, where all customer wishes (i.e., wanted configurations) are to be partitioned into a number of groups, each corresponding to a certain truck configuration. It is clear that Volvo wishes to produce trucks that are good in the sense of Pareto optimality, however the Pareto optimal set may be very large and it is not reasonable to offer all of these configurations to the customers. Instead, the wish is to reduce the offer to a limited discrete set of technical solutions. But our application possesses special characteristics leading to aggravating circumstances. First, due to the combinatorial complexity we cannot cluster the designs using only using information from the objective space Z. Second, since the quality of the configurations is the central property, neither it is possible to cluster the solutions in a good way using information from the design space X only. Clustering using a combination of the two is necessary.

Below, in Section 7.1, we briefly present some traditional clustering techniques which may be adapted to our application. In Section 7.2 we present a new methodology which clusters the design solutions implicitly using optimization. This procedure by construction takes care of the difficulties of treating the two spaces X and Z simultaneously.

7.1 Explicit clustering

Assuming that all customers possess objective functions ordering the configurations identically, then without considering the product structure, the only configuration designs that are interesting to produce are the ones in the Pareto optimal set $\mathcal{P}$. However, $\mathcal{P}$ (or $\mathbf{f}(\mathcal{P})$ which is the interesting set since the quality of a configuration is measured in the objective space), is a very large set and it is not motivated to make all these configurations available for the customers. Instead, $\mathbf{f}(\mathcal{P})$ should be clustered to a limited set Z_D of points each corresponding to a point in the design space X such that to every customer wish, i.e., a point $\mathbf{z}^* \in \mathbf{f}(\mathcal{P})$ (corresponding to a point $\mathbf{x}^* \in X$), there is an available configuration $\mathbf{x} \in X_D$ (where X_D is the available set of configurations, i.e., $\mathbf{f}(X_D) = Z_D$) at some small enough distance $c(\mathbf{x}^*, \mathbf{x})$. Such

a clustering is illustrated in Figure 21. This clustering lacks an important property which is crucial for our application; it does not consider how X_D is structured in X.

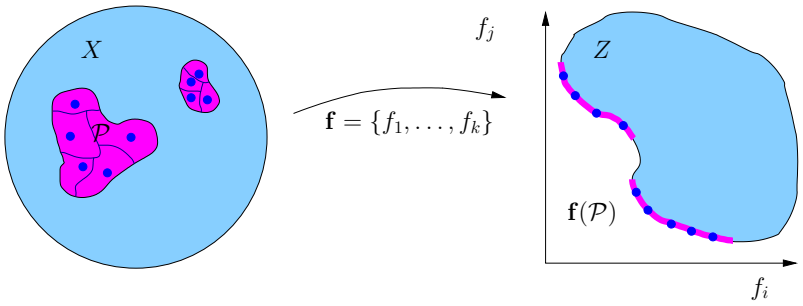


Figure 21: A clustering of the Pareto optimal set $\mathcal{P}$ into the set X_D, such that $\mathbf{f}(X_D)$ is "evenly" distributed over $\mathbf{f}(\mathcal{P})$.

The above description can be denoted as explicit clustering in the Z space. The clustering is done of $\mathbf{f}(\mathcal{P})$ which is a numeric set. This is fairly simple since there are natural distance measures between pairs of numerical points (e.g., the Euclidean distance). There are, however, a large variation of clustering algorithms that could be used. In [19] many of these are described, examples of well-known algorithms are, e.g., *k-means clustering*, and *single-link-clustering*. What is common to the explicit algorithms is that they try to partition the set into subsets, where the distances from the points in each subset to its representative solution are small and where the distances between points in different subsets are large. The way towards this goal and the definitions of distances are varying depending on the actual clustering technique. There are *hierarchical* techniques that iteratively updates the clusters (e.g. single-link clustering), and there are *partitional* techniques that establishes all clusters at once (e.g. k-means clustering).

It is also possible to define an explicit clustering in the X space, possibly with the aim of partitioning $\mathcal{P} \subseteq X$ into groups. However, then it is not clear how to control that the set of representative points $Z_D = \mathbf{f}(X_D)$ is a good representation of $\mathbf{f}(\mathcal{P})$ (cf. Section 5.3). But even more serious is that it is hard to control the structure of X_D, such that the number of contained variants are of reasonable size. A third difficulty is that it is needed to define distance measures between points in the X space, which is hard when it consists of categorical variables.

Possibly, explicit clustering in the composition of the two sets, i.e., in $X \times Z$, could be used. How such a clustering should be defined is however not known to us. Instead, we propose what we call *Implicit clustering*, in which we solve an optimization problem whose solution represents a clustering both with good structure in X and representing a good approximation of $\mathbf{f}(\mathcal{P}) \in Z$. Further, the problem of defining a

suitable distance measure in the design space is not there.

7.2 Implicit clustering

Presupposing the existing product structure with X^j, being the set of possible variants in variant family $j = 1, \ldots, n$, we can write $X = X^1 \times X^2 \times \cdots \times X^n$ (cf. Section 4.1). We have that $X = X_{\text{feas}} \cup X_{\text{infeas}}$, and $X_{\text{feas}} \cap X_{\text{infeas}} = \emptyset$, where X_{feas} is the subset of X fulfilling all of the restrictions. By restricting $|Z_D| \leq N$, explicit clustering of $\mathbf{f}(\mathcal{P}) \in Z$ may lead to up to N variants in each variant family, since there is no control of the resulting structure of X_D. To get around this problem we instead propose a clustering of the whole objective space Z, not just $\mathbf{f}(\mathcal{P})$, under the constraint that the configurations chosen in the design space lie in a "grid" where the size restriction lies on each variant family, i.e., $|X^j| \leq m_j,\ j = 1, \ldots, n$. The objective for the "implicit" clustering is now to select a limited set $X_D^j \subseteq X^j$, $j = 1, \ldots, n$, such that the "quality" of the product set $X_D = X_D^1 \times \cdots \times X_D^n$ is as good as possible. Mathematically, this is expressed as to minimize the set measure (7) between $\mathcal{P}$ and $P(X_D, \mathbf{f})$ (the non-dominated part of the resulting configurations). The implicit clustering is studied in the appended Paper 2, *"Multi-Objective Design of a Combinatorial Structure"*. Figure 22 illustrates a clustering where the available configurations belong to the product set of the variants chosen. An arrow in the bottom part of the objective space in the picture illustrates the distance for one such vector $\mathbf{x}^*$ ($\mathbf{f}(\mathbf{x}^*)$ is marked as a white dot) to its nearest vector $\mathbf{x} \in P(X_D, \mathbf{f})$.

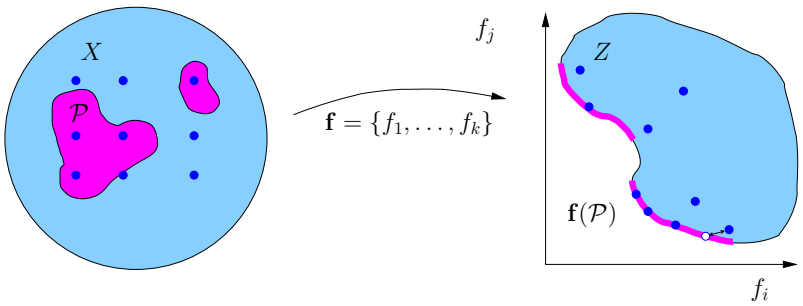


Figure 22: An illustration of an "implicit" clustering of the Pareto optimal set where X_D is required to be a product set. The dark dots represent X_D and Z_D in the design and the objective space, respectively. The arrow between a white dot and one of the elements of Z_D represents the distance from one Pareto optimal vector to its nearest available solution.

It is clear that we, using implicit clustering, do not require that $X_D \subseteq \mathcal{P}$, i.e., that the clustered objectives are Pareto optimal. Often, e.g., in industrial applications, or

when there are large uncertainties in the models, this is not a necessary requirement.

8 The scientific area — a multi-domain

A thesis like this, driven by a concrete application, naturally has to make use of theory from a number of separate but interconnected scientific areas or domains. One difficulty is to decide on how deep to go into each domain. It is necessary to gain sufficient knowledge in each domain in order to decide which parts in the respective domains to combine such that the resulting path leads towards the goal.

The most obvious domains needed for this work are *Multi-objective optimization* (cf. Sections 5 and 6.1), *Simulation-based optimization* (cf. Sections 5 and 8.1) and possibly also *Clustering analysis* (cf. Section 7), *Complex systems* (cf. Section 3), *Decision theory*, *Combinatorial optimization* and *Constraint programming*. However, also areas like *Behavioral sciences* (e.g., to analyze driver behavior, apprehension of qualities, perceived feature steps etc.) and *Mathematical statistics* (to analyze uncertainties in the models and data) are needed (but not considered in this thesis).

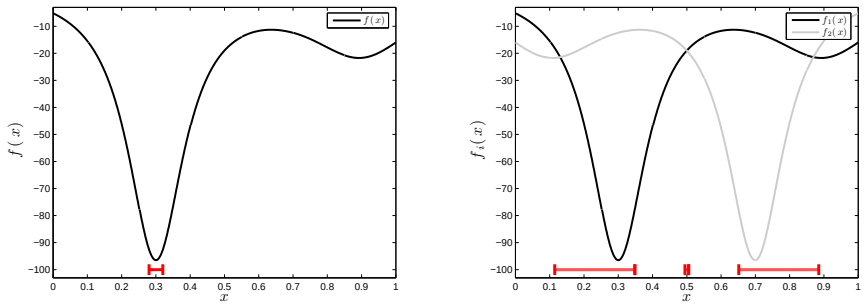
An underlying assumption defining an important delimitation of this work is that the actual configuration problem constitutes a well-defined optimization problem with multiple objectives. Such a problem formulation requires that all customers use the same set of objective functions to evaluate the quality of a certain configuration, although different customers assign different valuations of the individual objectives. If this is not the case, then Pareto optimality makes not much sense. Within this delimitation, we already indirectly assumed that to each quality there is some function associating a certain real number to each configuration, thus creating a total ordering of the configurations with respect to that quality.

In the following sections, we give a quick overview of the areas not yet described that are the most important ones for our application.

8.1 Simulation-based optimization

Simulation-based optimization is a generic term for optimization in which the objective function(s) and/or constraint function(s) are in some sense "expensive" to evaluate. When solving such problems the number of function calls should be kept to a minimum. The functions could be, e.g., outcomes of computationally intense simulations or practical experiments. Often there are no analytical derivatives available from the simulations. One standard approach when solving simulation-based problems is to sample the expensive functions iteratively and to create cheap approximations, or *surrogate models*, of them. A review of such methods, often called *Response surface methods*, can be found, e.g., in [28]. The surrogate models are often linear or quadratic approximations (see e.g. [39]), or some sort of interpolations, e.g., Kriging [28] or Radial basis functions [22], of the actual simulated functions. Other approaches use neural networks [2] or combine models of different fidelities [4]. Many simulation-based algorithms in the literature focus solely on single-objective problems (although some of them can be generalized to the multi-objective case). One

reason for this is that single-objective problems are much easier from an approximation perspective since the region of interest, in which the approximation must be good, in general is much larger for multi-objective problems. This is illustrated in Figure 23. Also, for approximations in multi-objective problems it is harder to define what is *good*. As mentioned earlier, it is important that the decisions based on the approximate models are good also if the true models would have been used. To interpret this statement into a performance metric is, however, not easy. It is directly relating to how approximate Pareto sets should be evaluated (cf. Section 5.3).



(a) The near-optimal region for a single-objective minimization problem.

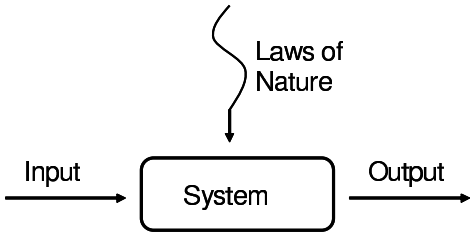
(b) The near-Pareto optimal region for a multi-objective minimization problem.

Figure 23: For the single-objective case it is enough that the approximate function is similar to the true function in a small region. For the multi-objective case, a larger region is interesting, so the respective approximations of the functions need to be good in a larger environment.

Examples of algorithms handling multiple objectives with expensive function evaluations are, e.g., `parEGO` [31], which is an extension of the Kriging-based single-objective algorithm `EGO` [29], and `qualSolve` [26], in which radial basis functions are used for approximations and where points to evaluate are found with an external optimization maximizing a certain quality measure. The algorithm `qualSolve` also handles noisy objective functions, where noise refers to both stochasticity and unpredictable highly non-linear contributions, which in engineering applications is due to, e.g., re-meshing in finite elements simulations.

8.2 Engineering design

A physical problem can be viewed in different ways depending on what is given and what is to be achieved. Figure 24, reproduced from [13], illustrates different possibilities.



Given	Find	Process
Input, Laws of Nature, System	Output	Analysis
Output, Laws of Nature, System	Input	Inverse Analysis
Input, Output, System	Laws of Nature	Science (induction)
Input, Output, Laws of Nature	System	Engineering Design

Figure 24: Different viewpoints of a physical system with various aims.

The design of trucks falls into the fourth category, *Engineering Design*. The inputs to the system are the environment in which the truck will be used and its usage or transport mission. The outputs are the behavior of the truck under excitation of the inputs. Given the inputs, the outputs depend on the (fixed) laws of nature together with the design of the system (the truck).

Engineering design deals with the identification of a need at the market, and the design of a system satisfying that need. If the need can be quantified then optimization might be the right tool to tackle the problem. The scientific area of applying optimization to engineering problem is often denoted *Engineering optimization*. Naturally, engineering optimization problems often possess expensive function evaluations. The subject is therefore strongly connected to simulation-based optimization (cf. Section 8.1). Engineering design, however, is a broader subject than engineering optimization. Areas such as, e.g., aesthetics, collaborative design and knowledge management belong to engineering design as well.

Let us now return to engineering optimization, which is not limited to single-objective optimization design problems. For a designer, it is important how the users evaluate the outputs of the system. This is related to *value*, a central concept of decision theory (cf. Section 8.3). Each user can attach a value measuring how well each design meets his needs. When comparing two designs of an engineering system, a user probably has one preference. Assuming that the users have *rational preferences*, i.e., they will never prefer A to B, B to C, and C to A, then, for each user there exists a perfect ordering of all possible designs. However, when explaining why one design is preferred to another, the user would probably need more than one measure. The overall judgement (value) is an aggregation of a number of value components. This characteristic fits well within a multi-objective optimization context, which makes a multi-objective formulation of the design problem natural. The thesis [1] deals with engineering optimization where there are multiple objectives. This topic is also sur-

veyed in [34].

8.3 Decision theory

Decision theory is concerned about the decisions that humans *should* make and what they *do* make when facing choices. The subject is related to optimization in a natural way, since the purpose of optimization often is to create basic data for making decisions. Knowledge about decision making is required both for formulation of suitable objective functions to use in the optimization (perhaps especially when the problems invokes multiple objectives), and the subject is also connected to how the output from the optimization is actualized.

In [24], a non-technical overview of the subject is presented. Some areas of decision making that are studied are, e.g., how humans make decisions under uncertainty (or stochastics) in the model, decisions in choices carrying too much complexity to be grasped, decision making by groups of people, and how decisions are made when the outcome of the decision is realized at different points in time. One branch of the area of decision theory is *Multi-criteria decision making* (MCDM), where it is explicitly assumed that competing criteria are involved. Obviously, MCDM is closely related to multi-objective optimization.

One area of decision theory concerns decision support tools. These are systems that present the different possibilities and trade-offs for the human decision makers and help them to select the solutions most suitable for their needs (the truly optimal solutions). This is not straightforward, since it can mean searching in a multi-dimensional space that cannot be visualized geometrically (cf. Section 5 where the Pareto optimal set is a subset of $\Re^k$). Inspired by the navigation tool developed and used by ITWM [32] for continuous and convex problems, we have developed our own navigation tool applicable to discrete and non-convex problems. The tool takes a database of objective vectors as input. Then it is based on taking step along some selected objective at the same time as the deviations in the other objectives are minimized. In this way, the axes of the multi-dimensional Pareto optimal objective space can approximately be followed towards a final most preferred point. A screenshot of the tool is presented in Figure 25.

For the configuration problem studied, a decision support tool should be incorporated within a sales tool, helping the customers to find the available configuration in the available set of configuration, giving them the largest utility. A remark is that such a tool must not be relying on the assumption of identical objective functions. The important feature of the tool is that it effectively helps the customer to find the solution $\mathbf{x} \in X_D$ maximizing his/her utility function u.

In some current work, we take a mathematical approach to decision theory in an attempt to define and study concepts that are crucial for using optimization to solve the configuration problem. The purpose of taking such an approach is, in the long run, to better understand how customers behave, so that the quality measures and performance metrics defined, e.g., for evaluating approximate Pareto optimal sets become adapted to what is requested by the market.

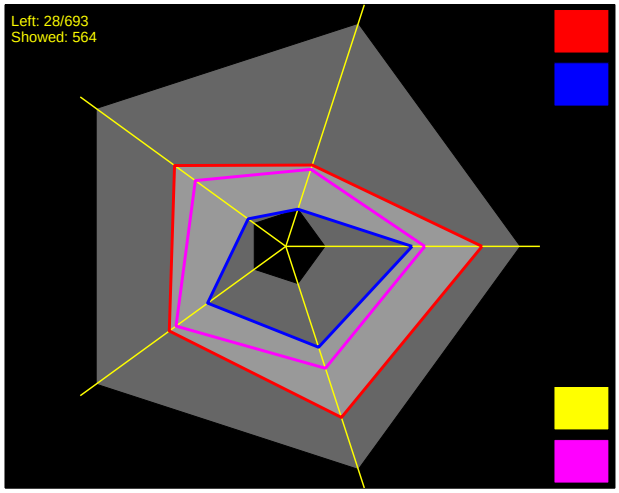


Figure 25: A screenshot of the tool developed for navigation in a Pareto optimal set of a discrete and/or non-convex problem. The actual problem in the Figure has five objective functions. The upper and lower bounds on the objectives have been modified and are represented by the polygons defining the lighter region. The polygon in the interior of the lighter region represents the current solution.

8.4 Constraint programming

Constraint programming (CP) is a discipline originally developed within the Computer science community. However, over the last years there has been a progress towards a merging with Operations research and (mathematical) optimization, see e.g. [21]. Traditionally, CP has mainly dealt with finding feasible solutions to different combinatorial (optimization) problems, and it has successfully been applied to, e.g., scheduling, resource allocation, and packing problems. References can also be found on application of CP to configuration management [7]. The configuration management is in this case mostly related to computer software development (e.g., to track changes of source code), and further, it does not consider the configuration of a population of products with an optimization focus.

Consider a system defined by design variables and a large set of constraints on these. An important solution method used in CP is *constraint propagation*, where, iteratively, the domains of the design variables are reduced for one constraint at a time with the aim of finding values on the decision variables such that all constraints are fulfilled. This filtering method is often combined with a backtracking technique, which is necessary to guarantee that a feasible solution, if one exists, will be found. While it is straightforward to incorporate any types of constraints into the problem

formulation, it is not trivial to introduce costs or an objective function in the CP solution procedure.

Techniques within CP could be used to (quickly) find feasible configurations that fulfil all the restrictions (at least for the mode 1 problem defined on the configuration space). The constraint propagation techniques could also be used as a preprocess for the search for a population of solutions for customers in some environment segment (cf. Section 4.3). Here, if the actual environment segment implies that certain variants in certain variant families must be selected, then these selections could be propagated through the restrictions, while decreasing the domains for the rest of the variant families. Moreover, if the environment segment implies that some variants cannot be selected, then this information could be used to reduce the set of restrictions for the resulting problem.

It is clear from above that some parts of CP can be applied to the configuration problem. However, further research is needed for adapting the techniques to our configuration problem such that only "good" feasible solutions are found.

8.5 The intersection of the domains

In the engineering optimization literature, simulation-based as well as multi-objective optimization problems are widely studied. What is missing for our application is the combinatorial part, where, instead of creating one single design, we are to design a whole population of designs.

Clustering analysis, on the other hand, is about reducing some (finite or infinite) set to a smaller set of clustered points. Clustering analysis is connected to optimization in a sense, since the clustering objective (e.g., maximum homogeneity within groups and maximum heterogeneity between groups) can be viewed as optimization. However, what seems to be missing in the literature is a clear connection to an underlying (multi-objective) optimization problem, and also what comes with a such, clustering in one space (the objective space) with restrictions in another (the decision space).

With this said, it seems to us that in the previous literature, the intersection of all the domains required to describe and solve the configuration problem with the proposed approach is empty.

9 Summary of appended papers

9.1 Paper 1 — Approximating the Pareto Optimal Set using a Reduced Set of Objective Functions

In this paper, we describe a reduction procedure for multi-objective optimization problems when the number of objective functions is large. Through this method, one creates a new optimization problem with fewer objectives and with a Pareto optimal set that is approximately the same as the Pareto optimal set of the original problem. The smaller number of objectives in the reduced problem makes it, in general,

computationally easier to solve.

We introduce concepts such as τ-Pareto optimality, which relaxes the concept of Pareto optimality, and ρ-centrality, which leads to a focus on the (probably most) interesting part of the Pareto optimal set. Our approximation goal is to minimize the distance between the ρ-central part of the original Pareto optimal set and the ρ-central τ-Pareto optimal set of the reduced problem.

Utilizing a new characterization of Pareto optimality (valid for finite decision spaces) which gives Definition 5.1 an explicit formulation, we derive a program whose solution represents an optimal reduction with respect to the approximation objective. We also propose an approximate formulation, computationally tractable contrary to the ideal formulation, which utilizes correlations between the objectives and separates the program into two parts. We demonstrate the method by applying it to a small industrial instance.

The motivation behind the method developed is that industrial problems often require computer intensive simulations and that they often possess a large number of objective functions. These characteristics make them computationally hard to solve. In addition, in practice it is not necessary to find the exact Pareto optimal set; it might be well motivated to lose some precision if the problem to solve becomes significantly smaller, and if the size of the error can be estimated. We have not found any other method in the literature suitable for our aim. Often, very strong assumptions are required for allowing a reduction of the set of objective functions.

A main contribution of this paper is the new explicit characterization of Pareto optimality, which might also be utilized in other applications. Another contribution is the actual method developed which can be used as a preprocess for large-scale multi-objective optimization problems. Here, a subset of the decision space consisting of a finite set of points (small enough to enable an exhaustive search for the Pareto optimal set) must be selected. Using this finite subset, a reduced problem can be constructed using the proposed method, and then the reduced problem formulation can be applied to the original problem.

9.2 Paper 2 — Multi-Objective Design of a Combinatorial Structure

In this paper, we approach the problem of incorporating the combinatorial complexity of product development (cf. Section 3.2) into a multi-objective optimization context and study the procedure denoted by "implicit clustering" and which is defined in Section 7.2.

In the mode 2 configuration problem of Section 4.1, variants are to be selected within variant families such that the resulting set of feasible configurations approximates the Pareto optimal set of the underlying multi-objective optimization problem in a good way. In this paper, we consider a simplification of this configuration problem where the design variables are assumed to be continuous and subject to box constraints only.

A (single-)objective optimization problem, the *Multi-Objective Combinatorial Design Problem* (MOCDP), is introduced. In MOCDP, an underlying multi-objective op-

timization problem (MONP) is used, to which a population of solutions is wanted, approximating its Pareto optimal set $\mathcal{P}$ (cf. Section 5.3 for a discussion on approximations Pareto optimal sets). The decision space is assumed to be combinatorial, i.e., a product set for which a solution is composed by one component in each dimension. Thereby, letting m_j denote the number of variants (i.e., decision variables) in dimension $j = 1, \ldots, n$, $\sum_{j=1}^{n} m_j$ decision variables characterize the (much larger number) $\prod_{j=1}^{n} m_j$ possible configurations (i.e., solutions).

MOCDP is non-convex and non-differentiable even under very strong assumptions on the underlying MONP. A two-step solution method is proposed for solving MOCDP. In the first step a representation of $\mathcal{P}$ is computed and in the second step, global and local optimization algorithms are combined to find a good approximation of the representation from the first step. The method is demonstrated on instances constructed from a standard test problem from the literature. Suggestions are also made on how to adapt the methodology for problems with expensive function evaluations.

The main contribution of the paper is the methodology of implicit clustering, which yields desired outcomes both in the decision space (a certain structure) and in the objective space (a population of solutions well approximating $\mathcal{P}$).

Two main issues that have to be addressed for the current methodology to apply to practical configuration problems are how to incorporate other variable types (integer and categorical) and how to handle more general constraints (or restrictions).

Appendix

A Discretization is a win-win concept

We provide an illustration of the fact that limiting the number of available trucks (to a number of suitable magnitude) might be advantageous for both the producer and the customers. The configuration problem is modeled as a multi-objective optimization problem. The idea is to show that a reduction of the number of configurations (which naturally decreases the indirect product costs) might lead to (independent of the preferred point on the Pareto front) better solutions at the same cost level. The example assumes a simple, yet reasonable, model of how costs enter into the multi-objective configuration problem.

A.2 An example

Assume that the problem comprises two objective functions to be minimized representing two (partially) conflicting performance measures. The product cost is then added to the problem as a third objective to minimize; this cost is a sum of the *direct product cost* and the *indirect product cost*. Assume that the indirect product cost (representing, e.g., product development costs, spare parts warehousing etc.) increases with the number of distinct configurations to be produced. Assume also that by increasing the direct product cost the two performance measures can be improved. An instance complying with these assumptions is given below.

The decision variables are

$$\begin{aligned} x_1 &\in [0, 1], \\ x_2 &\in [\pi, 3\pi/2], \\ x_3 &\in [0, 100], \end{aligned}$$

where x_3 represents the direct product cost. The objective functions to be minimized are

$$\begin{aligned} f_1(\mathbf{x}) &= x_1 \cos(x_2) - k_1 x_3, \\ f_2(\mathbf{x}) &= x_1 \sin(x_2) - k_1 x_3, \\ f_3(\mathbf{x}) &= x_3 + k_2 D, \end{aligned}$$

where f_1 and f_2 represent the performance measures and f_3 represents the total product cost. The parameters k_1 and k_2 are positive constants and D is the number of distinct configurations produced.

