Active Robust Optimization

Optimizing for Robustness of Changeable Products

Shaul Salomon

Department of Automatic Control and Systems Engineering

The University of Sheffield

A dissertation submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Control Systems

July 2017

Abstract

To succeed in a demanding and competitive market, great attention needs to be given to the process of product design. Incorporating optimization into the process enables the designer to find high-quality products according to their simulated performance. However, the actual performance may differ from the simulation results due to a variety of uncertainty factors. *Robust optimization* is commonly used to search for products that are less affected by the anticipated uncertainties. *Changeability* can improve the robustness of a product, as it allows the product to be adapted to a new configuration whenever the uncertain conditions change. This ability provides the changeable product with an *active* form of robustness.

Several methodologies exist for engineering design of changeable products, none of which includes optimization. This study presents the *Active Robust Optimization* (ARO) framework that offers the missing tools for optimizing changeable products. A new optimization problem is formulated, named *Active Robust Optimization Problem* (AROP). The benefit in designing solutions by solving an AROP lies in the realistic manner adaptation is considered when assessing the solutions' performance.

The novel methodology can be applied to optimize any product that can be classified as a changeable product, i.e., it can be adjusted by its user during normal operation. This definition applies to a huge variety of applications, ranging from simple products such as fans and heaters, to complex systems such as production halls and transportation systems.

The ARO framework is described in this dissertation and its unique features are studied. Its ability to find robust changeable solutions is examined for different sources of uncertainty, robustness criteria and sampling conditions. Additionally, a framework for *Active Robust Multi-objective Optimization* is developed. This generalisation of ARO itself presents many challenges, not encountered in previous studies. Novel approaches for evaluating and comparing changeable designs comprising multiple objectives are proposed along with algorithms for solving multi-objective AROPs.

The framework and associated methodologies are demonstrated on two applications from different fields in engineering design. The first is an adjustable optical table, and the second is the selection of gears in a gearbox.

Statement of Originality

Unless otherwise stated in the text, the work described in this thesis was carried out solely by the candidate. None of this work has already been accepted for any other degree, nor is it being concurrently submitted in candidature for any degree.

Candidate: Shaul

Shaul Salomon

Supervisor:

Peter J. Fleming

Acknowledgements

I was fortunate enough to have three extraordinary individuals guiding me through this process. Thank you for not only being great teachers and role models, but also for being real friends that are so fun to hang out with.

I would like to thank Dr. Gideon Avigad for introducing me to the world of science, tailoring my unusual PhD program and the financial support to enable it. With endless enthusiasm and devotion you have encouraged me to ask the right questions and refine raw ideas, until they form into a coherent framework.

I would like to thank Dr. Robin Purshouse for giving me such a hard time whenever you felt something is missing, but always making it very clear how confident you are in my work and how sound you think this thesis is. Your advice was priceless. I also acknowledge countless practical learning sessions on the drinking habits in the UK.

It is really hard to stress the honour of having Prof. Peter Fleming being my lead supervisor. With your rich experience and knowledge, you have provided me with focused and precise instructions for conducting the study and writing the dissertation. Thank you for being so modest and kind, for your great sense of humour and for always making sure that everyone around you feels great.

Further thank-yous go to Prof. Carlos Coello Coello and Dr. Oliver Schütze for their warm hospitality in CINVESTAV-IPA, and to Prof. George Knopf for an unforgettable year at UWO. I thank my colleagues from the CODeM group in USFD, Dr. Yiming Yan, Dr. João Duro, Dr. Ioannis Giagkiozis and Prof. Visakan Kadirkamanathan, for the enlightening discussions and the pleasant time together. I would also like to thank the friends we made in Sheffield and London, Ontario for making us feel at home, and for taking care of my family while I was busy.

Finally, I would like to thank my family. My beloved Ofer, I will never forget the huge sacrifice you made, joining me across the oceans on this crazy adventure. Putting your own career on hold, leaving behind family and friends and turning every place we have arrived into a home. My kids, Aviv and Karen, thank you for the endless joy you bring into my life. For spending so many weekends without daddy, and for the purest joy in having me back.