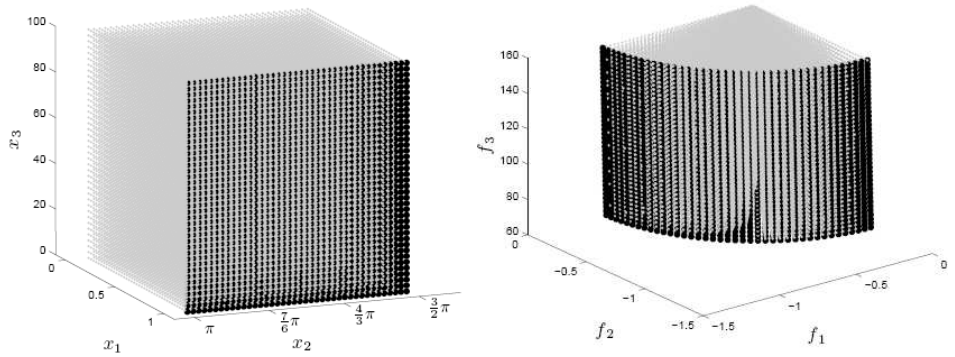
Assume that it is possible to realize 40 distinct variants of each decision variable, corresponding to 40^3 possible configurations. The customer preferences are not known a priori, and are assumed to be spread uniformly along the Pareto front. With $k_1 = \frac{1}{1000}, k_2 = \frac{60}{40^3}$, and $D = 40^3$, the realizations in the decision and the objective

spaces are as illustrated in Figures 26(a) and 26(b). For this problem we have the Pareto optimal set

$$\mathcal{P} = \left\{ \mathbf{x} \in \Re^3 \mid x_1 = 1,\ x_2 \in [\pi, 3\pi/2]\,,\ x_3 \in [0, 100] \right\}, \tag{9}$$

and the image of the Pareto optimal set

$$\mathbf{f}(\mathcal{P}) = \left\{ \mathbf{f} \in \Re^3 \,\middle|\, f_1 = \cos(x_2) - \frac{1}{1000}x_3,\ f_2 = \sin(x_2) - \frac{1}{1000}x_3,\ f_3 = x_3 + 75, \right. \\ \left. x_2 \in [\pi, 3\pi/2]\,,\ x_3 \in [0, 100] \right\}. \tag{10}$$



(a) (b)

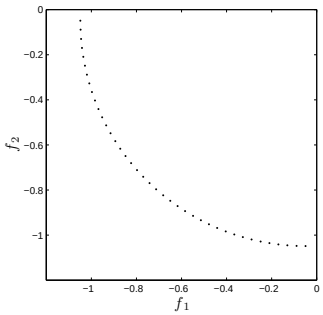
Figure 26: The realized solutions in (a) the decision space and (b) the objective space for the example. The non-dominated (Pareto optimal) solutions in both spaces are marked in black and the dominated part of the feasible set and its image in the objective space are marked in gray.

The total product cost of a configuration lies between 60 and 160 monetary units. We assume that a customer is interested in buying a truck in the middle of this price range ($f_3 = 110 \Leftrightarrow x_3 = 50$). Provided that all 40 non-dominated optimal configurations at the actual cost level f_3 are produced and available to the customer, the set in the objective space to choose from (depending on the appreciation of the two performance measures) is illustrated in Figure 27(a).

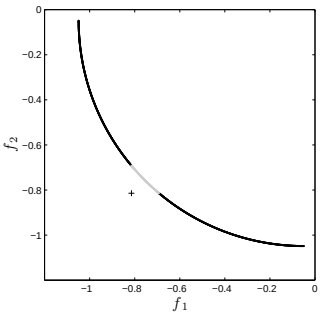
Suppose further that the producer wishes to produce fewer than the 40 possible Pareto optimal trucks. The question then is whether this can be done such that most (or possibly all) of the customers (representing all possible preferences between the performance measures) will benefit.

Obviously, if the discretization is too coarse, then original points in the extreme regions of the Pareto front cannot be covered (dominated) by the discretized points. Also if the discretization is made too fine, then the cost improvement will not be that large and the fewer points will not be able to cover a large part of the original Pareto

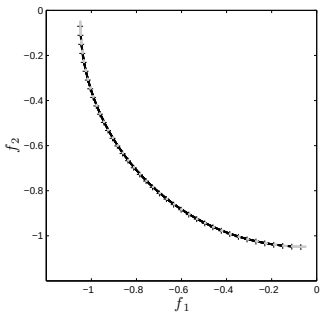
front. In the Figures 27(b)–(d) the part of the original Pareto front with the design restriction $x_3 = 50$ where the 40 configurations are located[5] is shown together with the representatives for the limited sets (represented by + signs). The black regions of the original front correspond to customer choices which are not dominated by any of the representatives, whereas the gray regions correspond to dominated choices. In the Figures 27(b)–(d) the size of the limited sets are chosen to be 1, 39 and 7, respectively.



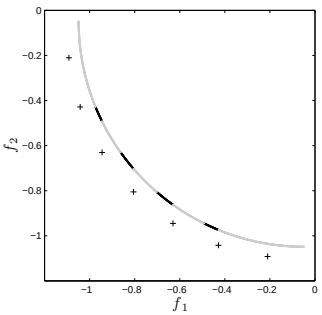
(a) The set of solutions (in objective space) to choose from provided that all 40 configurations are available.



(b) With few solutions in the limited set a small part of the original Pareto front will be dominated.



(c) With many solutions in the limited set a small part of the original Pareto front will be dominated.



(d) A large part of the original Pareto front is dominated by a limited set of cardinality 7.

Figure 27:

For the stated example it is in fact possible to choose discretizations, equally distributed over the Pareto front, such that each original point is dominated by some point in the limited set. Yet, it is not clear that such a coarseness in the discretization is the best one for the whole population of customers, since the subset of customers

[5]Observe that the original Pareto front is continuous and equal to $\mathbf{f}(\mathcal{P})$ from (10) intersected with $f_3 = 110$ given that 40 solutions are produced for each cost level. When the solutions are spread equidistant they are located as in 27(a), but they can be located wherever in the original Pareto front.

whose original choices become dominated will get configurations with indisputably higher quality than otherwise. Thus, there is a trade-off situation between pleasing the average customer and pleasing the worst-case customer.

Customers with preferences corresponding to the gray regions of the Pareto front will unambigously benefit from the discretization. This might be true also for customers in the black regions who would possibly change their preferences if they had to choose between configurations in the union of the black regions and the limited set[6].

Figure 28 shows how the proportion of customers (who are assumed to be equally spread along the Pareto front) who can be offered trucks dominating their original choices varies with the size of the limited set. An important remark, as noted above, is that this measure is a worst-case measure and is therefore possibly underestimating. Also for a considerably lower proportion than 100% of the population can the improvement on average be positive.

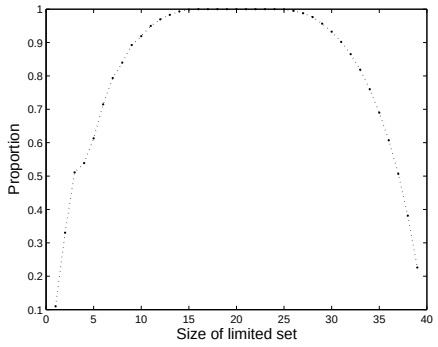


Figure 28: The proportion of the original Pareto front that is dominated by points in the limited set as a function of the size of the limited set.

A.3 Conclusions

The example above shows how a limitation of the number of available configurations can be motivated by domination. The resulting set of configurations to choose between may have better properties than the full set, in the sense that the solutions in the limited set may dominate several of the original solutions at a given product price. In some cases all original solutions are dominated, which can be interpreted as: regardless of a customers preferences, he/she will find a better configuration in the limited set than what would be possible in the original set.

[6]In the example when the size of the discretized size is equal to seven this is true for all customers if they evaluate the different configurations by weighing the two quality measures. This because of that the supported (cf. Section 5.2) non-dominated set of the decision points corresponding to the union of the black regions and the limited set is equal to the decision points corresponding to the limited set.

References

[1] J. ANDERSSON, *Multiobjective Optimization in Engineering Design*, PhD thesis, Division of Fluid and Mechanical Engineering Systems, Department of Mechanical Engineering, Linköping University, Linköping, Sweden, 2003.

[2] M. F. ANJUM, I. TASADDUQ, AND K. AL-SULTAN, *Response surface methodology: A neural network approach*, European Journal of Operational Research, 101 (1997), pp. 65–73.

[3] C. AUDET AND J. J. E. DENNIS, *Pattern search algorithms for mixed variable programming*, SIAM Journal on Optimization, 11 (2001), pp. 573–594.

[4] M. H. BAKR, J. W. BANDLER, K. MADSEN, AND J. SØNDERGAARD, *An introduction to the space mapping technique*, Optimization and Engineering, 2 (2001), pp. 369–384. Special issue on surrogate modeling and space mapping for engineering optimization (Lyngby, 2000).

[5] Y. BAR-YAM, *Dynamics of Complex Systems*, Studies in Nonlinearity, Addison-Wesley, Reading, MA, USA, 1997.

[6] W. M. CARLYLE, J. W. FOWLER, E. S. GEL, AND B. KIM, *Quantitative comparison of approximate solution sets for bi-criteria optimization problems*, Decision Sciences, 34 (1) (2003), pp. 63–82.

[7] T. COATTA AND G. NEUFELD, *Configuration management via constraint programming*, in International Workshop on Configurable Distributed Systems, J. Kramer, ed., London, UK, 1992, Institute of Electrical and Electronics Engineers, Inc., pp. 90–101.

[8] P. CZYŻAK AND A. JASZKIEWICZ, *Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis, 7 (1998), pp. 34–47.

[9] I. DAS AND J. E. DENNIS, *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*, Structural Optimization, 14 (1997), pp. 63–69.

[10] ———, *Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems*, SIAM Journal on Optimization, 8 (1998), pp. 631–657.

[11] K. DEB, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons Ltd., Chichester, England, 2001. With a foreword by David E. Goldberg.

[12] K. DEB, S. AGRAWAL, A. PRATAB, AND T. MEYARIVAN, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, in Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, 2000, Springer-Verlag, pp. 849–858.

[13] J. R. DIXON, *Design Engineering: Inventiveness, Analysis and Decision Making*, McGraw-Hill, New York, NY, USA, 1966.

[14] S. EDLUND AND P.-O. FRYK, *The right truck for the job with global truck application descriptions*, in SAE technical paper 2004-01-2645, New York, NY, USA, 2004.

[15] M. EHRGOTT, *Multicriteria Optimization*, Springer-Verlag, Heidelberg, Germany, 2005.

[16] M. EHRGOTT AND X. GANDIBLEUX, *A survey and annotated bibliography of multiobjective combinatorial optimization*, OR Spektrum, 22 (2000), pp. 425–460.

[17] M. EHRGOTT, D. TENFELDE-PODEHL, AND T. STEPHAN, *A level set method for multiobjective combinatorial optimization: application to the quadratic assignment problem*, Pacific Journal of Optimization. An International Journal, 2 (2006), pp. 521–544.

[18] F. FUXIN, *Evolution and Communication of Geometry Based Product Information within an Extended Enterprise*, PhD thesis, Division of Computer Aided Design, The Polhem Laboratory, Luleå University of Technology, Luleå, Sweden, 2005.

[19] G. GAN, C. MA, AND J. WU, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*, SIAM, Philadelphia, ASA, Alexandria, VA, USA, 2007.

[20] A. GENELL, *Annoyance, quality- and safety aspects of sounds in truck cabins*, Licentiate thesis, Division of Applied Acoustics, Department of Civil and Environmental Engineering, Chalmers University of Technology, Göteborg, Sweden, 2006.

[21] B. GROHE, *Cost Propagation — Numerical Propagation for Optimization Problems*, Licentiate thesis, Department of Computer Science and Engineering, Chalmers University of Technology and Department of Computer Science and Engineering, University of Gothenburg, Göteborg, Sweden, 2007.

[22] H.-M. GUTMANN, *A radial basis function method for global optimization*, Journal of Global Optimization, 19 (2001), pp. 201–227.

[23] H. W. HAMACHER, C. R. PEDERSEN, AND S. RUZIKA, *Finding representative systems for discrete bicriterion optimization problems*, Operations Research Letters, 35 (2007), pp. 336–344.

[24] S. O. HANSSON, *Decision theory: a brief introduction*, Technical report, Department of philosophy, Uppsala University, Uppsala, Sweden, 1994.

[25] C. L. HWANG AND A. S. M. MASUD, *Multiple Objective Decision Making—Methods and Applications*, vol. 164 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, Germany, 1979.

[26] S. JAKOBSSON, M. PATRIKSSON, J. RUDHOLM, AND A. WOJCIECHOWSKI, *A method for simulation based optimization using radial basis functions*, tech. report, Department of Mathematical Sciences, Chalmers University of Technology and Department of Mathematical Sciences, University of Gothenburg, Göteborg, Sweden, 2008.

[27] A. JASZKIEWICZ, *Genetic local search for multi-objective combinatorial optimization*, European Journal of Operational Research, 137 (2002), pp. 50–71.

[28] D. R. JONES, *A taxonomy of global optimization methods based on response surfaces*, Journal of Global Optimization, 21 (2001), pp. 345–383.

[29] D. R. JONES, M. SCHONLAU, AND W. J. WELSH, *Efficient global optimization of expensive black-box functions*, Journal of Global Optimization, 13 (1998), pp. 455–492.

[30] M. KARLSSON, *Load Modelling for Fatigue Assessment of Vehicles — a Statistical Approach*, PhD thesis, Department of Mathematical Sciences, Chalmers University of Technology and Department of Mathematical Sciences, University of Gothenburg, Göteborg, Sweden, 2007.

[31] J. KNOWLES, *ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems*, IEEE Transactions on Evolutionary Computation, 10 (2006), pp. 50–66.

[32] K.-H. KÜFER, M. MONZ, A. SCHERRER, P. SÜSS, F. ALONSO, A. S. A. SULTAN, T. BORTFELD, D. CRAFT, AND C. THIEKE, *Multicriteria optimization in intensity modulated radiotherapy planning*, ITWM Report 77, Fraunhofer Institut Techno- und Wirtschaftmathematik, Kaiserslautern, Germany, 2005.

[33] P. LINGMAN, *Integrated Brake Control: Downhill Driving Strategies*, PhD thesis, Department of Applied Mechanics, Chalmers University of Technology, Göteborg, Sweden, 2005.

[34] R. T. MARLER AND J. S. ARORA, *Survey of multi-objective optimization methods for engineering*, Structural and Multidisciplinary Optimization, 26 (2004), pp. 369–395.

[35] K. MIETTINEN, *Nonlinear Multiobjective Optimization*, Kluwer, Norwell, MA, USA, 1998.

[36] D. O. NORMAN AND M. L. KURAS, *Complex Engineered Systems: Science Meets Technology*, Springer NECSI, Cambridge, MA, USA, 2006, ch. Engineering Complex Systems, pp. 206–245.

[37] M. S. PHADKE, *Quality Engineering Using Robust Design*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1995.

[38] M. PILEGAARD HANSEN AND A. JASZKIEWICZ, *Evaluating the quality of approximations to the non-dominated set*, Technical report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1998.

[39] M. J. D. POWELL, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-scale Nonlinear Optimization, vol. 83 of Nonconvex Optimization and its Application, Springer-Verlag, New York, NY, USA, 2006, pp. 255–297.

[40] S. RUZIKA AND M. M. WIECEK, *Approximation methods in multiobjective programming*, Journal of Optimization Theory and Applications, 126 (2005), pp. 473–501.

[41] S. SAYIN, *Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming*, Mathematical Programming, 87 (2000), pp. 543–560.

[42] N. SENIN, D. R. WALLACE, N. BORLAND, AND M. J. JAKIELA, *Distributed modeling and optimization of mixed variable design problems*, Technical report, CADlab, Department of Mechanical Engineering, MIT, Cambridge, MA, USA, 1999.

[43] R. STEUER, *Multiple Criteria Optimization: Theory, Computation and Application*, John Wiley & Sons Inc, New York, NY, USA, 1986.

[44] A. VIANA AND J. PINHO DE SOUSA, *Using metaheuristics in multiobjective resource constrained project scheduling*, European Journal of Operational Research, 120 (2000), pp. 359–374.

[45] S. WATANABE, T. HIROYASU, AND M. MIKI, *NCGA: Neighborhood cultivation genetic algorithm for multi-objective optimization*, in Proceedings of the Genetic and Evolutionary Computation Conference, E. Cantú-Paz, ed., New York, NY, USA, 2002, pp. 458–465.

[46] E. ZITZLER, L. THIELE, M. LAUMANNS, C. FONSECA, AND V. FONSECA, *Performance assessment of multiobjective optimizers: an analysis and review*, IEEE Transactions on Evolutionary Computation, 7 (2003), pp. 117–132.

Approximating the Pareto Optimal Set using a Reduced Set of Objective Functions

Peter Lindroth Michael Patriksson Ann-Brith Strömberg

Abstract

Real - world applications of multi - objective optimization often involve a large number of objective functions. But while such problems are computationally very difficult in general, it is often not necessary to find the Pareto optimal set exactly. It is therefore motivated to lose some precision if the computational burden becomes significantly smaller and the size of the error can be estimated. We describe a method for an optimal reduction of the set of objectives yielding a smaller problem whose Pareto optimal set, in a Hausdorff distance sense, is as similar as possible to the Pareto optimal set of the original problem. Our focus lies only on the part of the decision space that is interesting in the sense that it is (near-)Pareto optimal. Using a new characterization of Pareto optimality, we derive a program whose solution represents an optimal reduction. We also propose an approximate formulation, computationally less demanding, which utilizes correlations between the objectives and separates the program into two parts. The method is illustrated with a graphical example, in which the obtained results can be viewed graphically. Numerical results for an industrial instance concerning the configuration of heavy-duty trucks are also reported, demonstrating the usefulness of the developed method. The results show that multi-objective problems can be simplified with an inducing error of which there is a measure.

1 Introduction

Engineering design problems typically deal with the construction of designs handling a number of more or less conflicting criteria. Such problems can mathematically be formulated as a multi-objective optimization problem with the standard notation

$$\min_{\mathbf{x}\in X} \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\}, \tag{1}$$

where $\mathbf{x} \in \Re^n$ denotes a vector of decision variables, $X \subseteq \Re^n$ is the set of feasible decision vectors (or the decision space), and each $f_i : X \to \Re,\ i = 1, \ldots, k$, is an objective function to be minimized. The vector of objective functions is denoted $\mathbf{f} = \{f_1, \ldots, f_k\}$. The objective space, Z, is defined as being the image of the decision

space, i.e., $Z = \mathbf{f}(X) = \{\mathbf{z} = \mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in X\}$. If the objective functions are at least partially in conflict, i.e., there is no feasible decision vector minimizing all objectives simultaneously, then the optimal solution to (1) is not well-defined since there exists no natural complete ordering of vectors. However, there exists a set of decision vectors in which the best solution by rational judgments must be contained regardless of the relative importance of each single objective, namely the *Pareto optimal set* (or, equivalently, the *efficient* or *non-dominated* set).

Definition 1.1 (Pareto optimality) *Given a set X of feasible decision vectors and a set $\{f_1, \ldots, f_k\}$ of objective functions to be minimized, a vector $\mathbf{x}^* \in X$ is defined as* Pareto optimal *if there exists no vector $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \{1, \ldots, k\}$. An objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$ is called* Pareto optimal *if the corresponding vector $\mathbf{x}^*$ is Pareto optimal. The set of all Pareto optimal decision vectors $\mathbf{x}^* \in X$ is denoted $\mathcal{P} \subseteq X$.*

Definition 1.2 (Domination) *A vector $\mathbf{x} \in X$ is said to* dominate *a vector $\mathbf{y} \in X$ if $f_i(\mathbf{x}) \leq f_i(\mathbf{y}),\ i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, \ldots, k\}$.*

The Pareto optimal set (in the objective space) is bounded from below by the *ideal* vector, $\mathbf{z}^{\text{ideal}}$, and from above by the *nadir* vector, $\mathbf{z}^{\text{nad}}$, both defined below.

Definition 1.3 (Ideal/nadir vector) *The* ideal *vector $\mathbf{z}^{\text{ideal}} \in \Re^k$ is the vector consisting of the component-wise minimum of each objective over the Pareto optimal set,*

$$\mathbf{z}^{\text{ideal}} = \Big(\min_{\mathbf{y} \in \mathcal{P}} f_1(\mathbf{y}), \ldots, \min_{\mathbf{y} \in \mathcal{P}} f_k(\mathbf{y}) \Big). \tag{2}$$

Likewise, the nadir *vector $\mathbf{z}^{\text{nad}} \in \Re^k$ is the vector consisting of the component-wise maximum of each objective over the Pareto optimal set,*

$$\mathbf{z}^{\text{nad}} = \Big(\max_{\mathbf{y} \in \mathcal{P}} f_1(\mathbf{y}), \ldots, \max_{\mathbf{y} \in \mathcal{P}} f_k(\mathbf{y}) \Big). \tag{3}$$

Definition 1.4 (Weak Pareto optimality) *A vector $\mathbf{x}^* \in X$ is defined as* weakly Pareto optimal *if there exists no other vector $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*),\ i = 1, \ldots, k$. An objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$ is called* weakly Pareto optimal *if the corresponding vector $\mathbf{x}^*$ is weakly Pareto optimal. The set of all weakly Pareto optimal vectors is denoted $\mathcal{P}_w$.*

We are interested in instances of the problem (1) for which the feasible set X is finite. For such problems, it is possible to give an equivalent formulation of Pareto optimality that appears to be new. We start by introducing this characterization, which is more explicit than Definition 1.1 and which is in some cases better suited for modeling actual problems.

Proposition 1.5 (Pareto system) *Let $N \geq 1$ be an arbitrary integer, $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, and $M \gg 1$. Then a vector $\mathbf{x}^* \in X$ is Pareto optimal with respect to the objective functions $\{f_1, \ldots, f_k\}$ if and only if the system* (4) *below is consistent.*

$$f_i(\mathbf{x}^*) < f_i(\mathbf{x}^j) + M(1 - u_{ij}), \qquad j = 1, \ldots, N,\ i = 1, \ldots, k, \tag{4a}$$

$$\sum_{i=1}^{k} f_i(\mathbf{x}^*) \leq \sum_{i=1}^{k} f_i(\mathbf{x}^j) + M(1 - u_{0j}), \qquad j = 1, \ldots, N, \tag{4b}$$

$$\sum_{i=0}^{k} u_{ij} \geq 1, \qquad j = 1, \ldots, N, \tag{4c}$$

$$u_{ij} \in \{0, 1\}, \qquad j = 1, \ldots, N,\ i = 0, \ldots, k. \tag{4d}$$

Proof. The Definition 1.1 of Pareto optimality can be restated as follows: for $\mathbf{x}^* \in X$ to be Pareto optimal, it is required that there is no $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, and $\sum_{i=1}^{k} f_i(\mathbf{x}) < \sum_{i=1}^{k} f_i(\mathbf{x}^*)$ hold. Equivalently, for each $\mathbf{x} \in X$, no more than k out of these $k+1$ constraints may be fulfilled for $\mathbf{x}^* \in X$ to be Pareto optimal. Equivalently, by expressing the complement, for each $j = 1, \ldots, N$, there must be at least one out of the $k+1$ constraints $f_i(\mathbf{x}^j) > f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, and $\sum_{i=1}^{k} f_i(\mathbf{x}^j) \geq \sum_{i=1}^{k} f_i(\mathbf{x}^*)$ that is fulfilled. This is equivalent to the system (4) being consistent. ∎

Remark 1.6 *Pareto optimality can be replaced by weak Pareto optimality in Proposition* 1.5 *if the strict inequality in* (4a) *is replaced by a (non-strict) inequality and if the inequalities* (4b) *are removed.*

By solving (1), we mean to find the set $\mathcal{P} \subseteq X$. As the number k of objectives increases the task of finding (a good approximation of) $\mathcal{P}$ in general becomes computationally increasingly difficult (cf. [8]; cf. also [3] in which the claim is the opposite for some special problems). The number k of objectives is large in the instances of (1) that we are interested in, and therefore our approach to solve these problems is based on the selection of a smaller set of objective functions. We next present an overview of such approaches.

1.1 Previous work

The notion of *redundant* (or *non-essential*) objectives was introduced in [12], referring to objective functions that would not affect the Pareto optimal set if they were removed from the problem formulation. It is proved that for linear multi-objective programs (problems of type (1) with all f_i linear and X polyhedral) an objective whose cost vector is a positive linear combination of some other cost vectors is redundant. From an application point of view, however, it is noted in [11] that dropping redundant functions often will affect the final solution(s) obtained from standard multicriteria decision making methods. In [17] it is stated that for non-linear problems, and especially in connection with interactive solution methods, it is more suitable to define redundancy on the basis of "conflicts" between the objectives, where no conflict between a pair of objectives means that all feasible decision vectors are sorted

equally by the two objectives. Agrell [1] proposes a different definition of conflict, only requiring the sorting to be equal over the efficient set. More definitions of conflict can be found in [4]. Measures of interdependencies are defined in [5]. Equal or opposite sorting over the decision space is required for two objectives to be interdependent. In [8], a method is proposed on how to reduce the set of objectives based on the Principal Component Analysis technique (PCA), where, roughly speaking, the objectives in the reduced problems are the ones that retain as much variation of the original objective space Z as possible.

The above-mentioned works aim at a reformulation of the problem (1) such that the Pareto optimal set is retained. However, after a reduction of redundant objectives their number may still be too large. We construct a measure of "partial redundancy" among objectives. Based on this measure we construct models with less than k objectives, having a lower computational complexity, and for which the Pareto optimal sets are *similar* to that of (1).

In [4] a method for reducing the set of objectives is presented. The focus lies on keeping the dominance structure in the whole of the decision space X.That is, the aim is to drop objectives such that for all $\mathbf{x}, \mathbf{y} \in X$; $\mathbf{x}$ dominates $\mathbf{y}$ in the reduced problem if and only if $\mathbf{x}$ dominates $\mathbf{y}$ in the original problem. The method is in [4] extended to allow some changes in the dominance structure leading to a change in the Pareto optimal set. The aim of this method is similar to ours, however the focus lies on the dominance structure in the complete decision space X (which might contain parts with completely uninteresting points) instead of on the minimization of the differences between the respective Pareto optimal sets (where it is important that the reduced problem is a good approximation of the original one).

The concept of *partial weighting* is introduced in [14]. This can be seen as a generalization of the traditional scalarization technique [17] for solving multi-objective optimization problems: some of the objectives are replaced by their weighted sum, thereby reducing the dimension of the problem. How the objectives should be grouped is however not clearly proposed. It is shown that the Pareto optimal set $\mathcal{P}^\beta$ for the reduced problem is a subset of $\mathcal{P}$, but the authors do not analyze or characterize the vectors of $\mathcal{P}$ that are lost. We contribute both with a method for the selection of objectives for the reduced problem and characterizations of the Pareto optimal vectors in the original and reduced problems.

1.2 Motivation

The main contribution of this work is a practical method for simplifying a multi-objective optimization problem without losing too much information. Noted in the literature is that usually very strong assumptions are made for allowing the number of objective functions to be reduced; it is often required that the Pareto optimal set is retained after the reduction (i.e., $\mathcal{P}^\beta = \mathcal{P}$). Practical engineering problems often invoke many objective functions and it may be well motivated to lose some precision of the Pareto optimal set if the problem to solve becomes significantly smaller and if there is some measure on the size of the error made. We mean to find a smaller representation—in terms of objective functions—of the problem such that $\mathcal{P}^\beta$ and $\mathcal{P}$

are as similar as possible. The method can be used at a preprocessing stage when solving optimization problems with many objectives and with a large set X' of feasible decision vectors. By studying a subset $X \subseteq X'$, small enough to enable an exhaustive search for the Pareto optimal set $\mathcal{P}$, the method developed reduces the problem in such a way that the difference between $\mathcal{P}$ and the Pareto optimal set of the reduced problem, $\mathcal{P}^\beta$, are held at minimum. This reduced problem formulation can then be applied to the complete decision space X'.

1.3 Outline

In Section 2 we investigate some relations between $\mathcal{P}$ and $\mathcal{P}^\beta$ (when modifying the set of objective functions) and propose several measures of the distance between them. By showing that $\mathcal{P}$ shrinks when using certain reduction rules we motivate the introduction of a *dominance tolerance* parameter to increase $\mathcal{P}^\beta$. We will also introduce a *centrality* parameter, aiming at focusing the approximation on the most important part of the resulting set. The number of objective functions and the value of the centrality parameter are to be fixed in the model, whence the variables defining the objectives in the reduced problem together with the tolerance parameter are the decision variables for which we search optimal values.

In Section 3 we utilize the above parameters and Proposition 1.5 to develop a binary linear program (an ideal model) whose solution represents the optimally reduced problem, i.e., the reduced problem whose Pareto optimal set is, with respect to the measures developed in Section 2, as similar as possible to the original Pareto optimal set. The complexity of this model is far too high for a problem of practical size to be solved in reasonable time. Therefore, in Section 4 we consider an approximation of the ideal model based on the separation of the decision variables. The approximate model can be solved effectively, and its solution represents an approximation of the optimally reduced multi-objective optimization problem.

In Section 6 the approximate model is applied to both an academic example which can be viewed graphically and also to a realistic problem concerning configurations of heavy-duty trucks.

2 Reducing the set of objective functions

In this section we study how the Pareto optimal set varies with the set of objective functions. We propose a quality measure for the Pareto optimal set of the reduced problem, $\mathcal{P}^\beta$, and introduce a parameter with the purpose of maximizing the similarity between the possibly most important regions, the *central* regions (cf. Section 2.1), of $\mathcal{P}$ and $\mathcal{P}^\beta$.

Reducing a multi-objective optimization problem by dropping objectives may induce an error in the sense that the Pareto optimal set $\mathcal{P}$ of the original problem differs from that of the reduced one. In [6], a review of quality measures for an approximate Pareto optimal set, here denoted $\hat{\mathcal{P}}$, is presented. These are mostly used to evaluate new metaheuristics. Noted is that there exists no standard measure or even set of

measures. Further, most proposals aim at evaluating the performance of proposed heuristics for which the size of $\hat{\mathcal{P}}$ is significantly smaller than the size of $\mathcal{P}$. A good approximation normally means that $\hat{\mathcal{P}}$ is well spread over $\mathcal{P}$ and that the points in $\hat{\mathcal{P}}$ are also near-optimal, meaning that each point in $\hat{\mathcal{P}}$ is close to some point in $\mathcal{P}$. One measure, introduced in [7] and also used in [20], is defined by the metric

$$\mathit{Dist2}(\hat{\mathcal{P}}) = \max_{\mathbf{x}\in\mathcal{P}} \left\{ \min_{\mathbf{y}\in\hat{\mathcal{P}}} c(\mathbf{x},\mathbf{y}) \right\}, \tag{5}$$

where $c(\cdot,\cdot)$ is some function measuring the *closeness* of a pair of points. In words, *Dist2* is the largest deviation of a point in $\mathcal{P}$ from its nearest point in $\hat{\mathcal{P}}$. We wish to find out how well $\mathcal{P}^\beta$ approximates $\mathcal{P}$ but also the opposite; how well $\mathcal{P}$ approximates $\mathcal{P}^\beta$. Therefore we extend the *Dist2* measure to the well-known Hausdorff distance measure, which measures how distant two non-empty compacts sets are in a certain metric. With $d(\cdot,\cdot)$ denoting the distance metric, the Hausdorff distance, also illustrated in Figure 1, between the closed, discrete, and non-empty sets E and F is defined as

$$d_H(E,F) = \max \left\{ \max_{\mathbf{u}\in E} \min_{\mathbf{v}\in F} d(\mathbf{u},\mathbf{v}); \max_{\mathbf{v}\in F} \min_{\mathbf{u}\in E} d(\mathbf{v},\mathbf{u}) \right\}. \tag{6}$$

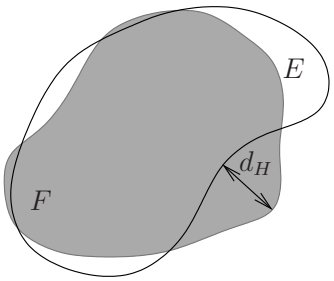


Figure 1: Illustration of the Hausdorff distance between the two sets E and F.

Given the multi-objective optimization problem (1), the question of how many objective functions that are required to define $\mathcal{P}$ is raised in [9]. For strictly quasi-convex functions f_i it is shown that for a problem with $n = 2$, a maximum of three objectives are enough. However, without assuming convexity, or even continuity, of the objective functions it is clear that, in principle, only one objective function is required, e.g., the indicator function

$$\xi(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \notin \mathcal{P}, \\ 0, & \text{if } \mathbf{x} \in \mathcal{P}. \end{cases}$$

Since our method is intended to be used on a subset of the complete decision space—hoping that the reduced set of functions are good also in the complete decision space—the indicator function is not a wise choice. In machine learning language; the

wish is to find a hypothesis (a set of new objective functions) that fits the entire data (X') well (leads to approximately the same Pareto optimal set), and not just the data measured (X). For this reason, hypotheses that over-fit the data must be avoided, meaning that there exists an alternative hypothesis that explains the measured data worse but explains the entire data better. Choosing a simpler hypothesis instead of a more complicated one sometimes goes under the name *Occam's razor* [18]. In our application, the set of all hypotheses are all functions $\{\mathbf{g} : X \to \Re^r\}$, where $r < k$ is the number of objectives in the reduced problem. In this set of functions we want to find the function $\mathbf{g}$ that approximates the Pareto optimal set the best over the entire set of data. We have chosen to limit the set of hypotheses to those for which the objective functions of the reduced problem are allowed to be linear combinations of the original ones. This leads to a fairly simple hypothesis that still seems reasonable. Furthermore, we have restricted the weights of the linear combinations to be uniform so that a set of two or more objectives may be replaced by their mean only. Thus, we end up with a procedure similar that introduced in [14]: a partial weighting of the original objectives.

Let the power set, i.e., the set of all subsets, of $\{1, \ldots, k\}$ be denoted by $\mathcal{R} = \{R_1, \ldots, R_{2^k}\}$. Our aim is to find a subset of $\mathcal{R}$ with maximal cardinality r defining which collections of original objective functions to aggregate as objectives in the reduced problem. In other words, the aim is to reduce the set of objective functions $\{f_1, \ldots, f_k\}$ to $\{g_{s_1}, \ldots, g_{s_r}\}$, $r < k$, where $g_{s_j} = \frac{1}{|R_{s_j}|} \sum_{i \in R_{s_j}} f_i$, and where each R_{s_j} indicates which of the original objectives that are included in the j:th chosen objective g_{s_j} in the reduced problem. We also require that each objective function f_i must be included among the terms of some linearly combined function g_{s_j}, i.e., $\cup_{j=1}^{r} R_{s_j} = \{1, \ldots, k\}$. As indicators of which elements in $\mathcal{R}$ that are chosen we introduce the binary variables

$$\beta_p = \begin{cases} 1, & \text{if collection } p \text{ of objectives is chosen,} \\ 0, & \text{otherwise,} \end{cases} \qquad p = 1, \ldots, 2^k. \tag{7}$$

Let $A \in \mathbb{B}^{k \times 2^k}$ be the incidence matrix indicating whether or not a certain objective is included in a certain collection, i.e., its elements are

$$a_{ip} = \begin{cases} 1, & \text{if objective } f_i \text{ is contained in collection } p, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, \ldots, k,\ p = 1, \ldots, 2^k.$$

The feasible choices of collections, for a prescribed maximal number r of objectives in the reduced problem, are then defined by all binary vectors $\boldsymbol{\beta} \in \{0, 1\}^{2^k}$ fulfilling

$$\begin{aligned} A\boldsymbol{\beta} &\geq \mathbf{1}^k, \\ \boldsymbol{\beta}^T \mathbf{1}^{2^k} &\leq r, \\ \boldsymbol{\beta} &\in \{0, 1\}^{2^k}. \end{aligned} \tag{8}$$

Now we return to a general case of problem modification, which is not restricted by our requirements above. It is interesting to investigate how the set of Pareto optimal decision vectors varies with the set of objective functions. Dropping an objective function (disallowed by our rules) will reduce the set of weakly Pareto optimal points for problems with convex objective functions; this is a consequence of, e.g., Corollary 1 in [15]. It is true also for a general problem, guaranteed by the following.

Proposition 2.1 *Consider the problem* (1). *Let* $K = \{f_1, \ldots, f_k\}$ *be the set of objective functions and* $K^\beta \subset K$ *be the remaining set when some of the objective functions have been removed. Let* $\mathcal{P}_w \subseteq X$ $(\mathcal{P}_w^\beta \subseteq X)$ *be the set of weakly Pareto optimal vectors corresponding to the set* K (K^β). *Then,* $\mathcal{P}_w^\beta \subseteq \mathcal{P}_w$.

Proof. Assume without loss of generality that $K^\beta = \{f_2, \ldots, f_k\}$. Suppose $\mathbf{y}^* \in \mathcal{P}_{\mathrm{w}}^\beta$. This implies that $\nexists \mathbf{y} \in X$ such that $f_i(\mathbf{y}) < f_i(\mathbf{y}^*),\ i \in \{2, \ldots, k\}$. Hence, $\nexists \mathbf{y} \in X$ such that $f_i(\mathbf{y}) < f_i(\mathbf{y}^*),\ i \in \{1, \ldots, k\}$, and thus, $\mathbf{y}^* \in \mathcal{P}_{\mathrm{w}}$. ∎

Remark 2.2 *Note that the analogous statement to Proposition* 2.1 *where weak Pareto optimality is replaced by Pareto optimality is false. Consider the case to minimize* $\{f_1(\mathbf{x}), f_2(\mathbf{x})\}$ *over* X, *with* $f_1(\mathbf{x}) = x_1$, $f_2(\mathbf{x}) = x_2$, *and* $X = \{(1,2); (2,1); (3,1)\}$. *Then the Pareto optimal set* $\mathcal{P} = \{(1,2); (2,1)\}$. *By dropping* f_1 *we obtain* $\mathcal{P}^\beta = \{(2,1); (3,1)\}$, *i.e.,* $(3,1) \in \mathcal{P}^\beta$ *but* $(3,1) \notin \mathcal{P}$.

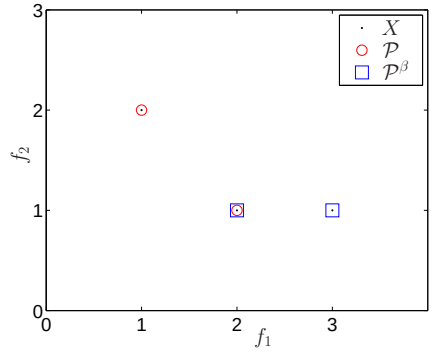


Figure 2: Illustration of the fact that weak Pareto optimality cannot be replaced by Pareto optimality in Proposition 2.1.

However, Proposition 2.3 below guarantees that by replacing a subset of the objective functions with a positively weighted sum of themselves the set of Pareto optimal solutions is reduced. Thus we conclude that if constructing a reduced problem by replacing certain subsets of objective functions with their respective weighted means, then $\mathcal{P}^\beta \subseteq \mathcal{P}$ holds.

Proposition 2.3 *Consider the problem* (1). *Let* $K = \{f_1, \ldots, f_k\}$ *be the set of objective functions and* $K^\beta = \{w_1 f_1 + w_2 f_2, f_3, \ldots, f_k\},\ w_1, w_2 > 0$, *be the reduced set where the first two objectives are replaced by a positively weighted sum. Let* $\mathcal{P} \subseteq X$ ($\mathcal{P}^\beta \subseteq X$) *be the set of Pareto optimal vectors corresponding to the set* K (K^β). *Then,* $\mathcal{P}^\beta \subseteq \mathcal{P}$.

Proof. See [14].

2.1 Centrality

Reducing the set of objective functions through partial weighting leads to the loss of extreme Pareto optimal solutions, i.e., solutions with very good values in one objective but arbitrarily poor values in the others. These are probably not very attractive anyway when a final solution is picked from the set of Pareto optimal solutions because of their extreme nature. In order to obtain a set $\mathcal{P}^\beta$ that differs from $\mathcal{P}$ as little as possible for the solutions that are likely to be chosen as final solutions, a centrality parameter $\rho \in [0, 1]$ is defined, and the vectors $\mathbf{x} \in E \subseteq X$, where E is any subset of X (e.g. $\mathcal{P}$ or $\mathcal{P}^\beta$), are partitioned into a *ρ-central* and a *non-ρ-central* part.

Definition 2.4 (ρ-centrality) *Consider the multi-objective optimization problem* (1). *The* ρ-central *part* $E^\rho \subseteq E$ *of a set* $E \subseteq X$ *is defined as*

$$E^\rho = \left\{ \mathbf{x} \in E \;\middle|\; f_i(\mathbf{x}) \leq (1-\rho) z_i^{\text{nad}} + \rho z_i^{\text{ideal}},\ i = 1, \ldots, k \right\}. \tag{9}$$

Proposition 2.5 *For any subset* $E \subseteq X$ *it holds that* $E^\rho = E \cap X^\rho$.

In words, the ρ-central part of the Pareto optimal set, $\mathcal{P}^\rho = \mathcal{P} \cap X^\rho$, consists of the Pareto optimal vectors that have no objective function value relative to the span of that objective over $\mathcal{P}$, closer than ρ to any component of the *nadir* vector. As special cases we have for a set $E \subseteq X$, $E^0 = \{\mathbf{x} \in E \mid \mathbf{f}(\mathbf{x}) \leq \mathbf{z}^{\text{nad}}\}$ and $E^1 = \{\mathbf{x} \in E \mid \mathbf{f}(\mathbf{x}) \leq \mathbf{z}^{\text{ideal}}\}$ ($= \emptyset$ if the objectives are partially in conflict). Figure 3 illustrates the concepts.

2.2 Dominance tolerance

We want to approximate the Pareto optimal set as well as is possible using the principles described above. Therefore, a non-negative tolerance parameter, $\tau \geq 0$, is introduced, with the task of enlarging the Pareto optimal set of the reduced problem to obtain a central part (with respect to the objectives in the reduced problem), $\mathcal{P}^{\beta,\rho}$, as similar as possible to the central part of the original Pareto optimal set, $\mathcal{P}^\rho$. We define the *τ-Pareto optimal set*, $\mathcal{P}_\tau$, as follows.

Definition 2.6 (τ-Pareto optimality) *For* $\tau \geq 0$, *a set* X *of feasible vectors and a set* $\{f_1, \ldots, f_k\}$ *of objective functions to be minimized, a vector* $\mathbf{x}^* \in X$ *is defined to be* τ-Pareto optimal *if there exists no vector* $\mathbf{x} \in X$ *such that* $f_i(\mathbf{x}) + \tau \leq f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, *and* $f_j(\mathbf{x}) + \tau < f_j(\mathbf{x}^*)$ *for at least one* $j \in \{1, \ldots, k\}$. *An objective vector* $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$

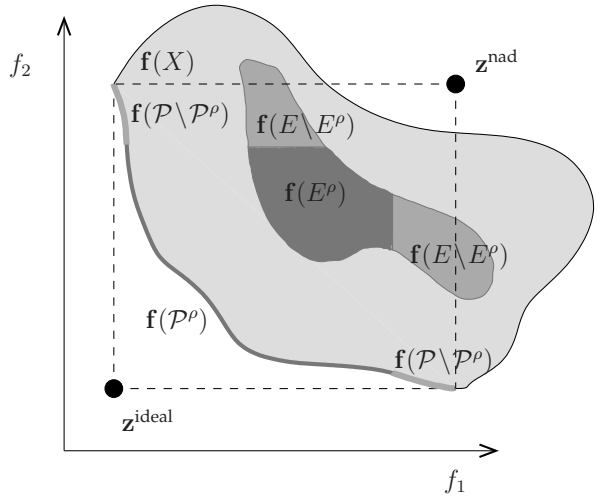


Figure 3: An illustration of the ρ-centrality concept in the objective space for a multi-objective optimization problem with two objective functions. The concept is shown for $\rho \approx 0.2$ both for the Pareto optimal set $\mathcal{P}$ and for another subset $E \subset X$. Shown are also the ideal and the nadir vectors for the problem.

is called τ-Pareto optimal *if the corresponding vector* $\mathbf{x}^*$ *is* τ*-Pareto optimal. The set of* τ*-Pareto optimal decision vectors* $\mathbf{x}^* \in X$ *is denoted* $\mathcal{P}_\tau \subseteq X$*.*

By construction, $\mathcal{P} \subseteq \mathcal{P}_{\tilde{\tau}} \subseteq \mathcal{P}_\tau$ holds for all $\tau \geq \tilde{\tau} \geq 0$.

We now summarize our goal: Given the original multi-objective optimization problem defined by the set X of feasible vectors and the set $\{f_1, \ldots, f_k\}$ of objective functions together with the number $r < k$ of objective functions in the reduced problem and the value $\rho \in [0, 1]$ of the centrality parameter, we wish to find the optimal set of collections $\{s_1, \ldots, s_r\} \subset \mathcal{R}$ (defining the set $\{g_{s_1}, \ldots, g_{s_r}\}$), and the value of the tolerance parameter $\tau \geq 0$, such that the Hausdorff distance $d_H(\mathbf{f}(\mathcal{P}^\rho), \mathbf{f}(\mathcal{P}_\tau^{\beta,\rho}))$, illustrated in Figure 4 (where we indicate that the sets we study are discrete), is minimized. The problem can be stated as that to

$$\begin{aligned} \underset{\boldsymbol{\beta},\tau}{\text{minimize}} \quad & \delta(\boldsymbol{\beta}, \tau) := d_H(\mathbf{f}(\mathcal{P}^\rho), \mathbf{f}(\mathcal{P}_\tau^{\beta,\rho})), \\ \text{subject to} \quad & A\boldsymbol{\beta} \geq \mathbf{1}^k, \\ & \boldsymbol{\beta}^T \mathbf{1}^{2^k} \leq r, \\ & \boldsymbol{\beta} \in \{0,1\}^{2^k}, \\ & \tau \geq 0. \end{aligned} \tag{10}$$

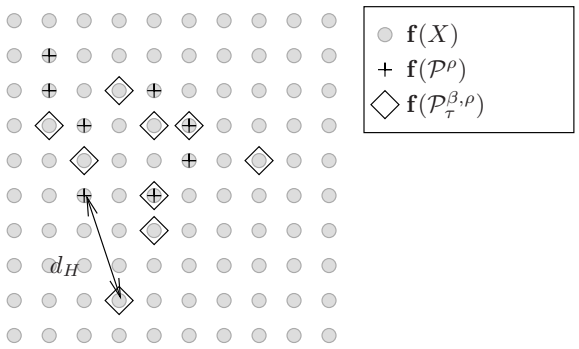


Figure 4: An illustration of the Hausdorff distance (using the Euclidean metric) between the (discrete) sets $\mathbf{f}(\mathcal{P}^{\rho})$ and $\mathbf{f}(\mathcal{P}_{\tau}^{\beta,\rho})$.

3 The ideal model

Starting from the characterization of Pareto optimality in Proposition 1.5, we here construct a binary linear program, the "ideal" model, for solving the problem (10).

3.1 An explicit formulation of Pareto optimality

Clearly, $\mathcal{P}_{\tau}^{\beta,\rho}$ depends on the choices of $\boldsymbol{\beta}$ and τ. Thus, a well-posed optimization problem requires an explicit formulation of Pareto optimality. Such a formulation is provided through Proposition 1.5. In the following sections we develop explicit constraints that distinguish τ-Pareto optimal vectors from non-τ-Pareto optimal ones. We start with a formulation for the general problem (1) and then in Section 3.2 we apply it to the reduced problem (the problem (1) with $\{g_{s_1}, \ldots, g_{s_r}\}$ instead of $\{f_1, \ldots, f_k\}$) as objectives) incorporating $\boldsymbol{\beta}$ and τ in the formulation.

We construct, using the system (4), a consistent system of inequalities, which partitions the set X into a Pareto optimal and a non-Pareto optimal set. Consistency of (4a), (4b) and (4d) is required (which is always feasible, e.g., with $u_{ij} = 0,\ \forall i, j$) and binary variables $w_\ell,\ \ell = 1, \ldots, N$, are introduced indicating if also (4c) holds together with (4a), (4b) and (4d) for a certain vector $\mathbf{x}^\ell$. Below, the auxiliary variables u and v are introduced in order to set the right values on w. The vectors $\mathbf{x}^j$ and $\mathbf{x}^\ell$ are two specific vectors in X.

Let

$$u_{ij\ell} = \begin{cases} 1, & \text{if } f_i(\mathbf{x}^\ell) < f_i(\mathbf{x}^j), \\ 0, & \text{otherwise,} \end{cases} \qquad j, \ell = 1, \ldots, N,\ i = 1, \ldots, k,$$

$$u_{0j\ell} = \begin{cases} 1, & \text{if } \sum_{i=1}^k f_i(\mathbf{x}^\ell) \leq \sum_{i=1}^k f_i(\mathbf{x}^j), \\ 0, & \text{otherwise,} \end{cases} \qquad j, \ell = 1, \ldots, N,$$

$$v_{j\ell} = \begin{cases} 1, & \text{if } \mathbf{x}^j \text{ does not dominate } \mathbf{x}^\ell \text{ (i.e., if } \sum_{i=0}^k u_{ij\ell} \geq 1), \\ 0, & \text{if } \mathbf{x}^j \text{ dominates } \mathbf{x}^\ell, \end{cases} \qquad j, \ell = 1, \ldots, N,$$

$$w_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in \mathcal{P} \text{ (i.e., if } v_{j\ell} = 1\ \forall j), \\ 0, & \text{if } \mathbf{x}^\ell \notin \mathcal{P}, \end{cases} \qquad \ell = 1, \ldots, N.$$

These variable declaration can be formulated as the following system of inequalities[1].

$$-Mu_{ij\ell} \leq f_i(\mathbf{x}^\ell) - f_i(\mathbf{x}^j) < M(1 - u_{ij\ell}), \qquad j = 1, \ldots, N,\ i = 1, \ldots, k, \tag{11a}$$

$$-Mu_{0j\ell} < \sum_{i=1}^k f_i(\mathbf{x}^\ell) - \sum_{i=1}^k f_i(\mathbf{x}^j) \leq M(1 - u_{0j\ell}), \quad j = 1, \ldots, N, \tag{11b}$$

$$v_{j\ell} \leq \sum_{i=0}^k u_{ij\ell} \leq (k+1)v_{j\ell}, \qquad j = 1, \ldots, N, \tag{11c}$$

$$Nw_\ell \leq \sum_{j=1}^N v_{j\ell} \leq w_\ell + N - 1, \tag{11d}$$

$$u_{ij\ell},\ v_{j\ell},\ w_\ell \in \{0, 1\}, \qquad j = 1, \ldots, N,\ i = 0, \ldots, k. \tag{11e}$$

Thus, we have the following result:

Proposition 3.1 *For the multi-objective optimization problem* (1)*, the system* (11) *of inequalities partitions the vectors* $\mathbf{x}^\ell \in X,\ \ell = 1, \ldots, N$*, into a Pareto optimal and a non-Pareto optimal set.*

Remark 3.2 *The strict inequalities in* (11a) *can be replaced by (non-strict) inequalities if a positive constant*

$$\upsilon = \min\left\{|f_i(\mathbf{x}^j) - f_i(\mathbf{x}^\ell)| :\ i \in \{1, \ldots, k\},\ j, \ell \in \{1, \ldots, N\},\ |f_i(\mathbf{x}^j) - f_i(\mathbf{x}^\ell)| > 0\right\},$$

is added on the left-hand side of the respective inequality. The same is true with analogous definitions for the strict inequality in (11b) *and for the strict inequalities in the following inequality systems.*

[1] Unlike in Proposition 1.5 we here use constraints "in both directions". The reason is that (4c) is not required here, whence $w_\ell = 0$ is feasible for a Pareto optimal point with $u_{ij\ell} = 0,\ \forall ij\ell$, if not the "extra" constraints are added. See further Remark 3.3.