This research was supported by a Marie Curie International Research Staff Exchange Scheme Fellowship within the 7th European Community Framework Programme, and by the Anglo-Israel Association. Additional support has been received from Ort Braude College of Engineering, thanks to the vision of its president, late Prof. Yohanan Arzi. I will always cherish his memory.

Dedicated to my family:

Ofer, Aviv and Karen.

Contents

Nomenclature

Roman Symbols

A	Amper, first used in p. 145
argmax	Argument of the maximum, first used in p. 46
argmin	Argument of the minimum, first used in p. 46
c	Confidence level, first used in p. 45
$\mathbf{d}$	Direction vector, first used in p. 103
$\mathrm{E}(\cdot)$	Expected value of a random variate, first used in p. 43
$F(\cdot)$	Cumulative distribution function of a random variate, first used in p. 46
$f(\cdot)$	Probability density function, first used in p. 43
$I[\cdot]$	Robustness indicator for a random variate, first used in p. 42
kg	Kilograms, first used in p. 130
l	Lower bound, first used in p. 56
$\mathcal{L}(\cdot)$	Laplace transform, first used in p. 127
max	Maximum, first used in p. 37
min	Minimum, first used in p. 40
m	Metres, first used in p. 130
mm	Millimetres, first used in p. 130
$\mathrm{N}(\mu, \sigma)$	Normal distribution with mean value μ and standard deviation σ, first used in p. 64
Nm	Newton metres, first used in p. 145
N	Newton, first used in p. 130
Pr	Probability of an event, first used in p. 24
$\mathbf{p}$	Vector of uncontrolled parameters, first used in p. 11
q	Quality indicator, first used in p. 22
rad	Radians, first used in p. 130
$\mathbb{R}$	The set of Reals, first used in p. 26
$s(\cdot)$	Scalarisation function, first used in p. 102
s	Seconds, first used in p. 130

$\mathrm{U}(a,b)$	Uniform distribution between a and b, first used in p. 64
u	Upper bound, first used in p. 56
$\mathrm{var}(\cdot)$	Variance, first used in p. 44
V	Volts, first used in p. 145
$\mathbf{w}$	Weighting vector, first used in p. 102
W	Watts, first used in p. 147
$\mathcal{X}$	The feasible domain of decision vectors, first used in p. 26
$\mathbf{x}$	Vector of decision variables, first used in p. 11
$\mathcal{Y}(\mathbf{x})$	Domain of adjustable variables for design $\mathbf{x}$, first used in p. 27
$\mathbf{y}$	Vector of adjustable variables, first used in p. 27
$\mathbf{z}(\cdot)$	Vector of functions, first used in p. 11

Greek Symbols

Δ	Difference operator, first used in p. 23
δ	Deviation from nominal value, first used in p. 42
γ	Stochastic objective that *can* be affected by adaptation, first used in p. 28
Ω	Ohm, first used in p. 145
ϕ	Stochastic objective that *cannot* be affected by adaptation, first used in p. 28
ψ	Deterministic objective, first used in p. 26
Σ	Summation operator, first used in p. 16
σ	Standard deviation, first used in p. 38

Superscripts

$\dot{a}$	First derivative in time of a, first used in p. 127
$\ddot{a}$	Second derivative in time of a, first used in p. 127
(i)	Candidate solution i, first used in p. 93
i	i^{th} member of a set, first used in p. 104
$\star$	Optimal solution, first used in p. 29

Subscripts

c	Confidence-based, first used in p. 46
E	Expected value, first used in p. 43
nom	Nominal value, first used in p. 78
q	Target-based, first used in p. 45
ss	Steady-state, first used in p. 155
tr	Transient, first used in p. 155
v	Variance, first used in p. 43

w	Worst-case, first used in p. 42

Grouped and Random Variables

$[a_1, \ldots, a_n]$	Elements of a vector listed in square brackets, first used in p. 26
$\mathbf{a}$	Vector, marked with bold font, first used in p. 26
$\{\mathbf{b}_1, \mathbf{b}_2\}$	Members in a set listed in curly brackets. May also include a rule, e.g., $\left\{\mathbf{a} \in \mathbb{R}^2 \mid a_1 = a_2\right\}$, first used in p. 27
$\underline{\mathbf{b}}$	Set of elements of the same type, marked with underline, first used in p. 27
C	Random variate of the variable c, first used in p. 27
$\bar{C}$	Sampled representation of the random variate C, first used in p. 27