Remark 3.3 *If the goal is just to find the Pareto optimal set $\mathcal{P} \subseteq X$, half of the inequalities in* (11) *can be dropped and, as in* (12) *below, be replaced by an objective function.*

$$\begin{aligned}
&\text{maximize} && \sum_{\ell=1}^{N} w_\ell, \\
&\text{subject to} && f_i(\mathbf{x}^\ell) - f_i(\mathbf{x}^j) < M(1-u_{ij\ell}), \quad j,\ell = 1,\dots,N,\ i = 1,\dots,k, && \text{(12a)}\\
& && \sum_{i=1}^{k} f_i(\mathbf{x}^\ell) - \sum_{i=1}^{k} f_i(\mathbf{x}^j) \le M(1-u_{0j\ell}), \quad j,\ell = 1,\dots,N, && \text{(12b)}\\
& && v_{j\ell} \le \sum_{i=0}^{k} u_{ij\ell}, \qquad j,\ell = 1,\dots,N, && \text{(12c)}\\
& && Nw_\ell \le \sum_{j=1}^{N} v_{j\ell}, \qquad \ell = 1,\dots,N, && \text{(12d)}\\
& && u_{ij\ell},\ v_{j\ell},\ w_\ell \in \{0,1\}, \qquad j,\ell = 1,\dots,N,\ i = 0,\dots,k. && \text{(12e)}
\end{aligned}$$

3.2 The explicit formulation of Pareto optimality applied to the reduced problem

There are 2^k possible objectives in the reduced problem; the ones chosen are indicated by the values of $\beta_p,\ p = 1,\dots,2^k$. However, we are interested in τ-Pareto optimality for the reduced problem, wherefore $\{f_1,\dots,f_k\}$ in (11) cannot be directly replaced with $\{\beta_1 g_1,\dots,\beta_{2^k} g_{2^k}\}$. The reason for this is that if some objective function h_i maps all $\mathbf{x} \in X$ to the same value, then for any $\mathbf{x}^* \in X$ there exists no $\mathbf{x} \in X$ such that $h_i(\mathbf{x}) + \tau \le h_i(\mathbf{x}^*)$ if $\tau > 0$; thus all $\mathbf{x} \in X$ will be τ-Pareto-optimal. This will be the case for every objective function corresponding to a $\beta_p = 0$ (since if so, $\beta_p g_p \equiv 0$). Therefore the system (11) must be modified so that all inequalities involving terms of an objective g_p such that $\beta_p = 0$ become redundant. We redefine the u, v and w

variables according to

$$u_{pj\ell} = \begin{cases} 1, & \text{if } g_p(\mathbf{x}^\ell) < g_p(\mathbf{x}^j) + \tau, \text{ and collection } p \text{ is chosen,} \\ 0, & \text{otherwise,} \end{cases} \qquad j, \ell = 1, \ldots, N,\ p = 1, \ldots, 2^k,$$

$$u_{0j\ell} = \begin{cases} 1, & \text{if } \sum_{p=1}^{2^k} \beta_p g_p(\mathbf{x}^\ell) \leq \sum_{p=1}^{2^k} (\beta_p g_p(\mathbf{x}^j) + \tau), \\ 0, & \text{otherwise,} \end{cases} \qquad j, \ell = 1, \ldots, N,$$

$$v_{j\ell} = \begin{cases} 1, & \text{if } \mathbf{x}^j \text{ does not } \tau\text{-dom. } \mathbf{x}^\ell \text{ (i.e., if } \sum_{p=0}^{2^k} u_{pj\ell} \geq 1), \\ 0, & \text{if } \mathbf{x}^j \ \tau\text{-dominates } \mathbf{x}^\ell, \end{cases} \qquad j, \ell = 1, \ldots, N,$$

$$w_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in \mathcal{P}_\tau^\beta \text{ (i.e., if } v_{j\ell} = 1\ \forall j), \\ 0, & \text{if } \mathbf{x}^\ell \notin \mathcal{P}_\tau^\beta, \end{cases} \qquad \ell = 1, \ldots, N.$$

For $\ell = 1, \ldots, N$, the system (13) has a solution with $w_\ell = 1$ if and only if $\mathbf{x}^\ell \in \mathcal{P}_\tau^\beta$. Observe that $\beta_p,\ p = 1, \ldots, 2^k$, are constants, not variables, in (13).

$$-Mu_{pj\ell} \leq M(1 - \beta_p) + \beta_p g_p(\mathbf{x}^\ell) - \beta_p g_p(\mathbf{x}^j) - \tau < 2M(1 - u_{pj\ell}), \qquad j = 1, \ldots, N,\ p = 1, \ldots, 2^k, \quad \text{(13a)}$$

$$-Mu_{0j\ell} < \sum_{p=1}^{2^k} \beta_p g_p(\mathbf{x}^\ell) - \sum_{p=1}^{2^k} \left(\beta_p g_p(\mathbf{x}^j) + \tau\right) \leq M(1 - u_{0j\ell}), \qquad j = 1, \ldots, N, \quad \text{(13b)}$$

$$v_{j\ell} \leq \sum_{p=0}^{2^k} u_{pj\ell} \leq (r+1)v_{j\ell}, \qquad j = 1, \ldots, N, \quad \text{(13c)}$$

$$Nw_\ell \leq \sum_{j=1}^{N} v_{j\ell} \leq w_\ell + N - 1, \quad \text{(13d)}$$

$$u_{pj\ell},\ v_{j\ell},\ w_\ell \in \{0, 1\}, \qquad j = 1, \ldots, N,\ p = 0, \ldots, 2^k. \quad \text{(13e)}$$

The main difference between the systems (11) and (13) is that the parameter τ is introduced in (13) to define the τ-Pareto optimal set. Note also the difference between the inequalities (11a) and (13a), where terms are added to make sure that $u_{pj\ell} = 0$ whenever $\beta_p = 0$. Also, in inequality (13c), the constant $k + 1$ from (11c) is replaced by $r + 1$; this number is large enough since no more than $r + 1$ of the u-variables can take the value 1. From the arguments above we have the following result.

Proposition 3.4 *Let the set $\{g_1, \ldots, g_{2^k}\}$ consist of the potential objective functions for a modified version of the multi-objective optimization problem* (1) *and let $\boldsymbol{\beta} \in \{0, 1\}^{2^k}$ be the unknown binary vector indicating which of the (at most r) objectives that are used. Then the vectors $\mathbf{x}^\ell \in X,\ \ell = 1, \ldots, N$, are partitioned into a τ-Pareto optimal and a non-τ-Pareto optimal set by the system of inequalities* (13).

3.3 An explicit formulation of centrality

The objective function $d_H(\mathbf{f}(\mathcal{P}^\rho), \mathbf{f}(\mathcal{P}_\tau^{\beta,\rho}))$ to minimize over the decision variables $\boldsymbol{\beta}$ and τ depends on the centrality parameter ρ. Therefore, centrality according to Definition 2.4 must also be characterized explicitly using constraints. In the following sections we derive a system of linear inequalities which partitions the set X into a ρ-central and a non-ρ-central part. By assuming that the objective functions $f_i,\ i = 1, \ldots, k$, are non-negative (i.e., $\mathbf{f}(X) \subseteq \Re_+^k$), Proposition 3.5 guarantees the existence of a linear inequality system characterizing ρ-centrality. Given that $\mathcal{P}$ has a spreading in each dimension, i.e., $z_i^{\text{nad}} > z_i^{\text{ideal}},\ i = 1, \ldots, k$, the assumption is valid for the resulting functions when replacing each $f_i(\mathbf{x}),\ i = 1, \ldots, k$, with

$$\frac{f_i(\mathbf{x}) - z_i^{\text{ideal}}}{z_i^{\text{nad}} - z_i^{\text{ideal}}}. \tag{14}$$

A similar characterization to that in Proposition 3.5 can be formulated without the assumption of non-negativity (through addition of more large constants); however, the scaling (14) may also be a numerically wise choice.

Proposition 3.5 *In the problem* (1)*, suppose that* $f_i : X \to \Re_+,\ i = 1, \ldots, k$*. Let* $\mathcal{P} \subseteq X$ *be indicated by the binary variables* $w_\ell = 1$ *if and only if* $\mathbf{x}^\ell \in \mathcal{P}$ *and let* $\rho \in [0, 1]$*. Then a vector* $\mathbf{x}^\ell \in X$ *is* ρ*-central if and only if* $\exists j \in \{1, \ldots, n\}$ *such that*

$$f_i(\mathbf{x}^\ell) \leq (1-\rho) w_j f_i(\mathbf{x}^j) + \rho f_i(\mathbf{x}^m) + M(1 - w_m), \quad i = 1, \ldots, k,\ m = 1, \ldots, N. \tag{15}$$

Proof. First, observe that the definition (9) of centrality can be rewritten as follows: For a vector $\mathbf{x}^\ell \in X$ to be ρ-central, a necessary and sufficient condition is that for each $i = 1, \ldots, k$, the inequality

$$f_i(\mathbf{x}^\ell) \ \leq \ (1-\rho) \max_{j:\ w_j = 1} f_i(\mathbf{x}^j) + \rho \min_{m:\ w_m = 1} f_i(\mathbf{x}^m), \tag{16}$$

holds. Now, (15)⇔(16) by the following arguments, in which the terms involving $\mathbf{x}^j$ and the terms involving $\mathbf{x}^m$ are studied separately.

- ⇐ If (16) holds for $\hat{j} \in \arg\max\{f_i(\mathbf{x}^j) \mid w_j = 1\}$ then (15) holds for at least one $j \in \{1, \ldots, N\}$. If (16) holds for $\hat{m} \in \arg\min\{f_i(\mathbf{x}^m) \mid w_m = 1\}$ then (15) holds for all $m = 1, \ldots, N$, since for all m such that $w_m = 1$, $f_i(\mathbf{x}^m) \geq \min_{m':\ w_{m'}=1} f_i(\mathbf{x}^{m'})$ and also it holds whenever $w_m = 0$.
- ⇒ If (15) holds for some $j \in \{1, \ldots, N\}$, it must hold for $\hat{j} \in \arg\max\{f_i(\mathbf{x}^j);\ w_j = 1\}$ since for all j, $\max_{j':\ w_{j'}=1} f_i(\mathbf{x}^{j'}) \geq w_j f_i(\mathbf{x}^j)$. For m, the right-hand side of (15) is smallest for the $\hat{m}$ corresponding to a minimal $f_i(\mathbf{x}^m)$ with $w_m = 1$. If the inequality holds for all $m = 1, \ldots, N$, it holds for $\hat{m}$ and thus (16) holds. ∎

We introduce a number of binary variables with the aim of creating a system of linear inequalities whose feasible solutions are such that a binary variable a_ℓ equals 1 if and only if $\mathbf{x}^\ell \in X$ is ρ-central (i.e., if $\mathbf{x}^\ell \in X^\rho$), $\ell = 1, \ldots, N$. Let

$$b_{ijm\ell} = \begin{cases} 1, & \text{if } f_i(\mathbf{x}^\ell) \le (1-\rho)w_j f_i(\mathbf{x}^j) + \rho f_i(\mathbf{x}^m) + M(1-w_m), \\ 0, & \text{otherwise,} \end{cases} \quad j, m, \ell = 1, \ldots, N,\ i = 1, \ldots, k,$$

$$c_{ij\ell} = \begin{cases} 1, & \text{if } \sum_{m=1}^{N} b_{ijm\ell} \ge N, \\ 0, & \text{otherwise,} \end{cases} \quad j, \ell = 1, \ldots, N,\ i = 1, \ldots, k,$$

$$e_{i\ell} = \begin{cases} 1, & \text{if } \sum_{j=1}^{N} c_{ij\ell} \ge 1, \\ 0, & \text{otherwise,} \end{cases} \quad \ell = 1, \ldots, N,\ i = 1, \ldots, k,$$

$$a_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in X^\rho \text{ (i.e., if } \sum_{i=1}^{k} e_{i\ell} \ge k), \\ 0, & \text{if } \mathbf{x}^\ell \notin X^\rho, \end{cases} \quad \ell = 1, \ldots, N.$$

The variable declarations above are equivalent to the inequality system (17) below, which for each $\ell = 1, \ldots, N$, has a solution with $a_\ell = 1$ if and only if $\mathbf{x}^\ell \in X^\rho$.

$$-2Mb_{ijm\ell} < f_i(\mathbf{x}^\ell) - \left((1-\rho)w_j f_i(\mathbf{x}^j) + \rho f_i(\mathbf{x}^m) + M(1-w_m)\right), \quad j, m, \ell = 1, \ldots, N,\ i = 1, \ldots, k, \tag{17a}$$

$$M(1-b_{ijm\ell}) \ge f_i(\mathbf{x}^\ell) - \left((1-\rho)w_j f_i(\mathbf{x}^j) + \rho f_i(\mathbf{x}^m) + M(1-w_m)\right), \quad j, m, \ell = 1, \ldots, N,\ i = 1, \ldots, k, \tag{17b}$$

$$Nc_{ij\ell} \le \sum_{m=1}^{N} b_{ijm\ell} \le c_{ij\ell} + N - 1, \qquad j, \ell = 1, \ldots, N,\ i = 1, \ldots, k, \tag{17c}$$

$$e_{i\ell} \le \sum_{j=1}^{N} c_{ij\ell} \le Ne_{i\ell}, \qquad \ell = 1, \ldots, N,\ i = 1, \ldots, k, \tag{17d}$$

$$ka_\ell \le \sum_{i=1}^{k} e_{i\ell} \le a_\ell + k - 1, \qquad \ell = 1, \ldots, N, \tag{17e}$$

$$b_{ijm\ell},\ c_{ij\ell},\ e_{i\ell},\ a_\ell \in \{0, 1\}, \qquad j, m, \ell = 1, \ldots, N,\ i = 1, \ldots, k. \tag{17f}$$

Proposition 3.6 *For a multi-objective optimization problem* (1) *with non-negative objective functions, the vectors* $\mathbf{x}^\ell \in X,\ \ell = 1, \ldots, N$, *are partitioned into a* ρ*-central and a non-*ρ*-central part by the system of inequalities* (17).

Combining the partitioning systems (11) and (17) and introducing the binary variables

$$\eta_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in \mathcal{P}^\rho, \\ 0, & \text{if } \mathbf{x}^\ell \notin \mathcal{P}^\rho, \end{cases} \qquad \ell = 1, \ldots, N,$$

and the constraints

$$w_\ell + a_\ell - 1 \leq \eta_\ell \leq w_\ell + a_\ell, \qquad \ell = 1, \ldots, N, \tag{18a}$$

$$\eta_\ell \in \{0, 1\}, \qquad \ell = 1, \ldots, N, \tag{18b}$$

finally yields a system with feasible solutions such that $\eta_\ell = 1$ for vectors $\mathbf{x}^\ell \in X$ that are both Pareto optimal and ρ-central ($\mathbf{x}^\ell \in \mathcal{P}^\rho$) and $\eta_\ell = 0$ for vectors $\mathbf{x}^\ell \in X$ that are not ($\mathbf{x}^\ell \in X \setminus \mathcal{P}^\rho$).

3.4 The explicit formulation of centrality applied to the reduced problem

We wish to decide which vectors $\mathbf{x} \in X$ that are ρ-central with respect to the (unknown) set of objectives $\{g_{s_1}, \ldots, g_{s_r}\}$. Let us denote this set by $X^{\beta,\rho} \subseteq X$. The construction of centrality in Definition 2.4 implies that we can instead set up the centrality inequalities for the (known) set of objectives $\{\beta_1 g_1, \ldots, \beta_{2^k} g_{2^k}\}$, since for all p with $\beta_p = 0$ the corresponding inequality in (9) is fulfilled. However, it is not possible to use the same variable definitions, since that would lead to a non-linearity in the constraints due to the multiplication of w and β (in the inequality corresponding to (17a)). Instead, we rewrite the inequality (15), equivalent to the definition of ρ-centrality, as follows:

$$f_i(\mathbf{x}^\ell) \leq (1-\rho) f_i(\mathbf{x}^j) - M(1 - w_j) + \rho f_i(\mathbf{x}^m) + M(1 - w_m), \tag{19}$$

which should hold for all $i = 1, \ldots, k$, all $m = 1, \ldots, N$, and for at least one $j \in \{1, \ldots, N\}$. For the partially weighted objectives the inequality (19) becomes

$$\beta_p g_p(\mathbf{x}^\ell) \leq (1-\rho)\beta_p g_p(\mathbf{x}^j) - M(1 - w_j) + \rho\beta_p g_p(\mathbf{x}^m) + M(1 - w_m), \tag{20}$$

and should hold for all $p = 1, \ldots, 2^k$, all $m = 1, \ldots, N$, and for at least one $j \in \{1, \ldots, N\}$.

As in Section 3.3, we use the following variable declarations for $p = 1, \ldots, 2^k$, and $j, m, \ell = 1, \ldots, N$. The first definition is modified in order to avoid the non-linearity, whereas the other three are analogous:

$$b_{pjm\ell} = \begin{cases} 1, & \text{if } \beta_p g_p(\mathbf{x}^\ell) \leq (1-\rho)\beta_p g_p(\mathbf{x}^j) - M(1-w_j) + \rho\beta_p g_p(\mathbf{x}^m) + M(1-w_m), \\ 0, & \text{otherwise,} \end{cases}$$

$$c_{pj\ell} = \begin{cases} 1, & \text{if } \sum_{m=1}^{N} b_{pjm\ell} \geq N, \\ 0, & \text{otherwise,} \end{cases}$$

$$e_{p\ell} = \begin{cases} 1, & \text{if } \sum_{j=1}^{N} c_{pj\ell} \geq 1, \\ 0, & \text{otherwise,} \end{cases}$$

$$a_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in X^{\beta,\rho} \text{ (i.e., if } \sum_{p=1}^{2^k} e_{p\ell} \geq 2^k), \\ 0, & \text{if } \mathbf{x}^\ell \notin X^{\beta,\rho}. \end{cases}$$

For the partially weighted objectives $\{\beta_1 g_1, \ldots, \beta_{2^k} g_{2^k}\}$ the centrality system corresponding to (17) is then expressed as:

$$-2Mb_{pjm\ell} < \beta_p g_p(\mathbf{x}^\ell) - \big((1-\rho)\beta_p g_p(\mathbf{x}^j) - M(1-w_j) + \rho\beta_p g_p(\mathbf{x}^m) + M(1-w_m)\big), \quad j,m,\ell = 1,\ldots,N,\ p = 1,\ldots,2^k, \quad (21a)$$

$$M(1-b_{pjm\ell}) \geq \beta_p g_p(\mathbf{x}^\ell) - \big((1-\rho)\beta_p g_p(\mathbf{x}^j) - M(1-w_j) + \rho\beta_p g_p(\mathbf{x}^m) + M(1-w_m)\big), \quad j,m,\ell = 1,\ldots,N,\ p = 1,\ldots,2^k, \quad (21b)$$

$$Nc_{pj\ell} \leq \sum_{m=1}^{N} b_{pjm\ell} \leq c_{pj\ell} + N - 1, \qquad j,\ell = 1,\ldots,N,\ p = 1,\ldots,2^k, \quad (21c)$$

$$e_{p\ell} \leq \sum_{j=1}^{N} c_{pj\ell} \leq N e_{p\ell}, \qquad \ell = 1,\ldots,N,\ p = 1,\ldots,2^k, \quad (21d)$$

$$2^k a_\ell \leq \sum_{p=1}^{2^k} e_{p\ell} \leq a_\ell + 2^k - 1, \qquad \ell = 1,\ldots,N, \quad (21e)$$

$$b_{pjm\ell},\ c_{pj\ell},\ e_{p\ell},\ a_\ell \in \{0,1\}, \qquad j,m,\ell = 1,\ldots,N\ p = 1,\ldots,2^k. \quad (21f)$$

As before, by combining the systems (13) and (21) and introducing the binary variables

$$\eta_\ell = \begin{cases} 1, & \text{if } \mathbf{x}^\ell \in \mathcal{P}_\tau^{\beta,\rho}, \\ 0, & \text{if } \mathbf{x}^\ell \notin \mathcal{P}_\tau^{\beta,\rho}, \end{cases} \qquad \ell = 1,\ldots,N,$$

and the constraints

$$w_\ell + a_\ell - 1 \leq \eta_\ell \leq w_\ell + a_\ell, \qquad \ell = 1,\ldots,N, \quad (22a)$$

$$\eta_\ell \in \{0,1\}, \qquad \ell = 1,\ldots,N, \quad (22b)$$

a consistent system is formed such that $\eta_\ell = 1$ if and only if a vector $\mathbf{x}^\ell \in X,\ \ell = 1,\ldots,N$, is ρ-central and τ-Pareto optimal in the reduced problem (i.e., if $\mathbf{x}^\ell \in \mathcal{P}_\tau^{\beta,\rho}$).

3.5 Formulating the ideal model

Using the explicit formulation of Pareto optimality together with ρ-centrality, the ideal formulation of the total problem (10) can now be stated. Assume that the number of ρ-central Pareto optimal points in the original problem (1) is Q. Then, without loss of generality we assume that $\mathcal{P}^\rho = \{\mathbf{x}^1, \ldots, \mathbf{x}^Q\}$. The pairwise distances in the objective space between all pairs of points $\mathbf{x}^q \in \mathcal{P}^\rho$ and $\mathbf{x}^\ell \in X$ are constants and are denoted by $d_{ql} = ||\mathbf{f}(\mathbf{x}^q) - \mathbf{f}(\mathbf{x}^\ell)||,\ q = 1,\ldots,Q,\ \ell = 1,\ldots,N$. Introducing an auxiliary variable $\theta \in \Re_+$ the objective in (10), $\min_{\boldsymbol{\beta},\tau} d_H(\mathbf{f}(\mathcal{P}^\rho), \mathbf{f}(\mathcal{P}_\tau^{\beta,\rho}))$ can be for-

mulated as that to

$$\begin{aligned} &\text{minimize} && \theta, \\ &\text{subject to} && \theta \geq \min_{\substack{\ell \in \{1,\dots,N\}: \\ \eta_\ell = 1}} d_{q\ell}, \quad q = 1, \dots, Q, \\ & && \theta \geq \min_{q \in \{1,\dots,Q\}} d_{q\ell}, \quad \ell \in \{1, \dots, N\} : \eta_\ell = 1, \end{aligned} \tag{23}$$

where the η-variables are implicitly depending on the decision variables $\boldsymbol{\beta}$ and τ. The problem (23) can be expressed as

$$\begin{aligned} &\text{minimize} && \theta, \\ &\text{subject to} && \theta \geq \min_{\ell \in \{1,\dots,N\}} \big((1-\eta_\ell)M + d_{q\ell}\big), \quad q = 1, \dots, Q, \\ & && \theta \geq \min_{q \in \{1,\dots,Q\}} \big(d_{q\ell} - (1-\eta_\ell)M\big), \quad \ell = 1, \dots, N, \end{aligned} \tag{24}$$

where $M \geq \max\big\{d_{q\ell} \mid q \in \{1, \dots, Q\},\ \ell \in \{1, \dots, N\}\big\}$. The min-operators can be replaced by the binary variables

$$y_{q\ell} = \begin{cases} 1, & \text{if } \theta \geq (1-\eta_\ell)M + d_{q\ell}, \\ 0, & \text{otherwise}, \end{cases} \qquad q = 1, \dots, Q,\ \ell = 1, \dots, N,$$

$$z_{q\ell} = \begin{cases} 1, & \text{if } \theta \geq d_{q\ell} - (1-\eta_\ell)M, \\ 0, & \text{otherwise}, \end{cases} \qquad q = 1, \dots, Q,\ \ell = 1, \dots, N,$$

and, finally, the complete problem (10) for the ideal model can be formulated as the binary linear program to

$$\begin{aligned} &\text{minimize} && \theta, \\ &\text{subject to} && (1-\eta_\ell)M + d_{ql} - \theta \leq 2M(1-y_{q\ell}), && q = 1, \dots, Q,\ \ell = 1, \dots, N, && \text{(25a)} \\ & && -(1-\eta_\ell)M + d_{ql} - \theta \leq M(1-z_{q\ell}), && q = 1, \dots, Q,\ \ell = 1, \dots, N, && \text{(25b)} \\ & && \sum_{\ell=1}^{N} y_{q\ell} \geq 1, && q = 1, \dots, Q, && \text{(25c)} \\ & && \sum_{q=1}^{Q} z_{q\ell} \geq 1, && \ell = 1, \dots, N, && \text{(25d)} \\ & && y_{q\ell} \in \{0,1\}, && q = 1, \dots, Q,\ \ell = 1, \dots, N, && \text{(25e)} \\ & && z_{q\ell} \in \{0,1\}, && q = 1, \dots, Q,\ \ell = 1, \dots, N, && \text{(25f)} \\ & && \tau \geq 0, && && \text{(25g)} \\ & && \boldsymbol{\beta} \text{ satisfies } (8), && && \text{(25h)} \\ & && (u, v, w, \boldsymbol{\beta}, \tau) \text{ satisfies } (13), && && \text{(25i)} \\ & && (b, c, e, a) \text{ satisfies } (21), && && \text{(25j)} \\ & && (w, a, \eta) \text{ satisfies } (22). && && \text{(25k)} \end{aligned}$$

The problem (25) has in the order of $N^3 2^k$ variables and constraints, the explicit formulation of centrality defining the magnitude. A binary linear program of such a size is impossible to solve in reasonable time for practical values of N and k. In the following section an approximate model is formulated that separates the optimization over $\boldsymbol{\beta}$ and τ, resulting in two easily solved sequential problems.

4 The approximate model

The underlying characteristic of the ideal model (25) that yields the high complexity is that the optimization is made simultaneously over $\boldsymbol{\beta}$ and τ. For this, τ-Pareto optimality as well as ρ-centrality must enter the model through constraints, as presented in Section 3. This leads to an explosion of binary variables and constraints. In an approximate model to be presented we separate the optimization over $\boldsymbol{\beta}$ and τ. The only constraints from the ideal model that are kept intact are (8). The rest of the constraints—including the large sets of constraints (13) and (21)—are or no longer necessary (and neither are the introduced variables in these systems). Instead a certain objective function is used in a first problem (cf. (I) on page 22) which is aimed at evaluating and deciding on a good choice of $\boldsymbol{\beta}$ without making use of τ. Then simple polynomial algorithms are used to find the (0-)Pareto optimal ρ-central part of X for the reduced problem and after that a second problem (cf. (II) on page 23) is solved to find the optimal value of τ given the already chosen value of $\boldsymbol{\beta}$.

Obviously, a solution found with the approximate model might not be optimal in (25), since decisions are made before all information is known, i.e., $\boldsymbol{\beta}$ is selected before it is known exactly how τ will affect the Pareto optimal set. Also, there is no measure of optimality provided for the approximate solution. However, the mechanism for selecting $\boldsymbol{\beta}$ is sensible, and the complexity of the problem formulation decreases enormously. This motivates the use of the approximate formulation.

4.1 Correlation between objectives

The *correlation coefficient* of two objective functions f_i and f_j over a set of vectors $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ is defined as

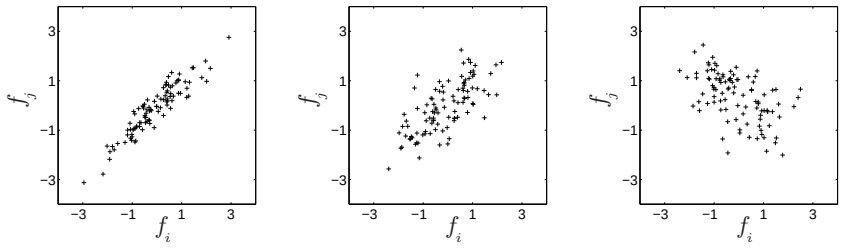
$$\hat{\rho}(f_i, f_j) = \frac{s_{ij}}{\sqrt{s_{ii} s_{jj}}} \in [-1, 1], \quad i, j \in \{1, \ldots, k\}, \tag{26}$$

where

$$s_{ij} = \frac{1}{N} \sum_{\ell=1}^{N} \left(f_i(\mathbf{x}^\ell) - \frac{1}{N} \sum_{m=1}^{N} f_i(\mathbf{x}^m) \right) \left(f_j(\mathbf{x}^\ell) - \frac{1}{N} \sum_{m=1}^{N} f_j(\mathbf{x}^m) \right), \quad i, j \in \{1, \ldots, k\}.$$

In Figure 5 we illustrate some correlations between pairs of functions using scatter plots.

The value of the pairwise correlation coefficients between two objective functions gives a measure of how similar the functions evaluate the set $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$. If



(a) $\hat{\rho}(f_i, f_j) = 0.9$ (b) $\hat{\rho}(f_i, f_j) = 0.7$ (c) $\hat{\rho}(f_i, f_j) = -0.5$

Figure 5: Illustration of correlations between a pair of functions.

the correlation is perfect, i.e., $\hat{\rho}(f_i, f_j) = 1$, then the Pareto optimal set $\mathcal{P}$ would be unchanged if the pair of objectives was replaced by its mean; one of the objectives in the pair would actually be redundant[2]. Using this reasoning we identify a *relatively redundant* objective function as a function which together with some other objective function has a large correlation coefficient. With this as a starting point we will in Section 4.2 derive the first problem to solve in the approximate model.

The objective when deciding on the optimal $\boldsymbol{\beta}$ is to maximize the least correlation coefficient between each original objective function $f_i \in \{f_1, \ldots, f_k\}$ and its most similar (in the correlation meaning) objective function in the reduced set of objectives $\{g_{s_1}, \ldots, g_{s_r}\}$. A rough illustration of this objective is given in Figure 6 by representing $\{f_1, \ldots, f_k\}$ with vectors with small angles between positively correlated f_i's and large angles between negatively correlated f_i's. The objective is then to find up to r sets in the power set of $\{1, \ldots, k\}$, each corresponding to a collection of objectives, such that to each collection there is an $f_i,\ i \in \{1, \ldots, k\}$, making an angle as small as possible with the mean of the vectors in the collection.

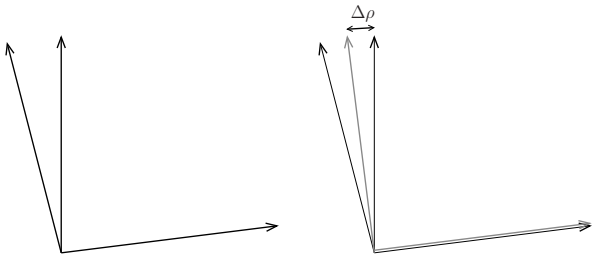


Figure 6: Original objective functions (black), optimally aggregated objective functions (gray) and a representation of the "correlation error", $\Delta\rho$.

[2]Note that $\hat{\rho}(f_i, f_j) = 1$ is not a necessary condition for $\mathcal{P}$ to remain intact; e.g., if $f_i(x) = x$ is redundant in a multi-objective optimization problem, then so is $f_j(x) = x^2$, but in general $\hat{\rho}(f_i, f_k) \neq \hat{\rho}(f_j, f_k)$.

4.2 Finding the optimal collections

In the ideal model, $\mathcal{R}$ denotes the power set of $\{1,\dots,k\}$ in which each element represents a set of original objectives that may be dropped from the system and being replaced by an aggregation of them. To reduce the computational burden of the model, we introduce a threshold, α, and only allow collections of original objectives such that all the correlation coefficients between the original objectives in the collection and its aggregated objective are larger than α. For any $\alpha \in [-1,1]$ we then have the restricted set of collection candidates

$$\mathcal{R}^\alpha = \left\{ \tilde{R} \in \mathcal{R} \;\middle|\; \hat{\rho}(f_i, f_j) \geq \alpha,\ \forall i,j \in \tilde{R} \right\}. \tag{27}$$

As before we want to reduce the set of original objectives $\{f_1,\dots,f_k\}$ to the smaller set $\{g_{s_1},\dots,g_{s_r}\}$, where each $g_{s_j} = \frac{1}{|R_{s_j}|}\sum_{i\in R_{s_j}} f_i$, i.e., equal to the mean of the original functions in the collection $R_{s_j} \in \mathcal{R}^\alpha = \{R_1^\alpha,\dots,R_{|\mathcal{R}^\alpha|}^\alpha\}$. As in Section 2 the decision vector $\boldsymbol{\beta} \in \{0,1\}^{|\mathcal{R}^\alpha|}$ indicates which collections to choose and the incidence matrix $A \in \mathbb{B}^{k\times|\mathcal{R}^\alpha|}$ defines which original objectives that belong to each collection.

Let $\psi_p := \min_{i\in R_p} \hat{\rho}(g_p, f_i)$, and introduce an auxiliary variable $z \in \Re$ and a scalar $M \geq 2$. The problem of choosing the best collections then is the binary linear program to

$$\begin{aligned}
\text{maximize} \quad & z, \\
\text{such that} \quad & z \leq \psi_p + (1-\beta_p)M, \quad p \in \mathcal{R}^\alpha, \\
& A\boldsymbol{\beta} \geq \mathbf{1}^k, \\
& \boldsymbol{\beta}^T \mathbf{1}^{|\mathcal{R}^\alpha|} \leq r, \\
& \boldsymbol{\beta} \in \{0,1\}^{|\mathcal{R}^\alpha|}.
\end{aligned} \tag{I}$$

This problem, which is close to a set partitioning-problem [19] has, for $\alpha = -1$, 2^k binary variables. However, the number of variables can be reduced substantially with a larger value of α, whence the complexity of (I) will not be a serious issue.

An observation to be made is given in Proposition 4.1, whose result is quite obvious from Figure 6.

Proposition 4.1 *Let the set of optimal solutions to* (I) *be denoted by $\mathcal{B}$. Then for some $\boldsymbol{\beta}^* \in \mathcal{B}$, exactly r objectives are used, i.e, $\sum_{p\in\mathcal{R}^\alpha} \beta_p^* = r$.*

Proof. Assume the result does not hold, i.e., $\nexists \boldsymbol{\beta}^* \in \mathcal{B}$ such that $\sum_{p\in\mathcal{R}^\alpha} \beta_p^* = r$. Let $\boldsymbol{\beta}$ be optimal to (I) with $\sum_{p\in\mathcal{R}^\alpha} \beta_p < r$ and assume that a collection p corresponding to $z = \psi_p$ consists of $\{f_{i_1},\dots,f_{i_t}\}$. We now extend the collection to $\{f_{i_1},\dots,f_{i_t},f_{i_{t+1}}\}$, which leads to that ψ_p will not increase. Since we are still feasible and thus optimal, the result follows by contradiction. ∎

4.3 Finding the optimal dominance tolerance

Solving (I) yields the optimal solution $\boldsymbol{\beta} = \boldsymbol{\beta}^*$, which defines the objectives to use in the reduced problem. Observe that we by optimal mean with respect to the approximate model, $\boldsymbol{\beta}^*$ is sub-optimal with respect to (25). By pairwise comparisons between all the points in X, the Pareto optimal set with respect to the selected objectives, $\mathcal{P}^{\beta^*}$, is then extracted. Further, by checking the ρ-centrality characterization from Definition 2.4 for all vectors $\mathbf{x} \in \mathcal{P}^{\beta^*}$ the ρ-central part $\mathcal{P}^{\beta^*,\rho} \subseteq \mathcal{P}^{\beta^*}$ is extracted. The idea is then in a second problem (II) , starting from $\mathcal{P}^{\beta^*,\rho} (= \mathcal{P}^{\beta^*,\rho}_{\tau=0})$, to increase the value of the tolerance parameter τ and find the value τ^* that minimizes the Hausdorff distance between $\mathbf{f}(\mathcal{P}^{\rho})$ and $\mathbf{f}(\mathcal{P}^{\beta^*,\rho}_{\tau})$, i.e., the second problem is that to

$$\begin{aligned} \underset{\tau}{\text{minimize}} \quad & d_H(\mathbf{f}(\mathcal{P}^{\rho}), \mathbf{f}(\mathcal{P}^{\beta^*,\rho}_{\tau})), \\ \text{such that} \quad & \tau \geq 0. \end{aligned} \tag{II}$$

In Figure 7 the solution process of the approximate model is illustrated. Included in the figure are the inputs and outputs of the two problems (I) and (II).

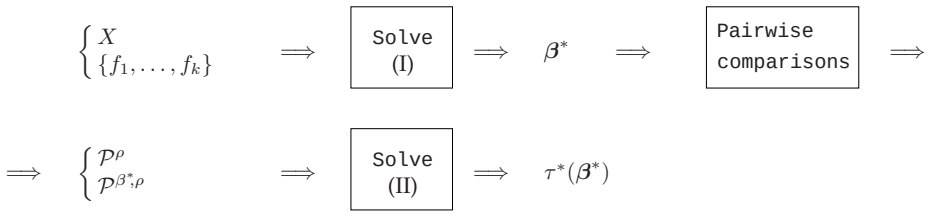


Figure 7: Solution procedure for the approximate model showing the inputs and outputs of the two sequential problems.

Different values of τ changes the Pareto set that ρ-centrality is relating to in Definition 2.4. This implies a difficulty since the value of τ will not be known until after problem (II) has been solved. The ρ-centrality partitioning could be done in each iteration in the solution process for (II) (cf. Algorithm 4.1), but to get an easy optimization problem for τ (which will be presented in the next section) we make another approximation. We fix the set to relate to for ρ-centrality by approximating $\mathcal{P}^{\beta^*}_{\tau}$—the set we ideally want to use—by $\mathcal{P}$ in the ρ-centrality partitioning, i.e., we use (cf. Definition 2.4)

$$\mathcal{P}^{\beta^*,\rho}_{\tau} :\approx \left\{ \mathbf{x} \in \mathcal{P}^{\beta^*}_{\tau} \;\middle|\; g_p(\mathbf{x}) \leq (1-\rho) \max_{\mathbf{y} \in \mathcal{P}} g_p(\mathbf{y}) + \rho \min_{\mathbf{y} \in \mathcal{P}} g_p(\mathbf{y}),\ \forall p \in \mathcal{R}^{\alpha} \right\}. \tag{28}$$

From Section 2 it is clear that $\mathcal{P}^{\beta^*} \subseteq \mathcal{P}$. It is also clear from Definition 2.4 that $\mathcal{P}^{\rho} \subseteq \mathcal{P}$. From our decision to let the centrality operator relate to $\mathcal{P}$ when deciding on $\mathcal{P}^{\beta^*,\rho}$ we can also conclude that $\mathcal{P}^{\beta^*,\rho} \subseteq \mathcal{P}^{\rho}$, since $\mathcal{P}^{\beta^*,\rho} = \mathcal{P}^{\beta^*} \cap X^{\rho} \subseteq \mathcal{P} \cap X^{\rho} = \mathcal{P}^{\rho}$. What is not obvious is the first equality, which does not hold without our approximation (28)

since X^ρ depends on the objective functions. The relations are, in the objective space, illustrated in Figure 8.

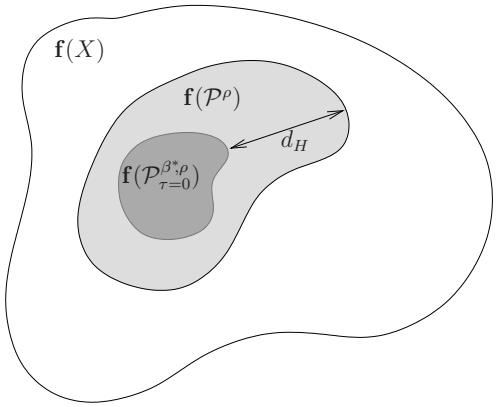


Figure 8: Originating from a subset of $\mathcal{P}^\rho$, problem (II) enlarges the inner set by increasing the value of τ.

Remember the definition of Hausdorff distance between two sets A and B as being the maximum of the largest distance from any point in A to its nearest point (with respect to some distance metric $d(\cdot,\cdot)$) in B and the largest distance from any point in B to its nearest point in A. We define $\phi_1(\tau)$ as the maximum distance from a point in $\mathcal{P}^\rho$ to its nearest point in $\mathcal{P}^{\beta^*,\rho}_\tau$ and $\phi_2(\tau)$ as the maximum distance from a point in $\mathcal{P}^{\beta^*,\rho}_\tau$ to its nearest point in $\mathcal{P}^\rho$, i.e.,

$$\phi_1(\tau) = \max_{\mathbf{x}\in\mathcal{P}^\rho}\Big(\min_{\mathbf{y}\in\mathcal{P}^{\beta^*,\rho}_\tau} d\big(\mathbf{f}(\mathbf{x}),\mathbf{f}(\mathbf{y})\big)\Big), \tag{29}$$

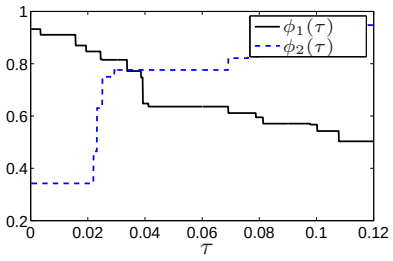
$$\phi_2(\tau) = \max_{\mathbf{y}\in\mathcal{P}^{\beta^*,\rho}_\tau}\Big(\min_{\mathbf{x}\in\mathcal{P}^\rho} d\big(\mathbf{f}(\mathbf{y}),\mathbf{f}(\mathbf{x})\big)\Big). \tag{30}$$

Then, our objective function in (10) to minimize can, for a fixed $\boldsymbol{\beta}^*$, be rewritten as

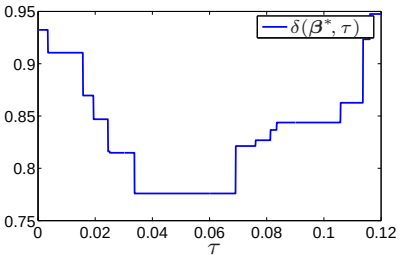
$$d_H(\mathbf{f}(\mathcal{P}^\rho),\mathbf{f}(\mathcal{P}^{\beta^*,\rho}_\tau)) = \delta(\boldsymbol{\beta}^*,\tau) = \max\left\{\phi_1(\tau),\phi_2(\tau)\right\}. \tag{31}$$

Since increasing the value of τ will increase the size of $\mathcal{P}^{\beta^*,\rho}_\tau$ and thus there will be more alternate points that can be nearest for each point in $\mathcal{P}^\rho$, $\phi_1(\tau)$ is a monotonically decreasing lower semi-continuous function. Conversely, $\phi_2(\tau)$ describes a monotonically increasing upper semi-continuous function. The sets $\mathcal{P}^\rho$ and $\mathcal{P}^{\beta^*,\rho}_\tau$ are discrete whence the functions $\phi_1(\tau)$ and $\phi_2(\tau)$ are piecewise constant. Thus, $\delta(\boldsymbol{\beta}^*,\tau)$ is a piecewise constant quasi-convex [2] function. Figure 9 illustrates the functions $\phi_1(\tau),\phi_2(\tau),\delta(\boldsymbol{\beta}^*,\tau)$ and the monotonically increasing upper semi-continuous function

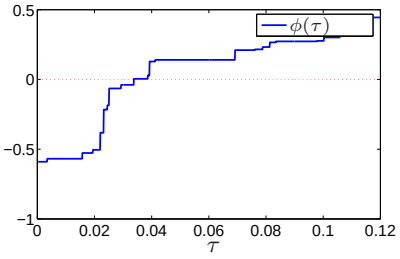
$$\phi(\tau) := \phi_2(\tau) - \phi_1(\tau). \tag{32}$$



(a) $\phi_1(\tau)$ and $\phi_2(\tau)$



(b) $\delta(\boldsymbol{\beta}^*, \tau)$



(c) $\phi(\tau)$

Figure 9: Typical appearances of the three functions (a) $\phi_1(\tau)$ and $\phi_2(\tau)$ from (29), (30), (b) $\delta(\boldsymbol{\beta}^*, \tau)$ from (31) and (c) $\phi(\tau)$ from (32). For visibility reasons, the graphs of all functions have been closed.

We are interested in finding

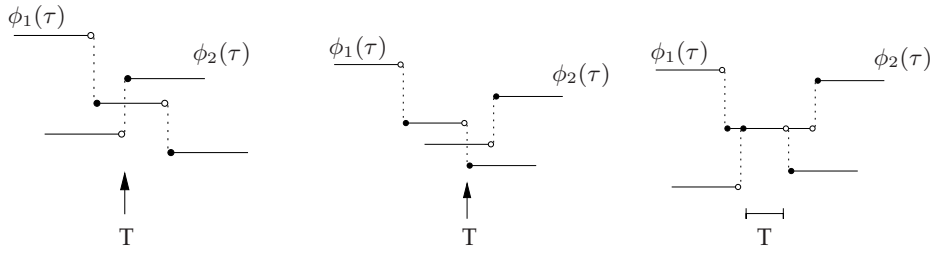
$$\tau^* \in \arg\min_{\tau} \ \delta(\boldsymbol{\beta}^*, \tau). \tag{33}$$

Assuming that $\phi_1(0) > \phi_2(0)$ (otherwise $\tau^* = 0$), the function $\phi(\tau)$ can possess three different characteristics around the value where it changes sign. We will demonstrate that no matter which of the characteristics, a bisection technique can be defined to solve (II), i.e., to find a τ^* solving (33). Let

$$T = \arg\min_{\tau} \{\phi(\tau) \mid \phi(\tau) \geq 0\}. \tag{34}$$

There can be three possible cases as presented below and illustrated in Figure 10.

$$\begin{aligned} &1) \ T \text{ is a singleton and } \phi_1(T) = \min_{\tau<T} \{\phi_1(\tau)\}, \\ &2) \ T \text{ is a singleton and } \phi_1(T) < \min_{\tau<T} \{\phi_1(\tau)\}, \\ &3) \ T \text{ is an interval.} \end{aligned} \tag{35}$$



(a) Case 1 (b) Case 2 (c) Case 3

Figure 10: An illustration of the three cases in (35).

For $\varepsilon > 0$ sufficiently small, we can in each of the three cases define an interval $\tilde{T} \subseteq \arg\min_\tau \ \delta(\boldsymbol{\beta}^*, \tau)$ according to:

$$
\begin{aligned}
&1) \ \ \tilde{T} = [T - \varepsilon, T), \\
&2) \ \ \tilde{T} = [T, T + \varepsilon], \\
&3) \ \ \tilde{T} = T.
\end{aligned}
$$

Hence it suffices to find one $\tau^* \in \tilde{T}$. This is done using a simple bisection technique on $\phi(\tau)$ where a termination criterion is set on the maximum size of the remaining interval with the parameter $\varepsilon^{\text{term}}$. In principle, since X is a discrete set of points, there exists a positive value of $\varepsilon^{\text{term}}$ that guarantees an optimal solution to (II) using the bisection technique, however this value is a priori unknown[3]. The algorithm in pseudo-code is given in Algorithm 4.1. The first output from the algorithm is τ^*, the (approximate) optimal value of the tolerance parameter τ together with the the already chosen value of $\boldsymbol{\beta}^*$ from Section 4.2. The other outputs are $\delta(\boldsymbol{\beta}^*, \tau^*(\boldsymbol{\beta}^*))$, the Hausdorff distance given the chosen τ^*, and $error^{\text{ub}}$, the maximum error in $\delta(\boldsymbol{\beta}^*, \tau)$ due to a possibly too large choice of $\varepsilon^{\text{term}}$.

Of course, instead of solving over τ for only one value of $\boldsymbol{\beta}$, it is possible to create a pool of candidate $\boldsymbol{\beta}$'s (e.g. by solving the program (I) repeatedly, and in each iteration add a constraint cutting away the previous solution) and compute $\tau^*(\boldsymbol{\beta})$ for each $\boldsymbol{\beta}$ in the pool. By comparing $\delta(\boldsymbol{\beta}, \tau^*(\boldsymbol{\beta}))$ for all $\boldsymbol{\beta}$'s in the pool, the pair $(\boldsymbol{\beta}, \tau)$ yielding the lowest value of $\delta(\boldsymbol{\beta}, \tau)$ can be selected as an approximate solution to the main reduction problem (10). By increasing the pool, the objective function value decreases monotonically, and for a sufficiently large pool (given that $\varepsilon^{\text{term}}$ is not chosen too small), the optimal solution to (10) will be found.

[3] A lower bound on the, for optimality necessary, value of $\varepsilon^{\text{term}}$ is given by $\bar{\varepsilon} = \min_{s,t}\{|\mathcal{G}(s) - \mathcal{G}(t)| : |\mathcal{G}(s) - \mathcal{G}(t)| > 0\}$, where $\mathcal{G}$ is the union of the function values of all objectives in the reduced problem, i.e., $\mathcal{G} = \cup_{i=1,\ldots,r} g_i(X)$. The explanation behind this expression is that the lengths of the piecewise constant segments of $\phi(\tau)$ are at most $\bar{\varepsilon}$ since the set $\mathbf{f}(\mathcal{P}_\tau^{\boldsymbol{\beta}^*,\rho})$ will not, from the Definition 2.6 of τ-Pareto optimality, change more than once if τ is increased with $\bar{\varepsilon}$. However, the tightness of the upper bound is unknown, whence it is not used in the algorithm.

Algorithm 4.1 Pseudo-code for the τ-optimization algorithm.

```
pick tau_1 = 0
pick tau_2 big enough s.t. phi(tau_2) > 0
until tau_2 - tau_1 < e^term {
    let tau_3 = (tau_1 + tau_2)/2
    if phi(tau_3) < 0 {
        let tau_1 = tau_3
        }
    elseif phi(tau_3) > 0 {
        let tau_2 = tau_3
        }
    else {
        output tau_star = tau_3
        delta_star = phi1(tau_star)
        error^ub = 0
        }
    }
output tau_star = argmin_{tau_1,tau_2} max(phi1(tau),phi2(tau))
output delta_star = max(phi1(tau_star),phi2(tau_star))
error^ub = phi(tau_2) - phi(tau_1)
```

5 An illustrating example

In this section the results are illustrated for an application of the approximate model to a small pedagogical example before we in Section 6 apply it to a larger industrial problem. To make the results visually interpretable, the dimension of the decision space in the first example is chosen to be $n = 2$. The example indicates that "similar" objectives may be aggregated with only a small loss of precision of $\mathcal{P}$, whereas the loss will be large if "non-similar" objectives are forced to be aggregated. However, it also indicates that the utilization of the tolerance parameter τ can substantially repair the damages from unsuitable aggregations.

The model has been implemented in MATLAB [16] in combination with AMPL [10] and the CPlex solver [13] for solving of problem (I).

Example 5.1 *The example instance of* (1) *is defined by*

$$
\begin{aligned}
f_1(\mathbf{x}) &:= (x_1 - 6)^2 + 4\,(x_2 - 4)^2,\\
f_2(\mathbf{x}) &:= 4\,(x_1 - 5)^2 + \frac{9}{4}\,(x_2 - 5)^2,\\
f_3(\mathbf{x}) &:= 4\,(x_1 - \frac{11}{2})^2 + 4\,(x_2 + 3)^2,\\
f_4(\mathbf{x}) &:= \frac{25}{4}\,(x_1 - 4)^2 + \frac{9}{4}\,(x_2 + 5)^2,\\
f_5(\mathbf{x}) &:= (x_1 + 3)^2 + (x_2 + 3)^2,
\end{aligned}
$$

and

$$X := \{x_1 \times x_2 \mid x_i \in \{-10, -9.75, -9.5, \ldots, 10\},\ i = 1, 2\}.$$

Figure 11 *shows level curves of the objective functions and the (convex hull of) Pareto optimal subset* $\mathcal{P} \subseteq X$.