Other Symbols

$\lvert\bar{\mathbf{U}}\rvert$	Cardinality of sampled set $\bar{\mathbf{U}}$, first used in p. 44
$\lvert a\rvert$	Absolute value of the number a, first used in p. 62
$:=$	Definition, first used in p. 11
\$	American dollars, first used in p. 145
$\equiv$	Equivalence, first used in p. 30
$\int_{\mathbf{U}}$	Integration over all scenarios of multivariate $\mathbf{U}$, first used in p. 44
$\lVert\mathbf{a}\rVert_k$	k^{th} norm of the vector $\mathbf{a}$, first used in p. 23
$\lVert$	Incomparable, first used in p. 33
$\prec$	Dominance relation, first used in p. 29
$\preceq$	Weak dominance relation, first used in p. 33
$\prec\!\prec$	Strict dominance relation, first used in p. 33
$\sim$	Probability distribution, first used in p. 64
$\lhd$	Better-than relation for sets, first used in p. 33

Acronyms

AD	Adaptable Design, first used in p. 19
AF	Approximated Front, first used in p. 30
ARMOP	Active Robust Multi-Objective Optimization Problem, first used in p. 5
ARO	Active Robust Optimization, first used in p. iii
AROP	Active Robust Optimization Problem, first used in p. iii
AS	Approximation Set, first used in p. 30
CDF	Cumulative Distribution Function, first used in p. 64
CNC	Computer Numerical Control, first used in p. 18
DE	Differential Evolution, first used in p. 115
DM	Decision-Maker, first used in p. 30
DOE	Design of Experiments, first used in p. 15

DOP	Dynamic Optimization Problems, first used in p. 40
EA	Evolutionary Algorithm, first used in p. 39
EMOA	Evolutionary Multi-Objective Optimization Algorithm, first used in p. 31
EMO	Evolutionary Multi-Objective Optimization, first used in p. 35
HV	Hypervolume, first used in p. 35
ISS	International Space Station, first used in p. 1
MOO	Multi-Objective Optimization, first used in p. 29
MOP	Multi-Objective Optimization Problem, first used in p. 7
MSD	Mean Square Deviation, first used in p. 16
OAP	Optimization of Adaptation Problem, first used in p. 7
PDF	Probability Density Function, first used in p. 62
PF	Pareto-Optimal Front, first used in p. 30
RMOP	Robust Multi-Objective Optimization Problem, first used in p. 91
RMS	Reconfigurable Manufacturing Systems, first used in p. 18
ROI	Region of Interest, first used in p. 30
RO	Robust Optimization, first used in p. 40
SNR	Signal to Noise Ratio, first used in p. 15
SOP	Single-objective Optimization Problem, first used in p. 30
UQ	Uncertainty Quantification, first used in p. 75

List of Figures

List of Tables

Chapter 1

Introduction

Optimization is a powerful tool that allows a designer to select the appropriate design variables to achieve high performance products. Many real-world optimization problems involve uncertainties. A solution for such a problem is expected to be robust to these uncertainties. Commonly, robustness is attained by designing the solution such that its performance is less influenced by negative effects of the uncertain parameters' variations. This robustness may be viewed as a passive robustness, because once the solution's design variables are chosen, the robustness is inherent in the solution and no further action is expected to suppress the effect of uncertainties.

This study deals with systems and products that can achieve robustness in an *active* manner. In contrast to the conventional approach for designing robust products, *active robustness* is attained by including some adjustable features in the product design, thus making it a *changeable* product. These features enable the changeable product to respond to variations of parameters and mitigate performance degradation in a cost-effective manner.

For example, consider the manner in which the international space station (ISS) is designed to protect itself against collisions with space debris. While orbiting our planet, the ISS is in a constant threat of getting hit by space debris, made of meteoroids and wrecked satellites. To ensure the safety of its crew and equipment, the space agencies could have adopted a passive robustness approach and shield the station with a *very* thick armour. This armour could have protected the ISS from some of the space debris, but the added weight to the station's modules would make the whole project infeasible. Instead, a thin and lightweight shield is installed that can protect the station from objects up to the size of 1cm. Larger objects are avoided by maneuvering the station with its thrusters whenever a collision is predicted (Garcia, 2013).