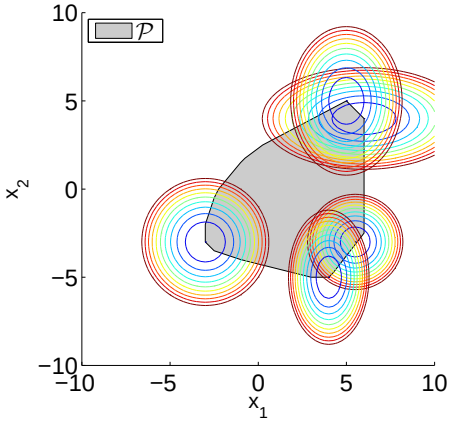


Figure 11: Level curves of the objective functions and the Pareto optimal subset $\mathcal{P} \subseteq X$.

The pairwise correlations, defined in (26)*, of the objective functions* $f_1, \ldots, f_5$ *are shown in Table 1.*

	f_1	f_2	f_3	f_4	f_5
f_1	1	0.74	0.13	0.11	-0.28
f_2		1	0.63	0.69	-0.35
f_3			1	0.97	0.15
f_4				1	0.09
f_5					1

Table 1: Correlation coefficients for the objective functions in the example.

Solving the approximate model for the example problem and with the number of objectives in the reduced problem restricted to $4, 3, 2$ *and* 1 *leads to the results in Figure* 12*. Shown are level curves for the respective aggregated objective functions, and the resulting (convex hulls of) Pareto optimal sets for the respective reduced problems, with and without a tolerance* $\tau > 0$*, as compared to the original* $\mathcal{P}$*. We have used the value* 0.15 *for the centrality parameter* ρ*. Note that the Hausdorff distance is not measured in the 2-dimensional decision space but in the 5-dimensional objective space.*

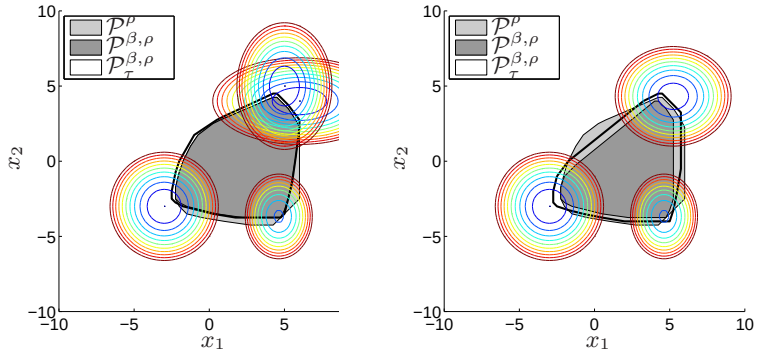
The figures clearly indicate that as long as it is possible to create a reduced problem where each objective function is aggregated by original objectives with large pairwise correlations, the loss of precision of the Pareto optimal set will not be too large. We also see that by allowing a tolerance $\tau > 0$*, the similarities of the original and the reduced Pareto optimal sets are increased significantly.*

6 An industrial application

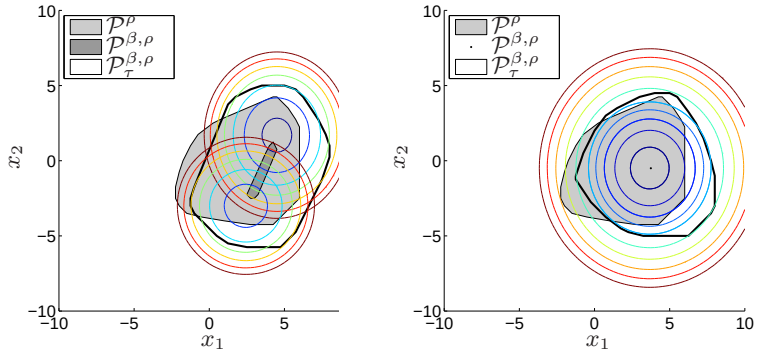
The approximate model has been applied to an industrial problem regarding the configuration of heavy-duty trucks. Typically, trucks are very customer adapted depending on differences in the environment in which the truck is to be used and for what transport mission. For this reason, the trucks are highly modularized, making an enormous number of configurations possible. The background to the problem approached is that there is no wish to produce a truck configuration that is worse than some other truck configuration in all possible quality measures for the truck. However, different customers may appreciate the quality measures differently. Thus, the target for the truck company should be to offer the customers a number of configurations that are in some sense well distributed over the Pareto optimal set of trucks. The first step is to identify this Pareto optimal set.

Here, a simple linear multi-body system model of a truck has been used programmed in MATLAB [16]. The model is illustrated in Figure 13. The complete decision (or, configuration) space X' is for this problem assumed to consist of all possible combinations of a number of cabs, front axle installations, rear axle installations, and superstructures. In this example $|X'| = 1296$.

From the outputs of the model, 12 quality measures (i.e., objective functions in the multi-objective optimization problem) have been defined, concerning, e.g., durability, driver environment, load comfort, and exposition of road wear.



(a) $r = 4,\ d_H = 0.11,\ \tau^* = 0.00068$ (b) $r = 3,\ d_H = 0.19,\ \tau^* = 0.0037$



(c) $r = 2,\ d_H = 0.39,\ \tau^* = 0.15$ (d) $r = 1,\ d_H = 0.41,\ \tau^* = 0.26$

Figure 12: Illustration of results for different numbers of objective functions in the reduced problem.

Solving the original multi-objective optimization problem (1), where $\{f_1, \ldots, f_{12}\}$ represent the 12 quality measures and where X consists of the 1296 feasible configurations showed that the proportion of originally Pareto optimal solutions is $\frac{|\mathcal{P}|}{|X'|} \approx 0.19$. The pairwise correlations between the objectives are shown in Table 2, where it is clear that some of the objectives are positively correlated, some are negatively correlated, and some are rather independent of each other.

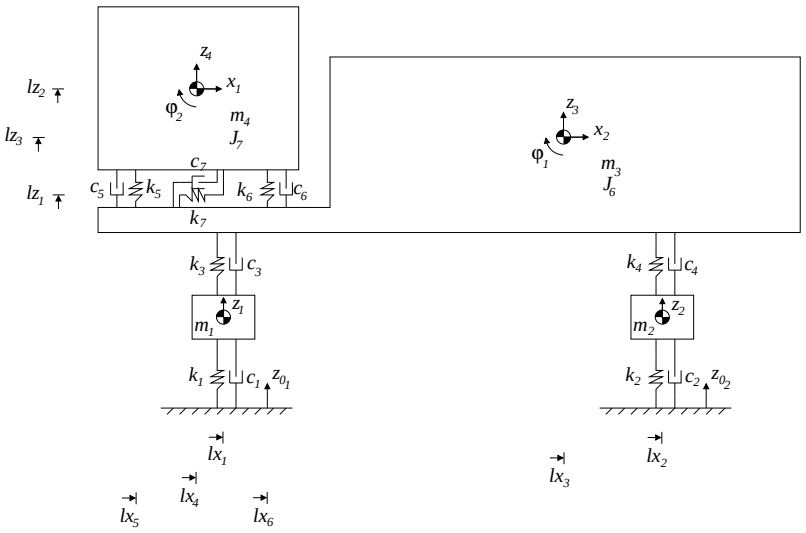


Figure 13: The truck model used. A configuration is defined by the positions lx and lz, the spring constants k, the damping constants c, the masses m and the moments of inertia J. Inputs to the model are road excitations z_0 and outputs are displacements, velocities, accelerations and forces in the degrees of freedom x, z and φ.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}
f_1	1	-0.24	0.29	-0.06	0.19	0.78	-0.15	0.25	-0.31	0.39	-0.08	-0.21
f_2		1	0.63	-0.40	0.19	-0.17	0.22	0.37	0.13	-0.45	-0.22	0.09
f_3			1	0.06	0.77	0.39	0.11	0.61	-0.18	-0.19	-0.29	-0.10
f_4				1	0.68	-0.03	0.13	0.05	0.15	0.14	0.18	0.11
f_5					1	0.28	0.18	0.49	-0.03	-0.04	-0.09	0.01
f_6						1	-0.24	0.22	-0.45	0.39	-0.07	-0.28
f_7							1	0.44	0.80	0.08	0.50	0.81
f_8								1	0.06	-0.39	-0.32	0.19
f_9									1	0.22	0.73	0.81
f_{10}										1	0.76	0.21
f_{11}											1	0.58
f_{12}												1

Table 2: Correlation coefficients for the objective functions in the industrial application.

For the definition of τ-Pareto optimality to make sense it is important that the objective functions are of approximately the same magnitude. As already introduced

in Section 3.3, the objective functions are scaled, where each $f_i(\mathbf{x})$ is replaced by the expression in (14), leading to the range $[0, 1]$ of each scaled objective function over $\mathcal{P}$.

The default parameter setting that has been used is given in Table 3.

r	6
ρ	0.15
α	0
Distance measure	Euclidean
$\varepsilon^{\text{term}}$	10^{-4}

Table 3: Default algorithm settings.

To simulate a real application of the method developed, X' is partitioned randomly into two equally large parts X and X^{ref}. The set X is the set for which we find $\boldsymbol{\beta}$ and τ, which defines the reduced problem. X^{ref} is a reference set, to which we apply the reduced problem defined and where the quality of the resulting ρ-central, τ-Pareto optimal set $\mathcal{P}_\tau^{\boldsymbol{\beta},\rho}$ is measured. Through this procedure we decrease the risk of over-fitting the data (cf. Section 2). In Table 4 the results for different values of r are shown. We present the Hausdorff distances and also what is defined in (36), the mean distance over all points in $\mathcal{P}^\rho$ and $\mathcal{P}_{\tau^*}^{\boldsymbol{\beta}^*,\rho}$ to their respective nearest points in $\mathcal{P}_{\tau^*}^{\boldsymbol{\beta}^*,\rho}$ and $\mathcal{P}^\rho$ respectively (cf. the Hausdorff distance (6)).

$$d_{\text{mean}}(E, F) = \text{mean}\left\{\min_{\mathbf{v}\in F} d(\mathbf{u}, \mathbf{v}); \min_{\mathbf{u}\in E} d(\mathbf{u}, \mathbf{v})\right\} \tag{36}$$

The results are presented both for X and X^{ref}. Also the optimal tolerance value, τ^*, and the optimal objective value, z^*, in problem (I) are presented. All values in the table are averaged over 10 runs.

r	d_H	d_{mean}	d_H^{ref}	$d_{\text{mean}}^{\text{ref}}$	τ^*	z^*
11	0.18	0.002	0.29	0.005	0	0.95
10	0.23	0.004	0.38	0.009	0	0.93
9	0.26	0.011	0.44	0.015	0	0.93
8	0.36	0.021	0.47	0.022	0	0.92
7	0.46	0.034	0.76	0.068	0.008	0.91
6	0.46	0.041	0.75	0.103	0.016	0.89
5	0.56	0.068	1.05	0.170	0.033	0.77
4	0.53	0.085	0.98	0.207	0.078	0.69
3	0.58	0.113	0.93	0.216	0.105	0.55
2	0.56	0.106	0.83	0.213	0.190	0.44
1	0.52	0.099	0.75	0.152	0.236	0.18

Table 4: Numerical results for varying numbers of aggregated objectives.

There is no sharp transition point where the error increases dramatically when r is decreased, thus it is hard to draw any general conclusions on how many objective functions that are needed for a reasonable error size. Also, it is not possible to draw any clear conclusions on how the errors will turn out based on the correlation matrix.

To enable a better interpretation of the numbers in the tables, we present in Table 5, in the first two columns the objective values (non-scaled) for the active pair $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of solutions defining the maximum distance (cf. Figure 14) for a particular run resulting in a Hausdorff distance $d_H = 0.585$.

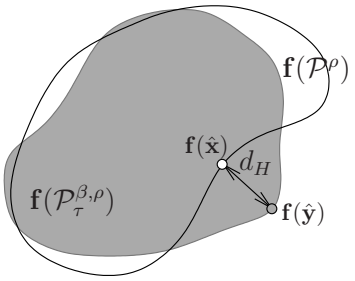


Figure 14: Illustration of the active pair of points $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ defining the Hausdorff distance.

The third column contains the span,

$$\text{Span}_i = \max_{\mathbf{x} \in X} f_i(\mathbf{x}) - \min_{\mathbf{x} \in X} f_i(\mathbf{x}),$$

i.e., the difference between the largest and the smallest value of each function over the configuration space. In the fourth column the difference between the active pair relative to the span is presented:

$$\text{Diff}_i = \frac{|f_i(\hat{\mathbf{x}}) - f_i(\hat{\mathbf{y}})|}{\text{Span}_i}.$$

	$\mathcal{P}^{\rho}$	$\mathcal{P}^{\beta^*,\rho}_{\tau^*}$	Span	Diff
f_1	1.905	1.990	0.69	0.12
f_2	2.251	2.380	0.58	0.22
f_3	1.280	1.338	0.79	0.07
f_4	2.721	2.783	1.15	0.05
f_5	1.447	1.546	1.29	0.08
f_6	1.632	1.662	0.88	0.03
f_7	-15550	-15550	1000	0.00
f_8	27460	27440	3940	0.01
f_9	189.8	191.3	43.8	0.04
f_{10}	738.6	822.9	303	0.28
f_{11}	1.061	1.100	0.39	0.10
f_{12}	2.486	3.267	3.42	0.23

Table 5: Difference between the pair defining the maximum difference ($d_H = 0.585$).

	$\mathcal{P}^{\rho}$	$\mathcal{P}^{\beta^*,\rho}_{\tau^*}$	Span	Diff
f_1	1.893	1.896	0.69	0.01
f_2	2.275	2.257	0.58	0.03
f_3	1.338	1.302	0.79	0.04
f_4	2.741	2.750	1.15	0.01
f_5	1.416	1.392	1.29	0.02
f_6	1.556	1.530	0.88	0.03
f_7	-15550	-15550	1000	0.00
f_8	27620	27647	3940	0.01
f_9	191.2	190.9	43.8	0.01
f_{10}	732.6	739.6	303	0.02
f_{11}	1.057	1.068	0.39	0.03
f_{12}	2.427	2.539	3.42	0.03

Table 6: Difference between the pair closest to the mean distance ($d_{\text{mean}} = 0.093$).

Table 5 shows that the difference between a point in the ρ-central part of the original Pareto optimal set and a point in the ρ-central part of the τ-Pareto optimal set of the reduced problem may be quite large. For the two points in the table defining the largest distance in the objective space ($\Re^{12}$), in this case 0.0585, the relative differences between the points are over 20% for three of the objectives. However, Table 6 indicates that even though the maximum difference may be large, the differences for a pair with a distance close to the mean distance are small. For the pair in Table 6 the relative differences are only a few percent in each objective with a maximum value of 4.4% and a mean value of 1.9%.

Table 7 shows how the value of the centrality parameter ρ affects the results (by varying ρ while keeping the other parameters in Table 3 fix).

ρ	d_H	d_{mean}	d_H^{ref}	$d_{\text{mean}}^{\text{ref}}$	τ^*
0	0.53	0.044	0.69	0.051	0.017
0.05	0.49	0.050	0.89	0.131	0.022
0.10	0.48	0.047	0.81	0.117	0.021
0.15	0.46	0.041	0.75	0.103	0.016
0.20	0.37	0.034	0.65	0.092	0.015
0.25	0.34	0.028	0.77	0.147	0.014
0.30	0.22	0.018	0.78	0.195	0.007

Table 7: Numerical results when varying the centrality parameter.

As might have been expected, the errors tend to decrease with an increased centrality parameter, since the more extreme Pareto optimal solutions, hard for the reduced problem to catch, are filtered out. However, the pros of using a larger value of

ρ must be balanced against the cons, where a large value means that a smaller part of the Pareto optimal set is assumed to be interesting. We also see in the table that the values for the reference set X^{ref} do not decrease as nicely as in X. A plausible explanation is that for larger values on ρ, $|\mathcal{P}^\rho|$ is smaller, which increases the risk of over-fitting the decision variables when finding the reduced problem.

7 Conclusions

The motivation behind this work is the fact that industrial multi-objective optimization problems often are computationally hard to solve and that it might not be very important to find the true Pareto optimal set. We have created two models that reduce the original problem by decreasing the cardinality of the set of objective functions using aggregations and such that the precision lost is minimized. The first model, leading to an exact solution given the stated problem, is based on an explicit characterization of Pareto optimality. However, the complexity of this model is far too high for it to be solved for a practical case. The second model solves the problem approximately by separating the optimization into two sequential problems in a way such that the practical complexity is substantially reduced.

The second model has been applied to an industrial application showing that it is possible to reduce the problem and to get a measure on how large the induced errors are. We have not found any clear a priori indicators, e.g., in terms of the correlation matrix for the objective functions, on how many objectives that are required for a reasonably small error. It is hard to say a priori how much could be earned in problem reduction for a certain error level. The error seems to increase rather smoothly, suggesting that the model should be applied for different values of the number of objectives in the reduced problem before deciding on a suitable reduction. We leave it to the decision maker (or, the problem owner) to decide on how large error that is tolerable for the specific application and how much the problem-specific improvement in computation time by the reduction is worth. The numerical experiments also show that even if the distance between the worst pair (defining the Hausdorff distance) is large, the distance between an average pair might be small.

To summarize, we have provided a method for problem reduction that can be applied to any multi-objective optimization problem. The approximation is focused to the interesting part of the decision space, i.e., the part that is (near-)Pareto optimal. The outcome of the reduction is both a simplified problem formulation to use instead of the original one, and also a measure of the induced error by using it.

References

[1] P. AGRELL, *On redundancy in multi criteria decision making*, European Journal of Operational Research, 98 (3) (1997), pp. 571–586.

[2] M. BAZARAA, H. SHERALI, AND C. SHETTY, *Nonlinear Programming Theory and Algorithms*, Wiley, Hoboken, NJ, USA, 2006.

[3] D. BROCKHOFF, T. FRIEDRICH, N. HEBBINGHAUS, C. KLEIN, F. NEUMANN, AND E. ZITZLER, *Do additional objectives make a problem harder?*, in Genetic and Evolutionary Computation Conference (GECCO 2007), New York, NY, USA, 2007, ACM Press.

[4] D. BROCKHOFF AND E. ZITZLER, *On objective conflicts and objective reduction in multiple criteria optimization*, TIK Report 243, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, Switzerland, Feb. 2006.

[5] C. CARLSSON AND R. FULLÉR, *Multiple criteria decision making: The case for interdependence*, Computers & Operations Research, 22 (3) (1995), pp. 251–260.

[6] W. M. CARLYLE, J. W. FOWLER, E. S. GEL, AND B. KIM, *Quantitative comparison of approximate solution sets for bi-criteria optimization problems*, Decision Sciences, 34 (1) (2003), pp. 63–82.

[7] P. CZYŻAK AND A. JASZKIEWICZ, *Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis, 7 (1998), pp. 34–47.

[8] K. DEB AND D. SAXENA, *On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems*, KanGAL Report 2005011, Department of Mechanical Engineering, IIT Kanpur, India, 2005.

[9] M. EHRGOTT AND S. NICKEL, *On the number of criteria needed to decide Pareto optimality*, Mathematical Methods of Operations Research, 55 (2002), pp. 329–345.

[10] R. FOURER, D. GAY, AND B. KERNINGHAN, *AMPL – A Modeling Language for Mathematical Programming*, Brooks/Cole, Toronto, Canada, 2003.

[11] T. GAL AND T. HANNE, *Consequences of dropping nonessential objectives for the application of MCDM-methods*, European Journal of Operational Research, 119 (2) (1999), pp. 373–378.

[12] T. GAL AND H. LEBERLING, *Redundant objective functions in linear vector maximization problems and their determination*, European Journal of Operational Research, 1 (1977), pp. 176–184.

[13] ILOG Inc. *ILOG CPLEX 10.110*, `www.ilog.com/products/cplex`.

[14] J. KOSKI AND R. SILVENNOINEN, *Norm methods and partial weighting in multicriterion optimization of structures*, International Journal for Numerical Methods in Engineering, 24 (1987), pp. 1101–1121.

[15] T. LOWE, J.-F. THISSE, J. WARD, AND R. WENDELL, *On efficient solutions to multiple objective mathematical programs*, Management Science, 30 (1984), pp. 1346–1349.

[16] The Mathworks. *MATLAB 7.3.0 (R2006b)*, `www.mathworks.se/products`.

[17] K. MIETTINEN, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, MA, USA, 1998.

[18] T. MITCHELL, *Machine Learning*, McGraw-Hill, Singapore, 1997.

[19] G. NEMHAUSER AND L. WOLSEY, *Integer and Combinatorial Optimization*, Wiley, New York, NY, USA, 1988.

[20] A. VIANA AND J. PINHO DE SOUSA, *Using metaheuristics in multiobjective resource constrained project scheduling*, European Journal of Operational Research, 120 (2000), pp. 359–374.

Multi-Objective Design of a Combinatorial Structure

Peter Lindroth Michael Patriksson Ann-Brith Strömberg

Abstract

Engineering design problems are often formulated as multi-objective optimization problems. We consider the problem of designing an optimal population of configurations, where the configurations are composed by common elements. Searching for a population of solutions that are good with respect to different combinations of the multiple objectives can be seen as a search for a clustering of the Pareto optimal set to the multi-objective optimization problem. Further, a natural wish is to use common parts to construct the population of design solutions. This paper proposes a (single-objective) optimization problem through which the clustering is performed in a way such that the resulting solutions approximate the Pareto optimal solution well, while at the same time the variables in the decision space are, by construction, required to be common. The procedure is applied to instances constructed from test functions from the literature with interesting results. The usefulness of applying the procedure to practical problems and what types of sensitivity analyses that can be performed are discussed and demonstrated. Suggestions are also made on how to adapt the developed methodology to simulation-based multi-objective optimization problems.

1 Introduction

A frequent wish in engineering design of mass-market products is to create a large variety of product configurations using just a few variants of each part. One example is trucks, another is kitchen cupboards.

In this paper we tackle the problem of deciding which variants to create by using a mathematical modeling approach with a strategy based on an underlying multi-objective optimization problem. We assume that the quality of a configuration is measured by a number of objective functions, each to be optimized, and that each configuration comprises a number of parts, each to be selected from a specific set of possible designs. The goal is to maximize the quality of the total product variety given the sets of possible parts. To our knowledge this approach is not apparent in the literature.

1.1 Motivation

A common property of engineering design problems is that they invoke a number of more or less conflicting criteria (or, objectives). Examples are *weight* $\leftrightarrow$ *durability* and *cost* $\leftrightarrow$ *feature level*. The objectives may be appreciated differently by different customers, for example depending on how and in which environment the product should be used and on the financial strength of the customer. This makes a multi-objective approach for solving such problems natural. Further, for cost and flexibility reasons, it is advantageous to design a small number of variants of each of the parts to be combined, forming a large number of possible configurations. Moreover, it is not enough to require that each configuration is good in itself; due to synergies of scale, the set of all produced configurations must be evaluated as a collective. We study the problem of how to systematically design the different variants such that the resulting collection of variants yields an, in a certain sense, optimal set of configurations.

We denote the technique to be introduced by *Implicit clustering*. Through traditional clustering or *Explicit clustering*, one partitions a set into groups, where objects belonging to the same group are similar, whereas objects belonging to different groups are dissimilar. An extensive overview of clustering techniques is found in [8]. Explicit clustering cannot be applied directly to our problem, since there is both a decision space and an objective space, both in which it is important where the resulting configurations are located. Clustering in the decision space only leads to no control of the distribution in the objective space, and clustering in the objective space only leads to a set of configurations without structure in the decision space. The technique presented in this paper resolves these problems by considering both spaces simultaneously, by the construction of a certain optimization problem.

1.2 Outline

In Section 2 we give a mathematical formulation of the design problem, discuss how to measure the quality of a set of configurations, and investigate the mathematical properties of the problem. We present a solution procedure in Section 3. In Section 4 we discuss the type of sensitivity analyses that can be performed for a practical problem and in Section 5 we solve some instances of the design problem using test functions from the literature. Then, in Section 6 we propose a procedure to be added to the solution process, which is reasonable if the objective functions are computationally intense. Finally, in Section 7 we conclude the paper and give some propositions for future work. These intend to make the solution strategy applicable to a larger class of design problems than that originally considered.

2 A mathematical formulation of the problem

We begin this section by defining multi-objective optimization. We then formulate our design problem, which, since it utilizes an underlying multi-objective optimiza-

tion problem in its objective function, is called the *Multi-Objective Combinatorial Design Problem* (MOCDP). The objective function to use is discussed and some mathematical properties of MOCDP are analyzed.

2.1 Multi-objective optimization

A multi-objective (non-linear) optimization problem (MONP) can mathematically be formulated by the standard notation

$$\min_{\mathbf{x}\in X} \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\}, \tag{1}$$

where $\mathbf{x} \in \Re^n$ is a vector of decision variables, $X \subseteq \Re^n$ denotes the decision space, and each function $f_i : X \to \Re,\ i = 1 \ldots, k$, is an objective function to be minimized. We adopt the convention of letting the minimization operator apply to vectors. If the objective functions are at least partially in conflict, i.e., there exists no $\mathbf{x} \in X$ that simultaneously minimizes all k objectives, then an optimal solution to (1) is not well-defined since there exists no natural complete ordering between vectors. However, there exists a set of decision vectors, in which the best solution by rational judgements, provided the mathematical formulation, must be contained regardless of the relative importance of each single objective. This is the *Pareto optimal set* (or, equivalently, the *efficient* or *non-dominated* set).

Definition 2.1 *Given a set X of feasible vectors and a set $\{f_1, \ldots, f_k\}$ of objective functions to minimize, a vector $\mathbf{x}^* \in X$ is defined as* Pareto optimal *if there exists no other vector $\mathbf{x} \in X$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*),\ i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one $j \in \{1, \ldots, k\}$. An objective vector $\mathbf{z}^* = \mathbf{f}(\mathbf{x}^*)$ is called* Pareto optimal *if the corresponding vector $\mathbf{x}^*$ is Pareto optimal. The set of all Pareto optimal vectors is denoted $\mathcal{P} \subseteq X$.*

Definition 2.2 *Given a set X of feasible vectors and a set $\{f_1, \ldots, f_k\} \in \Omega = \{\mathbf{f} \mid \mathbf{f} : \Re^n \to \Re^k\}$ of objective functions to minimize, the* Pareto operator $P : \Re^n \times \Omega \to \Re^n$ *is defined by $P(X, \mathbf{f}) = \mathcal{P}$.*

2.2 The multi-objective combinatorial design problem

Initially, we assume that the decision variables are continuous and that $X \subseteq \Re^n$ is defined by box constraints: $X = \prod_{j=1}^{n} X^j$, where $X^j = [l_j, u_j],\ j = 1, \ldots, n$, with $-\infty < l_j < u_j < \infty$. The assumption that each configuration consists of a fixed number of parts that are combined then translates to that each variant x_j of part j is to be selected from the interval $X^j = [l_j, u_j]$, and that each configuration has the representation $\mathbf{x} = (x_1, \ldots, x_n),\ x_j \in X^j,\ j = 1, \ldots, n$.

Assume that part j may have m_j different variants, and let the variants selected be represented by the variables $x_{j\ell},\ \ell = 1, \ldots, m_j$. The available configurations are then defined by the product set $X_D = \prod_{j=1}^{n} \{x_{j1}, \ldots, x_{jm_j}\}$. Such a configuration set is illustrated in Figure 1 where $n = 2, m_1 = 3$, and $m_2 = 2$.

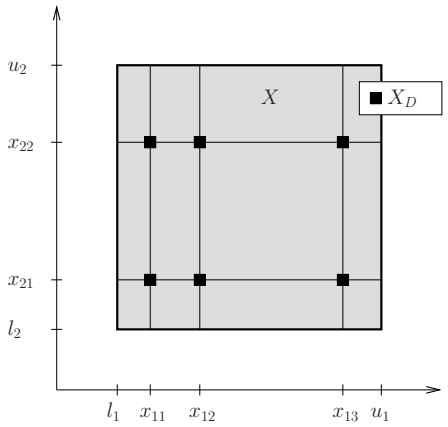


Figure 1: Illustration of a set of configurations $X_D \subset X$.

We wish to select the values of the variables $x_{j\ell},\ \ell = 1, \ldots, m_j,\ j = 1, \ldots, n$, such that the product set of configurations is, in a certain sense, optimal. We collect the decision variables in the vector

$$\mathbf{y} = (x_{11}, \ldots, x_{1m_1}, x_{21}, \ldots, x_{2m_2}, \ldots\ldots, x_{n1}, \ldots, x_{nm_n}) \in Y \subseteq \Re^{\sum_{j=1}^{n} m_j}, \quad (2)$$

where

$$Y = \underbrace{X^1 \times \cdots \times X^1}_{m_1 \text{ factors}} \times \underbrace{X^2 \times \cdots \times X^2}_{m_2 \text{ factors}} \times \cdots\cdots \times \underbrace{X^n \times \cdots \times X^n}_{m_n \text{ factors}}, \quad (3)$$

and denote the resulting set of available configurations as $X_D(\mathbf{y})$. The motivation behind the problem formulation is that with just $m := \sum_{j=1}^{n} m_j$ decision variables we decide on (the much larger number) $\prod_{j=1}^{n} m_j$ configurations. Further, let $\mathcal{Q}_{\mathcal{R}} : \Re^{\sum_{j=1}^{n} m_j} \to \Re$ be a function measuring the negative collective quality of a set of configurations (negative quality is utilized in order to obtain a minimization problem). The subscript $\mathcal{R}$ on the quality function represents a possible reference set for the configurations to be compared to.

We next introduce the (single-)objective optimization problem, the *Multi-Objective Combinatorial Design Problem* (MOCDP):

$$\mathcal{Q}^*_{\mathcal{R}}(m_1, \ldots, m_n) = \underset{\mathbf{y}}{\text{minimize}} \quad \mathcal{Q}_{\mathcal{R}}(\mathbf{y}) \quad (4a)$$

$$\text{subject to} \quad l_j \le x_{j\ell} \le x_{j,\ell+1} \le u_j, \qquad \ell = 1, \ldots, m_j - 1, \quad j = 1, \ldots, n. \quad (4b)$$

The constraints (4b) ensure that the value of each decision variable is chosen from its feasible interval, i.e., $\mathbf{y} \in Y$ and thus $X_D \in X$; they also exclude solutions that are

equivalent due to symmetry. For $m_j = 1$, the constraints (4b) should be replaced by $l_j \leq x_{j1} \leq u_j,\ \ j = 1, \ldots, n$.

The above formulation uses exactly m_j variants of part j (however not necessarily distinct). One could think that "at most" m_j variants would be more appropriate. These two formulations are, however, equivalent in the sense that their optimal $\mathcal{Q}_{\mathcal{R}}$-values are the same. The latter formulation is a relaxation of the former, and the optimal objective function $\mathcal{Q}^*_{\mathcal{R}}(m_1, \ldots, m_n)$ is monotonously decreasing with each $m_j,\ j = 1, \ldots, n$, for all reasonable quality functions $\mathcal{Q}_{\mathcal{R}}(\cdot)$.

Remark 2.3 *The formulation (4) can, with a suitable definition of $\mathcal{Q}_{\mathcal{R}}(\cdot)$, also be used for single-objective optimization design, with the aim of finding a combinatorial set of solutions which, as a collective, is robust with respect to variations or uncertainties in the underlying optimization problem.*

We next present an instance of MOCDP (without specifying the quality measure $\mathcal{Q}_{\mathcal{R}}(\cdot)$), which will be used for illustrative purposes in the paper.

Example 2.4 *Let the underlying MONP be defined by the decision space $X = [0, 1]^2$ and the objective functions $\mathbf{f}(\mathbf{x}) := \{f_1(\mathbf{x}), f_2(\mathbf{x})\}$, where*

$$f_1(\mathbf{x}) = \left(x_1 + \frac{1}{4}\right)^2 + \left(x_2 + \frac{1}{4}\right)^2, \tag{5a}$$

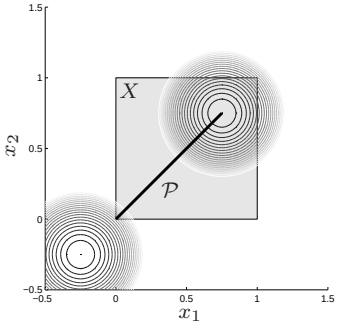
and

$$f_2(\mathbf{x}) = \left(x_1 - \frac{3}{4}\right)^2 + \left(x_2 - \frac{3}{4}\right)^2. \tag{5b}$$

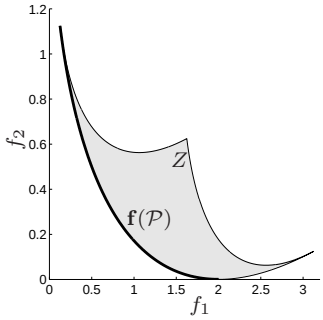
The Pareto optimal set $\mathcal{P} = \left\{\mathbf{x} \in \Re^2 \mid 0 \leq x_1 = x_2 \leq \frac{3}{4}\right\}$. Let now $m_1 = m_2 = 2$, which leads to the vector $\mathbf{y} = (x_{11}, x_{12}, x_{21}, x_{22})$ of decision variables and the configuration set $X_D(\mathbf{y}) = \{(x_{11}, x_{21}), (x_{11}, x_{22}), (x_{12}, x_{21}), (x_{12}, x_{22})\}$. We get the MOCDP

$$\begin{aligned} &\text{minimize} && \mathcal{Q}_{\mathcal{R}}(\mathbf{y}), && && (6a)\\ &\text{subject to} && 0 \leq x_{j1} \leq x_{j2} \leq 1, && j = 1, 2. && (6b) \end{aligned}$$

Figure 2 illustrates (a) the decision space and (b) the objective space of the underlying MONP.



(a) The decision space, X, level curves for the two objective functions, and the Pareto optimal set $\mathcal{P} \subseteq X$.



(b) The objective space, $Z = \mathbf{f}(X)$, and the image, $\mathbf{f}(\mathcal{P})$, of the Pareto optimal set.

Figure 2: Illustration of the design and objective spaces of the MONP in Example 2.4. The Pareto optimal set $\mathcal{P} \subseteq X$, and the image of the Pareto optimal set $\mathbf{f}(\mathcal{P}) \subseteq Z = \mathbf{f}(X)$, are marked in black in the respective figures.

2.3 Measuring the quality of the set of configurations

It is not obvious how to define the quality function $\mathcal{Q}_{\mathcal{R}}(\cdot)$, but it is clear that $\mathbf{f}(\mathcal{P})$ in some suitable sense should be approximated by $\mathbf{f}(X_D)$. As noted e.g. in [16, 18, 3, 21] there is no standard technique in the literature for measuring the quality of approximate Pareto sets, and for many of the intuitive measures one can easily construct examples that shows good results for obviously bad approximations and vice versa.

Two measures that have been designed for evaluation of metaheuristics for multi-objective optimization problems, and which make sense also in our application, are $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$, proposed in [4] and also used in e.g. [19]. We give the definitions of $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$ below. We will replace $\mathcal{Q}_{\mathcal{R}}(\cdot)$ by $Dist1_{\mathcal{R}}$, $Dist2_{\mathcal{R}}$, or with a combination of these. These two metrics reward approximate sets (in our case X_D) that comprise points that are near-Pareto optimal while being evenly distributed over the Pareto set. For the evaluation of the approximate set X_D, both $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$ require a reference set $\mathcal{R} \subset X$, which should be a discrete approximation of the true Pareto optimal set $\mathcal{P}$. If $\mathcal{P}$ is known, $\mathcal{R}$ can be an evenly spread discrete subset of $\mathcal{P}$, which is the ideal situation. If $\mathcal{P}$ is not known, $\mathcal{R}$ may consist of (a subset of) the non-dominated points found using any solution method for multi-objective optimization.

A high quality of a set X_D means that to each vector $\mathbf{x}^r \in \mathcal{R}$ there is a vector $\mathbf{x}^d \in X_D$ *close* to $\mathbf{x}^r$. The closeness, $c_{\mathbf{w}}(\mathbf{x}^r, \mathbf{x}^d)$, of the vectors $\mathbf{x}^r \in \mathcal{R}$ and $\mathbf{x}^d \in X_D$ is a non-symmetric measure defined as

$$c_{\mathbf{w}}(\mathbf{x}^r, \mathbf{x}^d) = \max_{i \in \{1,\ldots,k\}} \left\{ \max \left\{ 0, w_i\big(f_i(\mathbf{x}^d) - f_i(\mathbf{x}^r)\big) \right\} \right\}, \quad \mathbf{x}^r \in \mathcal{R},\ \mathbf{x}^d \in X_D, \qquad (7)$$

where $w_i \geq 0$ is the weight assigned to objective i, $i = 1, \ldots, k$, and $\mathbf{w} = \{w_1, \ldots, w_k\}$. In contrast to the original definition in [4] we will from now on replace X_D in the definition (7) of closeness by $\mathcal{P}_D = P(X_D, \mathbf{f})$, since a dominated solution is never preferred to a non-dominated solution by a rational decision maker. Thus, the closeness of a point $\mathbf{x}^r \in \mathcal{R}$ to a point $\mathbf{x}^d \in X_D$, where $\mathbf{x}^r$ is (weakly) dominated by $\mathbf{x}^d$, (i.e., $\mathbf{f}(\mathbf{x}^d) \leq \mathbf{f}(\mathbf{x}^r)$), is defined to be zero. Otherwise, the closeness is given by the maximum weighted deterioration of an objective value over the set of objective functions. The weights in the expression (7) are set to

$$w_i = \frac{1}{\max\limits_{\mathbf{x} \in \mathcal{R}} f_i(\mathbf{x}) - \min\limits_{\mathbf{x} \in \mathcal{R}} f_i(\mathbf{x})}, \quad i = 1, \ldots, k, \tag{8}$$

i.e., inversely proportional to the range of f_i over $\mathcal{R}$ [1]. An illustration of the closeness between two points $\mathbf{x}^r$ and $\mathbf{x}^d$ is given in Figure 3.

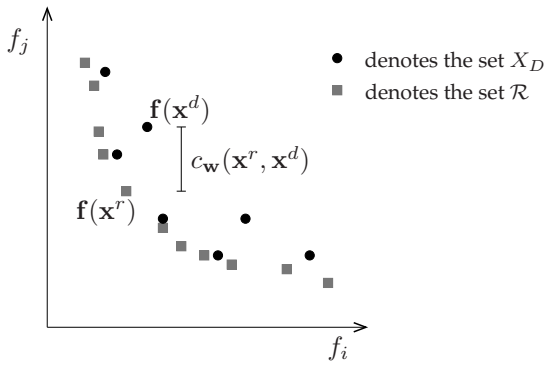


Figure 3: An illustration of the closeness between two points $\mathbf{x}^r$ and $\mathbf{x}^d$ according to the definition (7). Here, $w_i = w_j = 1$.

The $Dist1_{\mathcal{R}}$ measure yields information on the average distance from a point $\mathbf{x}^r \in \mathcal{R}$ to its closest point in X_D, and is defined as

$$Dist1_{\mathcal{R}}(\mathbf{y}) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{x}^r \in R} \left(\min_{\mathbf{x}^d \in X_D(\mathbf{y})} c_{\mathbf{w}}(\mathbf{x}^r, \mathbf{x}^d) \right). \tag{9}$$

Correspondingly, $Dist2_{\mathcal{R}}$ yields information on the maximum distance and is defined as

$$Dist2_{\mathcal{R}}(\mathbf{y}) = \max_{\mathbf{x}^r \in \mathcal{R}} \left\{ \min_{\mathbf{x}^d \in X_D(\mathbf{y})} c_{\mathbf{w}}(\mathbf{x}^r, \mathbf{x}^d) \right\}. \tag{10}$$

Note that if the points in $\mathcal{R}$ are more dense in some region of X, $Dist1_{\mathcal{R}}$ will lead to a biased result, since the denser part of the approximation will possess a larger weight in the sum.

[1] Assumed is that the range is non-zero which is a reasonable assumption for practical problems. If this assumption is not valid, then a positive constant could be added to the denominator in (7) or an estimation could be made of the "scale" of each objective over the interesting region.

Remark 2.5 *The $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$ measures are adopted from the evaluation of metaheuristics for applications for which $\mathcal{P}$ is not known. If there exists an explicit expression for $\mathcal{P}$ one can choose $\mathcal{R} = \mathcal{P}$ and replace the sum in $Dist1_{\mathcal{R}}$ with an integral. This is also possible if there exists a function describing $\mathcal{R}$, e.g. by interpolating the non-dominated points found by some (approximate) solution method.*

2.4 Some mathematical properties of MOCDP

We are interested in MOCDP applied to practical problems. The purpose of this section is to analyze enough mathematical properties of MOCDP such that a suitable solution method can be proposed for such problems.

Proposition 2.6 *MOCDP with the quality function $\mathcal{Q}_{\mathcal{R}}(\cdot)$ being either $Dist1_{\mathcal{R}}$ or $Dist2_{\mathcal{R}}$ is continuous if the underlying MONP is continuous.*

Proof. All f_i's are continuous since MONP is continuous and the $\max$- and $\min$-operators in $Dist1_{\mathcal{R}}$, $Dist2_{\mathcal{R}}$ are continuous. A composition of continuous function is continuous. The feasible set of MOCDP is continuous and, hence, so is the problem. ∎

Since the closeness function defined in (7) is non-differentiable for $\mathbf{y} \in Y$ such that $c_{\mathbf{w}}(\mathbf{x}^r, \mathbf{x}^d) = c_{\mathbf{w}}(\mathbf{x}^s, \mathbf{x}^d),\ r, s \in \mathcal{R}$ for some $\mathbf{x}^d \in X_D(\mathbf{y})$ (i.e. when $\mathbf{x}^d$ changes its nearest point in $\mathcal{R}$), we have the following result:

Proposition 2.7 *If $\mathcal{R}$ a discrete set of points then MOCDP is non-differentiable.*

We continue with investigating convexity properties of MOCDP. If MOCDP is convex, then a local optimum is a global optimum and the problem can be solved to global optimality using a local optimization algorithm. Unfortunately, as shown below, this is not the case, even under very strong assumptions on the underlying MONP.

Example 2.8 *Recall Example 2.4. The underlying MONP is convex since both objective functions are convex and the feasible decision space X is a convex set. Let the reference set be $\mathcal{R} = \{(\frac{1}{10}, \frac{1}{10}), (\frac{6}{10}, \frac{6}{10})\}$. Then $\mathcal{R} \subset \mathcal{P} = \left\{(x_1, x_2) \in \Re^2 |\ 0 \leq x_1 = x_2 \leq \frac{3}{4}\right\}$. A globally optimal solution to MOCDP is then $\mathbf{y}^* = (x_{11}^*, x_{12}^*, x_{21}^*, x_{22}^*) = (\frac{1}{10}, \frac{6}{10}, \frac{1}{10}, \frac{6}{10})$ with $\mathcal{Q}_{\mathcal{R}}(\mathbf{y}^*) = 0$ since the number of decision variables is enough to meet all elements in $\mathcal{R}$* [2]*. However, experimenting with a local optimizer shows that it is possible to end up in a local minimum with a positive quality measure. The instance of MOCDP is clearly non-convex which is exemplified below. Let*

$$\begin{aligned} \mathbf{y}^1 &= (0, 0.50, 0.35, 0.70)^{\mathrm{T}}, \\ \mathbf{y}^2 &= (0, 0.70, 0.35, 0.70)^{\mathrm{T}}, \\ \lambda &= 0.5. \end{aligned}$$

[2] This is always true when each $m_j,\ j = 1, \ldots, n$, is larger than the number of distinct values in $\mathcal{R}$ in the corresponding dimension.

A necessary condition for a function to be convex is that a linear interpolation of two function values never is lower than the function itself at the corresponding interpolation between the decision variables. We have a counterexample for convexity of MOCDP for both $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$, since

$$0.149 \approx \mathcal{Q}_{\mathcal{R}}(\lambda \mathbf{y}^1 + (1-\lambda)\mathbf{y}^2) \not\leq \lambda \mathcal{Q}_{\mathcal{R}}(\mathbf{y}^1) + (1-\lambda)\mathcal{Q}_{\mathcal{R}}(\mathbf{y}^2) \approx 0.117$$

for $\mathcal{Q}_{\mathcal{R}}(\cdot) = Dist1_{\mathcal{R}}$ and

$$0.150 \approx \mathcal{Q}_{\mathcal{R}}(\lambda \mathbf{y}^1 + (1-\lambda)\mathbf{y}^2) \not\leq \lambda \mathcal{Q}_{\mathcal{R}}(\mathbf{y}^1) + (1-\lambda)\mathcal{Q}_{\mathcal{R}}(\mathbf{y}^2) \approx 0.148 \qquad (11)$$

for $\mathcal{Q}_{\mathcal{R}}(\cdot) = Dist2_{\mathcal{R}}$.

The disappointing non-convexity result is obviously true also when the quality function $\mathcal{Q}_{\mathcal{R}}$ is a convex combination of $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$. As far as we know there exists no reasonable quality function that preserves the convexity property.

Assuming that the underlying MONP is continuous, we conclude that MOCDP is in general continuous, non-differentiable, and non-convex; we conclude that some suitable global optimizer is needed to solve it.

Our interest in the following result is motivated by the fact that we intend to use penalty-based methods in the solution procedure to handle constraints. The result requires some weak assumptions that are quite vaguely formulated. The important point is that it is likely to hold for practical problems, which are of our interest. Figure 4 helps to understand the result.

Proposition 2.9 *Let $X = \prod_{j=1}^{n}[l_j, u_j]$ be a box-constrained decision space to the underlying MONP to a MOCDP. Assume that MOCDP has a reference set $\mathcal{R}$ which is sufficiently large compared to the cardinality of the configuration set X_D. Assume further that $\mathcal{R}$ is sufficiently spread in X, that $\mathbf{f}(\mathcal{R})$ is sufficiently spread in $\mathbf{f}(X)$, and that the objectives $f_1, \ldots, f_k$, are sufficiently well-behaved. Then to MOCDP, there exists optimal solutions $\mathbf{y}^* \in \text{int}(Y)$ or, equivalently, $X_D(\mathbf{y}^*) \subset \text{int}(X)$.*

Proof. If $\mathcal{R}$ is sufficiently large and spread, then at an optimal solution $\mathbf{y}^* \in Y$, many $\mathbf{x}^r \in \mathcal{R}$ will share the same $\mathbf{x}^d \in X_D(\mathbf{y}^*)$ as their nearest point in X_D. In particular, each $\mathbf{x}^r \in \partial X$ has other vectors $\hat{\mathbf{x}}^r \in \text{int}(X)$ with the same nearest point $\mathbf{x}^d \in X_D$. Then, if $f_1, \ldots, f_k$, are sufficiently well-behaved there will be an optimal solution $\mathbf{y}^*$ where each $\mathbf{x}^d \notin \partial X$, or, equivalently, $\mathbf{y} \in \text{int}(Y)$. This because a small movement of a $\mathbf{x}^d \in \partial X$ out from the boundary will not decrease the maximum closeness between each $\mathbf{x}^r \in \mathcal{R}$ to its nearest $\mathbf{x}^d \in X_D$. ∎

Remark 2.10 *A similar result as in Proposition* (2.9) *with an analogous proof can be formulated for the symmetry-breaking constraints $x_{j\ell} \leq x_{j,\ell+1}$ in* (4b)*. In an optimal solution to MOCDP, there are (likely to exist) optimal solutions where the constraints are not active.*

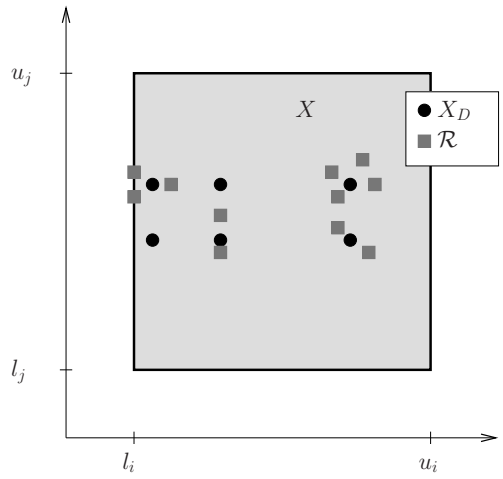


Figure 4: A rough illustration of the fact that the optimal configuration likely will lie in the interior of the decision space X to the underlying MONP for most practical problems. For sufficiently well-behaved objective functions, the closeness between the upper-left-most $\mathbf{x}^d \in X_D$ and its neighbor $\mathbf{x}^r \in \text{int}(X)$ to the right, will not be minimal if $\mathbf{x}^d$ is moved to the left onto the boundary.

3 A solution procedure

First, observe that for a real application the numbers $m_j,\ j = 1, \ldots, n$, of variants may often be decision variables. We suggest to treat them as input parameters, and to solve the problem for different values of m_j to study the sensitivities, i.e., how the optimal solution to MOCDP varies with changes in m_j (cf. Section 4).

We propose a two-step method for solving MOCDP. If there is no problem-specific distance measure known for the evaluation of an approximate Pareto optimal set, we suggest using either of the functions $Dist1_{\mathcal{R}}$, $Dist2_{\mathcal{R}}$ or a combination of these. In the first step of the procedure, a representation of the Pareto optimal set of the underlying MONP should be found. The method for this is arbitrary, and should be chosen with respect to the actual MONP. If this is a non-convex problem with unknown problem characteristics, some evolutionary method [5] might be a reasonable choice.

In the second step of the procedure some global optimization method should be used to find the optimal decision variables $\mathbf{y}^* \in Y$ given the reference set found in step 1. Figure 5 illustrates the two steps of the solution process.

To handle the box and symmetry-breaking constraints (4b) we have used a modified barrier method where linear/logarithmic penalties are added to the objective instead of using constraints. An illustration of modified linear/logarithmic penalty functions is given in Figure 6. By "modified" we mean here that the logarithmic

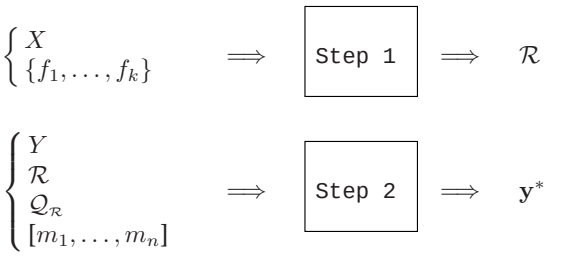


Figure 5: The solution process is divided into two steps. In step 1 , the underlying MONP is solved using some multi-objective optimization solver and $\mathcal{R}$ is defined. In step 2 a global optimizer is used to minimize $\mathcal{Q}_{\mathcal{R}}$ over Y given the reference set $\mathcal{R}$ and the number of allowed variants m_j in each dimension $j,\ j = 1, \ldots, n$.

penalties which lead to the objective function being undefined in parts of the domain, are replaced by linear functions near and outside the boundaries. For example, the optimization problem

$$\begin{aligned} &\text{minimize} && f(x), \\ &\text{subject to} && x \leq u, \\ & && x \in \Re, \end{aligned} \tag{12}$$

is replaced by

$$\begin{aligned} &\text{minimize} && f(x) - \nu \left(1_{(-\infty, u-\epsilon\}}(x) \log(u-x) + 1_{(u-\epsilon, +\infty)}(x) \left(\frac{x-u}{\epsilon} + 1 - \log \epsilon \right) \right), \\ &\text{subject to} && x \in \Re, \end{aligned} \tag{13}$$

where $1_S(x)$ is an indicator function, i.e., equal to one if $x \in S$ and zero otherwise, and ϵ is the distance from the boundary where the logarithmic function is replaced by a linear function. ν is a penalty parameter. For a sufficiently well-behaving function f a globally optimal solution to (13) converges towards a globally optimal solution to (12).

Due to the result in Proposition 2.9 we have good reasons to believe that an optimal solution $\mathbf{y}^*$ lies in the interior of Y and where the symmetry-breaking constraints are non-active. Here, the added penalty does not affect the objective function that much even for a penalty parameter with a positive value of significant size.

Since neither step 1 nor step 2 in general will reach a point, where it is not possible to improve anymore, the maximum allowed computational time of both steps must be set. In step 1, the longer time the algorithm is permitted to work, the more accurate representation of $\mathcal{P}$ is generally obtained. In step 2, the longer the global algorithm is applied, the higher the probability of finding a good solution. However, for sensitivity studies (cf. Section 4) step 1 only has to be performed once.