The American National Aeronautics and Space Administration (NASA) has the ability to track the objects orbiting the Earth and calculate their path within a high degree of precision. Therefore, the robustness against collisions, provided by the maneuverability of the ISS, is higher than the passive robustness that could have been attained by a heavy armour. Furthermore, implementing this ability does not come at the cost of design infeasibility due to overweight.

Many changeable products are designed and manufactured in industry today. A few examples are wind turbines, irrigation systems, automobiles and reconfigurable robots. All of these products are designed to operate in a changing environment, and can be adjusted to perform in an optimal manner whenever the environmental conditions change. Despite some guidelines existing in the literature on how to incorporate changeability into the design (Koren et al., 1999; Gu et al., 2004; Siddiqi et al., 2006; Haldaman and Parkinson, 2010), optimization is not taking place in the design process. The reason for this is the lack of a methodology to support such action. In order to optimize changeable products, a designer needs the ability to compare the predicted performance of alternative solutions. To do so, the effects of the adjustable features on predicted performance, subject to the various uncertainty factors, must be understood.

In this study, the *Active Robust Optimization* methodology for optimizing changeable products is presented. It considers products that are able to adapt to varying conditions by reconfiguration of some adjustable properties. The ability of the ARO approach to optimize changeable products is rooted in the manner in which the evaluation functions are modelled. A distinction is made between three types of variables that affect the performance of the candidate design:

1. **Parameters** that cannot be controlled by the designer, some of which are uncertain.

2. **Fixed decision variables** that can be decided during the design phase. These variables define the solution.

3. **Adjustable decision variables** that can be changed by the user during product operation, in response to actual changes of the previously uncertain parameters.

The choice of which adjustable features to include in the product is made during the design phase, and therefore, it is determined by the fixed decision variables. This choice defines the solution's range of adaptability. The configuration of these features, determined by the user, is represented by adjustable decision variables.

Using the above distinction, a new optimization problem named *Active Robust Optimization Problem* is formulated. The AROP considers the influence of adaptation on the candidate solution's performance. With the aid of criteria for selecting solutions according to their variate of performance function(s), a robust optimization scheme is conducted to find a high quality changeable solution.

The novel methodology can be applied to optimize a great variety of products in many different fields, ranging from simple products such as fans and heaters to complex systems such as manufacturing halls and transportation systems. In fact, any product that has several operation modes, aimed for different operating conditions, can be optimized with the proposed approach, as long as its performance can be predicted for different combinations of the uncertain parameters and decision variables.

1.1 Motivation

The main goal of this study is to establish a new engineering design methodology, aimed at products that can cope with uncertainties in a cost-efficient manner through adaptation. The methodology should be based on optimization to support a decision on which properties of the products should be made changeable, and to what extent. In developing the methodology, the following research gaps should be addressed:

1. **Develop a framework for robust optimization of changeable products**
   - (a) Provide a mathematical definition for a changeable product. This should make it clear whether or not a certain product can be optimized with the proposed approach.
   - (b) Formulate the Active Robust Optimization Problem.
   - (c) Understand the effects of various factors on the problem and its solution. This includes the types of uncertainty, definition of robustness and algorithmic issues.

2. **Extend the framework to consider multiple conflicting objectives**
   - (a) Extend the notion of optimality of changeable products when optimizing for multiple conflicting objectives.
   - (b) Suggest evaluation measures for the robustness of changeable products in a multi-objective setting.
   - (c) Suggest optimization methods that can incorporate the above evaluation measures to find a robust changeable product when multiple objectives are concerned.

3. **Demonstrate the framework**
    (a) Present a simple, easy to follow, analytic example that highlights the various issues that may arise when solving AROPs. The analytic example can be used as a reference when optimizing more complicated real-world applications.
    (b) Demonstrate how the framework can be applied to real-world design activities. This should include how models of changeable products can be constructed to simulate their performance within the optimization process.