The fact that the solution algorithm is partitioned into two steps can be taken advantage of for problems with expensive function evaluations, e.g., given by com-

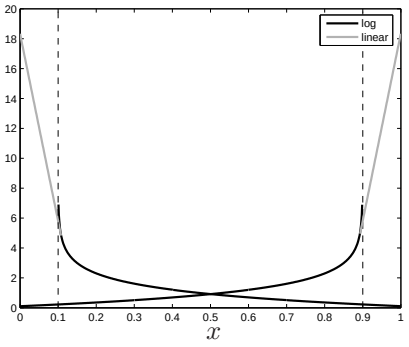


Figure 6: Example of a penalty function using a modified barrier method added to a problem with box constraints, $x \in [0.1, 0.9]$. The logarithmic function transcends smoothly into a linear function at a distance ϵ from the borders. Here $\epsilon = 0.008$.

putationally intensive simulations. The evaluations of the objective function made in step 1 can then be used for the computation of explicit response surfaces to be used in step 2, cf. Section 6 for more details.

So far we have not specified which quality function to use. The two possibilities $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$ seem reasonable for general MOCDP's. However, it is possible that the MOCDP concerns some special application for which there is some other better measure, i.e., leads to solutions that are more attractive from a practical standpoint.

The main disadvantage of $Dist1_{\mathcal{R}}$ is its sensitivity for the distribution of the points in $\mathcal{R}$. It is a well-known fact that evolutionary algorithms often output solution sets $\hat{\mathcal{P}}$ (approximate Pareto optimal sets) whose "density" varies heavily and possesses an a priori unknown distribution. Hence, $Dist1_{\mathcal{R}}$ might not be a good choice for the quality measure. The main disadvantage of $Dist2_{\mathcal{R}}$ is that it is a worst-case measure, only considering the point in $\mathcal{R}$ with the largest distance (closeness) to a point in $\mathcal{P}_D$. As a special case, it assigns the same quality to a set $\mathcal{P}_D$ from which every point in $\mathcal{R}$ lies at distance d as to a set where $\mathcal{P}_D$ that coincides with $\mathcal{R}$ but for a single point in $\mathcal{R}$ at distance d from $\mathcal{P}_D$.

The two proposed quality functions $Dist1_{\mathcal{R}}$ and $Dist2_{\mathcal{R}}$ are obviously correlated; their characteristics are, however, different. By using a convex combination of the quality functions, their disadvantages can be diminished.

4 Sensitivity analysis

When modeling and solving a practical problem as a MOCDP, an important and interesting analysis is to study how the number m_j of variants, $j = 1, \ldots, n$, in the n decision dimensions of the MONP affects the optimal solution. For example, if there

is a limitation on the total number of variants allowed, then it is critical to investigate how sensitive the resulting objective value is to the distribution of the variants in the respective dimensions. For the example in Figure 1, could it be favourable to use three variants in the x_2 dimension and just two in the x_1 dimension? Or perhaps it might be best to use only one variant in the x_1 dimension and four in the x_2 dimension. Further, if the costs of adding a new variant, or the savings of removing one, in a certain dimension, is known, then this information could be used when designing a good set of configurations.

Assume that the set of variants at some point in time is given by the vector $\hat{\mathbf{y}}$ and that the cost of adding a variant in dimension j is δ_j. Observe that the optimal objective value to MOCDP, $\mathcal{Q}^*_{\mathcal{R}}$, is a function of the underlying MONP (defined by $\mathbf{f}$ and X) together with the number of allowed variants $m_j,\ j = 1, \ldots, n$:

$$\mathcal{Q}^*_{\mathcal{R}} = \mathcal{Q}^*_{\mathcal{R}}(\mathbf{f}, X, m_1, \ldots, m_j, \ldots, m_n).$$

The decision to make is whether the quality increase is worth the extra cost, i.e., if the profits gained by reducing the quality measure with

$$\mathcal{Q}^*_{\mathcal{R}}(\mathbf{f}, X, m_1, \ldots, m_j + 1, \ldots, m_n) - \mathcal{Q}^*_{\mathcal{R}}(\mathbf{f}, X, m_1, \ldots, m_j, \ldots, m_n)$$

is larger than the cost δ_j. An analogous study can be made for a possible removal of variants by comparing the savings for removing the variants with the difference of the quality measure when decreasing m_j.

An assumption made above, which may not be valid in many real applications, is that the cost for modifying the current set of variants is zero. For many practical problems, there is a fixed set of current variants and costs arise when adding variants to the fixed set. A sensitivity analysis of MOCDP could be used for this case as well. The $\mathcal{Q}_{\mathcal{R}}$ measure has to be computed for the current setup. Then MOCDP is solved with m_j equal to the number of added variants in each dimension where the current variants specified by $\hat{\mathbf{y}}$ are added to $\mathbf{y}$ in the computation of the configurations. The improvement in $\mathcal{Q}^*_{\mathcal{R}}$ must now be compared to the cost of adding variants.

To investigate whether an existing variant should be removed is not possible without calculating $\mathcal{Q}_{\mathcal{R}}$ for all possible choices variant removals. That is to say, to analyze whether one variant should be added, n problems need to be solved, one for each dimension. To analyze whether one variant should be removed, $m = \sum_{j=1}^{n} m_j$ problems have to be solved, one for each current variant. The latter problems, however, are very easy since there are no decision variables at all. What has to be done is to compute the quality function $\mathcal{Q}_{\mathcal{R}}$ for the $\sum_{j=1}^{n} m_j$ reduced configuration sets.

5 Numerical experiments

The purpose of this section is to exemplify how the MOCDP can be utilized, by presenting some selected numerical experiments. By using the standard vector-valued test function *kursawe* [15] as the underlying MONP, MOCDP has been formulated and solved with the procedure proposed in Section 3 for different values of

$m_j,\ j = 1, \ldots, n$. We have used a box constrained variable space of dimension three in the MONP.

The objective functions and the feasible region are given by (14).

$$\begin{aligned} f_1(\mathbf{x}) &= \sum_{i=1}^{n-1} \left(-10 \exp\left(-0.2\sqrt{x_i^2 + x_{i+1}^2} \right) \right), \\ f_2(\mathbf{x}) &= \sum_{i=1}^{n} \left(|x_i|^{0.8} + 5\sin^3(x_i) \right), \\ \mathbf{x} &\in [-5, 5]^n. \end{aligned} \tag{14}$$

The *kursawe* function is a standard test function for the evaluation of multi-objective evolutionary algorithms (see [11] for an extensive review).

In step 1 of the solution procedure—to find a representation $\mathcal{R}$ of the Pareto optimal set—we have used `multiOb` [10], a population-based evolutionary algorithm. Examples of other evolutionary-based algorithms for solving MONP's that could be used are `NCGA` [20] and `NSGA-II` [6].

In step 2 of the solution process—in which a global optimization is to be performed—we have chosen the algorithms `DIRECT` [14, 7] and `NEWUOA` [17] to be used in sequence. The former is a space-filling algorithm sampling the decision space around points that either have low objective values or are far from already sampled points. The termination criterion for `DIRECT` can be the number of space-dividing iterations or the number of function evaluations. The output from the algorithm—the best point measured so far—is then provided as a starting point for `NEWUOA`. This is a local optimization algorithm for unconstrained derivative-free single-objective optimization based on quadratic approximations of the objective function.

An approximation of the image of the Pareto optimal set (found by applying `multiOb` with 2000 generations and with a population size of 4000) is shown in Figure 7.

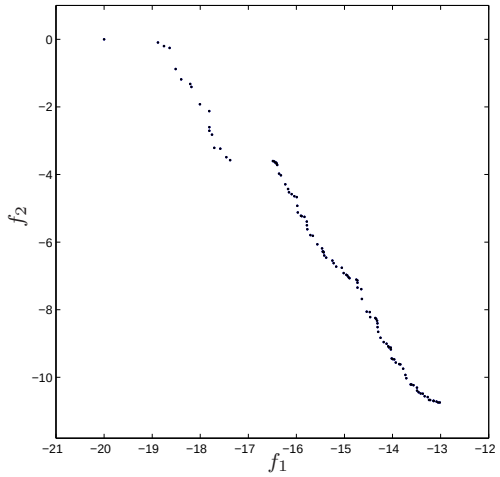


Figure 7: An approximate Pareto front for the test problem *kursawe* with $n = 3$.

Many other test problems in the literature are limited to two decision variables and/or have a Pareto optimal set that has a special structure in X that seems unnatural for a practical problem. To generate more test problems we have kept the objectives of (14) but chosen to rotate the decision space for the first objective. That is, we let the objectives be $\{f_1(A^p\mathbf{x}), f_2(\mathbf{x})\}$, $p = 0, \ldots, 3$, where A^p denote the rotation matrices

$$A^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{(no rotation)}, \quad A^1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \text{(rotation around } x_1\text{)},$$
$$A^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \text{(rotation around } x_2\text{)}, \quad A^3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{(rotation around } x_3\text{)}. \tag{15}$$

This yields four corresponding MONP's, denoted by MONP^p, $p = 0, \ldots, 3$. In the numerical experiments we have for each MONP^p tested all combinations of numbers $[m_1, m_2, m_3]$ of allowed variants in the set $M = \{\mathbf{m} \in \mathbb{N}^3 \mid 1 \leq \sum_{j=1}^{3} m_j \leq 8\}$. We assume here that the variants in the three dimensions are equally expensive and that the important issue is the total number of variants used.

For all numerical results presented below the quality measure

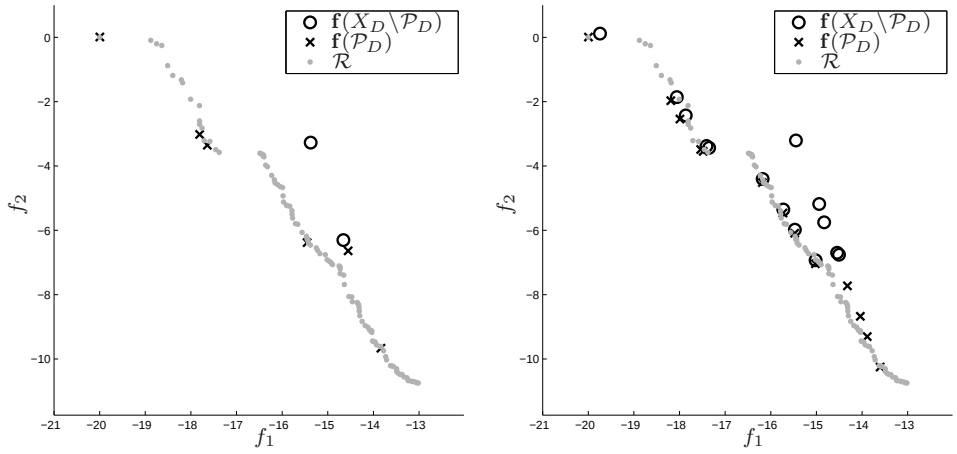
$$\mathcal{Q}_{\mathcal{R}} = 0.01 Dist1_{\mathcal{R}} + 0.99 Dist2_{\mathcal{R}}.$$

The first step of the solution procedure is to find a reference set $\mathcal{R}$ to use in the second step and in the sensitivity analyses. We applied `multiOb` to each MONP^p

using 2000 generations and a population size of 4000 to generate the corresponding reference set $\mathcal{R}^p$, selected to be all found non-dominated points.

In the second step of the procedure we applied `DIRECT` with a termination criterion defined as a maximum number of function evaluations. The best point found by `DIRECT` was then used as a starting point for `NEWUOA`.

The Figures 8 (a) and (b) show the objective values for the solutions found to MOCDP for MONP^0 using $\mathbf{m} = [2, 2, 2]$ and $\mathbf{m} = [3, 3, 3]$, respectively. It is interesting to note that, even if this it not the aim, our algorithm in the second solution step manages to find solutions that dominate parts of $\mathcal{R}$ (at $(f_1, f_2) \approx (-18, -3)$ both in Figure 8 (a) and (b)). It is not a large part of $\mathcal{R}$ that is dominated. However, the computational time for finding $\mathcal{R}$ in the first solution step was around 10 minutes while the time for the second step of solving the MOCDP was only around 20 seconds. It may be possible to create a new class of algorithms for solving multi-objective optimization problems based on ideas similar to ours. Another interesting observation is that the resulting solutions in X_D seem to form a good approximation at the "knee" regions of $\mathbf{f}(\mathcal{R})$ which have the character that a small improvement in either objective will cause a large deterioration in the other (see Figure 8(a)). The knee regions are the most interesting solutions for decision makers whose evaluation of the trade-offs between the conflicting objectives are relatively constant.



(a) 2 variants in each dimension. (b) 3 variants in each dimension.

Figure 8: The objective space with the solutions found using MONP^0 as the underlying problem and with $\mathbf{m} = [2, 2, 2]$ and $\mathbf{m} = [3, 3, 3]$, respectively.

The Tables 1–4 contain results on the quality measures found together with their corresponding variant distributions. Due to space limitations we present the results for the subset of combinations for which $\max_{i,j}\{|m_i - m_j|\} \leq 1$ only. In each row, i.e., for each value of m, the best solution(s) is (are) written in bold face.

$\sum_j m_j$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$
3	[1 1 1]	**0.555**				
4	[1 1 2]	**0.344**	[1 2 1]	**0.344**	[2 1 1]	**0.344**
5	[1 2 2]	**0.222**	[2 1 2]	0.331	[2 2 1]	**0.222**
6	[2 2 2]	**0.100**				
7	[2 2 3]	0.064	[2 3 2]	0.081	[3 2 2]	**0.063**
8	[2 3 3]	0.058	[3 2 3]	**0.046**	[3 3 2]	0.058
9	[3 3 3]	**0.046**				
10	[3 3 4]	0.048	[3 4 3]	**0.045**	[4 3 3]	0.048
11	[3 4 4]	**0.040**	[4 3 4]	0.042	[4 4 3]	0.048
12	[4 4 4]	**0.040**				

Table 1: Numerical results for the solution of MOCDP with different numbers of variants of the three underlying decision variables. The underlying multi-objective problem is MONP^0.

$\sum_j m_j$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$
3	[1 1 1]	**0.558**				
4	[1 1 2]	0.345	[1 2 1]	**0.344**	[2 1 1]	**0.344**
5	[1 2 2]	**0.221**	[2 1 2]	**0.221**	[2 2 1]	0.329
6	[2 2 2]	**0.104**				
7	[2 2 3]	0.072	[2 3 2]	**0.061**	[3 2 2]	**0.061**
8	[2 3 3]	0.061	[3 2 3]	0.061	[3 3 2]	**0.046**
9	[3 3 3]	**0.046**				
10	[3 3 4]	0.047	[3 4 3]	**0.046**	[4 3 3]	**0.046**
11	[3 4 4]	0.046	[4 3 4]	0.045	[4 4 3]	**0.041**
12	[4 4 4]	**0.041**				

Table 2: Numerical results for the solution of MOCDP with different numbers of variants of the three underlying decision variables. The underlying multi-objective problem is MONP^1.

$\sum_j m_j$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$
3	[1 1 1]	**0.551**				
4	[1 1 2]	**0.342**	[1 2 1]	**0.342**	[2 1 1]	**0.342**
5	[1 2 2]	**0.220**	[2 1 2]	0.331	[2 2 1]	**0.220**
6	[2 2 2]	**0.104**				
7	[2 2 3]	**0.064**	[2 3 2]	0.076	[3 2 2]	**0.064**
8	[2 3 3]	0.064	[3 2 3]	**0.049**	[3 3 2]	0.064
9	[3 3 3]	**0.053**				
10	[3 3 4]	**0.049**	[3 4 3]	0.053	[4 3 3]	**0.049**
11	[3 4 4]	**0.049**	[4 3 4]	**0.049**	[4 4 3]	0.050
12	[4 4 4]	**0.049**				

Table 3: Numerical results for the solution of MOCDP with different numbers of variants of the three underlying decision variables. The underlying multi-objective problem is MONP[2].

$\sum_j m_j$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$	$\mathbf{m}$	$\mathcal{Q}^*_{\mathcal{R}}$
3	[1 1 1]	**0.557**				
4	[1 1 2]	**0.344**	[1 2 1]	**0.344**	[2 1 1]	**0.344**
5	[1 2 2]	0.329	[2 1 2]	**0.221**	[2 2 1]	**0.221**
6	[2 2 2]	**0.101**				
7	[2 2 3]	**0.065**	[2 3 2]	**0.065**	[3 2 2]	0.084
8	[2 3 3]	**0.049**	[3 2 3]	0.065	[3 3 2]	0.065
9	[3 3 3]	**0.058**				
10	[3 3 4]	0.049	[3 4 3]	**0.048**	[4 3 3]	0.058
11	[3 4 4]	**0.041**	[4 3 4]	0.049	[4 4 3]	0.048
12	[4 4 4]	**0.046**				

Table 4: Numerical results for the solution of MOCDP with different numbers of variants of the three underlying decision variables. The underlying multi-objective problem is MONP[3].

An important point that has to be kept in mind is that this optimization problem is non-convex and non-linear. Thus, there is no guarantee for the optimality of the solutions found. Study Table 4 and compare for $\mathbf{m} = [3, 4, 4]$ and $\mathbf{m} = [4, 4, 4]$. The latter corresponds to a relaxed MOCDP compared to the former; however it possesses a higher objective value. This shows an optimality gap, i.e., a relative distance from the global optimum, of at least $\frac{0.046-0.041}{0.041} \approx 12\%$ for the latter problem.

The explanation for the similarity of the results in the four problems and for the frequent non-unique solutions found for a certain number of variants comes from the symmetries in the underlying MONP's. One interesting point is that it is not always advantageous to use the largest number of variants in a certain dimension. See Table 4 and compare the rows corresponding to $m = 5$ and $m = 8$. In the

former, an optimal distribution of variants is $\mathbf{m} = [1, 2, 2]$ however $\mathbf{m} = [2, 3, 3]$ is not an optimal choice in the latter. Another point which might be interesting in a real application is that (however, we state no generality of this), it is possible to reach optimal variant distributions at all m-levels by local steps, adding one variant at a time, moving to the optimal distribution.

From the Tables 1–4 it seems like the solution sets are being "saturated", meaning that adding more variants do not decrease the objective value significantly. One algorithmic reason for this is that the larger m, the higher dimensional space has to be (globally) searched. Since the global search is limited by the number of function evaluations, this means that the quality of its output decreases when the dimension of the decision space is increased.

6 Extending to simulation-based MOCDP's

The purpose of this section is to is propose how to adapt the procedure developed to MOCDP for the case when MONP belongs to the class of simulation-based optimization problems.

A general simulation-based optimization problem has expensive objective (or constraint) function(s), e.g., involving computationally intense simulations. Such problems require special treatment since the total number of function evaluations is limited. A simulation-based MOCDP is a problem in which at least one of the objective functions $\{f_1, \ldots, f_k\}$ of the underlying MONP is expensive.

When solving MOCDP, a very large number of function evaluations is required in step 2 of the solution process, since for each variable vector $\mathbf{y}$, a total number of $|X_D| = \prod_{j=1}^{n} m_j$ configurations must be evaluated.

The good thing, however, is that the solution procedure is divided into two steps and that step 1 can be used not only for finding a good reference set $\mathcal{R}$. Simultaneously, it can be used for constructing explicit, computationally cheap response surfaces $\{\hat{f}_1, \ldots, \hat{f}_k\}$ (see [13, 2]) that can be used instead of the expensive simulation-based functions in step 2. The response surfaces can be continuously updated during step 1, such that, by using the response surfaces within step 1, the number of expensive function evaluations also in this step is limited. In [12] the algorithm `qualSolve` is described. This algorithm uses radial basis functions [9] with the aim of approximating the expensive functions that are sampled iteratively such that a certain quality measure is maximized. The algorithm can be applied to multi-objective optimization problems, and the quality measure is then related to how good the approximations are in regions near the Pareto optimal set of the approximating problem. `qualSolve`, or a similar algorithm, can be used in step 1, producing response surfaces to use in place of the original functions from there on.

As already stated, for sensitivity analyses, which might constitute a large part of the computation time for a real application of the MOCDP-procedure, the expensive functions must only be used once, since only step 2 is repeated when these analyses are carried through. Furthermore, even in step 1 the number of expensive function calls must not be that large, since the algorithm (e.g., `qualSolve`) mostly uses its

current response surfaces and only now and then samples the original functions.

7 Conclusions and future work

We have presented a two-step procedure which can be regarded as an implicit clustering of points in the objective space of a multi-objective optimization problem such that the structure of the points in the decision space is controlled.

We have demonstrated the procedure on some test problems and discussed the potential of using different types of sensitivity analyses to perform depending on the actual application.

We have also proposed how the procedure can be adapted to simulation-based problems for which the number of (expensive) function evaluations must be kept low. The solution procedure consists of two steps and we have discussed how the first step can be used for finding computationally cheap approximate functions to use instead of the original ones in the second step. Thus, by construction of the method, the large number of function calls that have to be made in the second step is not a bigger issue for simulation-based MONP's than for regular MONP's with explicit objective functions.

The results are encouraging and we see a potential to apply the methodology to many real-world problems in industry.

There are some issues that should be addressed in order to adapt the current methodology to a larger class of problems such that it will apply to more real-world problems. One improvement would be to develop the procedure presented such that it can handle more general constraints than box constraints in MOCDP. Examples of such are general linear and non-linear constraints on the decision variables. Other examples are constraints on the objective function values in the underlying MONP. Finally, constraints in the decision space that are more connected to real configuration applications are important to govern, e.g., that combinations of certain values of the decision variables are forbidden.

In the original formulation of MOCDP, the decision variables are required to be continuous. For many real-world applications, the decision variables are required to be discrete. Also, the incorporation of the special type of discrete variables, *categorical variables* [1], that can be assigned a discrete number, but where this number has no physical meaning (e.g., representing a certain material, a certain suspension type, etcetera), would substantially increase the range of applications for the procedure presented.

References

[1] C. Audet and J. J. E. Dennis, *Pattern search algorithms for mixed variable programming*, SIAM Journal on Optimization, 11 (2001), pp. 573–594.

[2] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, Trosset, and M. W., *A rigorous framework for optimization of expensive functions by surrogates*, 17, pp. 1–13.

[3] W. M. CARLYLE, J. W. FOWLER, E. S. GEL, AND B. KIM, *Quantitative comparison of approximate solution sets for bi-criteria optimization problems*, Decision Sciences, 34 (1) (2003), pp. 63–82.

[4] P. CZYŻAK AND A. JASZKIEWICZ, *Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis, 7 (1998), pp. 34–47.

[5] K. DEB, *Multi-objective optimization using evolutionary algorithms*, Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons Ltd., Chichester, England, 2001. With a foreword by David E. Goldberg.

[6] K. DEB, S. AGRAWAL, A. PRATAB, AND T. MEYARIVAN, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, in Proceedings of the Parallel Problem Solving from Nature VI Conference, Paris, France, 2000, Springer-Verlag, pp. 849–858.

[7] D. E. FINKEL, *DIRECT Optimization Algorithm User Guide*, Tech. Report, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, USA, 2003.

[8] G. GAN, C. MA, AND J. WU, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*, SIAM, Philadelphia, ASA, Alexandria, VA, USA, 2007.

[9] H.-M. GUTMANN, *A radial basis function method for global optimization*, Journal of Global Optimization, 19 (2001), pp. 201–227.

[10] T. HANNE, *A multiobjective evolutionary algorithm for approximating the efficient set*, European Journal of Operational Research, 176 (2007), pp. 1723–1734.

[11] S. HUBAND, P. HINGSTON, L. BARONE, AND L. WHILE, *A review of multiobjective test problems and a scalable test problem toolkit*, IEEE Transactions on Evolutionary Computation, 10 (2006), pp. 477–506.

[12] S. JAKOBSSON, M. PATRIKSSON, J. RUDHOLM, AND A. WOJCIECHOWSKI, *A method for simulation based optimization using radial basis functions*, tech. report, Department of Mathematical Sciences, Chalmers University of Technology and Department of Mathematical Sciences, University of Gothenburg, Göteborg, Sweden, 2008.

[13] D. R. JONES, *A taxonomy of global optimization methods based on response surfaces*, Journal of Global Optimization, 21 (2001), pp. 345–383.

[14] D. R. JONES, C. D. PERTTUNEN, AND B. E. STUCKMAN, *Lipschitzian optimization without the Lipschitz constant*, Journal of Optimization Theory and Applications, 79 (1993), pp. 157–181.

[15] F. KURSAWE, *A variant of evolution strategies for vector optimization*, in Parallel Problem Solving from Nature. 1st Workshop, PPSN I, H. P. Schwefel and R. Männer, eds., vol. 496, Berlin, Germany, 1991, Springer-Verlag, pp. 193–197.

[16] M. PILEGAARD HANSEN AND A. JASZKIEWICZ, *Evaluating the quality of approximations to the non-dominated set*, Technical report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1998.

[17] M. J. D. POWELL, *The NEWUOA software for unconstrained optimization without derivatives*, in Large-scale Nonlinear Optimization, vol. 83 of Nonconvex Optimization and its Application, Springer-Verlag, New York, NY, USA, 2006, pp. 255–297.

[18] S. SAYIN, *Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming*, Mathematical Programming, 87 (2000), pp. 543–560.

[19] A. VIANA AND J. PINHO DE SOUSA, *Using metaheuristics in multiobjective resource constrained project scheduling*, European Journal of Operational Research, 120 (2000), pp. 359–374.

[20] S. WATANABE, T. HIROYASU, AND M. MIKI, *NCGA: Neighborhood cultivation genetic algorithm for multi-objective optimization*, in Proceedings of the Genetic and Evolutionary Computation Conference, E. Cantú-Paz, ed., New York, NY, USA, 2002, pp. 458–465.

[21] E. ZITZLER, L. THIELE, M. LAUMANNS, C. FONSECA, AND V. FONSECA, *Performance assessment of multiobjective optimizers: an analysis and review*, IEEE Transactions on Evolutionary Computation, 7 (2003), pp. 117–132.