1.2 Outline of the Thesis

This dissertation is composed of six chapters. Following the introduction in this chapter, Chapter 2 includes a literature review of the related research fields and provides the required background to understand the rest of the dissertation. In Chapter 3, the Active Robust Optimization problem is formulated, explained and analysed. Chapter 4 includes a generalisation of the AROP to applications that include multiple conflicting objectives. In Chapter 5, the methodology is demonstrated on two applications from engineering design. Chapter 6 concludes the thesis. The content of each chapter is described in the outline below.

- Chapter 2 surveys the relevant literature in the fields of optimization and engineering design that handle uncertainties and changeable systems. First, the types of uncertainties considered in engineering design are classified and their sources are explained. Next, the existing design paradigms for designing products that are robust to uncertainties are surveyed. The manner in which changeability is incorporated into the product to cope with uncertainties is given special attention, and existing measures for changeable products are examined. Following this, the concept of optimization is explained. Multi-objective optimization, dynamic optimization and set-based optimization are explained and common methods for solving optimization problems are presented. In order to understand the notation that is used throughout the dissertation, the nomenclature, that appears in a table form in the preface, is explained in detail before it is first used. The existing literature on robust optimization is presented. This specifically focuses on studies concerning robust multi-objective optimization and the robust optimization of changeable products. Finally, gaps in the current literature are identified, and the location of this research is illustrated with respect to the current state-of-the-art.
- Chapter 3 establishes the foundations of the active robust optimization framework. It starts with the formal definition of the active robust optimization problem and

its building blocks. Then, an analytic example for an AROP is presented and its characteristics are examined. Next, the difference between robustness and *active* robustness is demonstrated for various descriptions of optimality in the presence of uncertainties. Following that, further analysis is conducted, such as the effect of sampling size of the uncertainty domain on the obtained solution, and variations of the AROP that consider various types of uncertainty.

- Chapter 4 extends the basic AROP to problems with multiple objectives. At the beginning of the chapter, a formal definition of the *Active Robust Multi-Objective Optimization Problem* (ARMOP) is presented and its notion of optimality is discussed. Then, the unique features of an ARMOP are demonstrated through an analytic example, which is a multi-objective extension to the example used in Chapter 3. The complexities are added into the problem one-by-one in order to understand the effects of each feature. Once the problem is understood, several strategies for evaluating and comparing the performance of candidate solutions of an ARMOP are suggested. This evaluation is a very challenging task due to the ability of a changeable product to adapt to several, equally optimal, configurations for every realisation of the uncertainties. The chapter ends with high-level descriptions of algorithms that can be constructed according to the suggested evaluation approaches.

- Chapter 5 demonstrates how the methodology can be applied to a variety of real-world applications. Both single and multi-objective formulations AROPs are presented for two applications from different fields in engineering design. The first is the optimization of components of an adaptable optical table, and the second is the optimization of a gearbox for an uncertain load demand.

- Chapter 6 concludes the dissertation. The key results are highlighted and the contributions of this thesis are discussed in detail. Then, the limitations and caveats in using the presented framework are addressed. Finally, additional research and new directions are identified.

1.3 Contributions

Main contributions

1. **Framework for Active Robust Optimization**. The framework provides the tools to optimize changeable products. It is based on a new class of optimization problems–the *Active Robust Optimization Problem.* The AROP considers the

uncertain conditions in which the product is expected to operate, and the ability of the product to respond to changes of these uncertain conditions.

The methodology was first introduced in Salomon et al. (2014), and part of the analysis provided in this thesis was published in Salomon et al. (2016a). The methodology is described, demonstrated and analysed in Chapter 3.

2. **Framework for Active Robust Multi-Objective Optimization**. The extension of the ARO framework to optimize for multiple performance criteria has many unique features that are not present in existing multi-objective optimization problems. These features are described, and the challenges they present are discussed in Chapter 4. Some strategies for addressing these challenges are suggested in Chapter 4 as well.

   The *Active Robust Multi-Objective Optimization Problem* was first introduced in Salomon et al. (2015).

3. **Metrics for comparing ARMOP solutions**. The performance of a candidate solution to an ARMOP can be described as a set of alternative objective vectors for every realisation of the uncertainties. When the entire uncertain domain is considered, the performance becomes a variate of sets. Some metrics to evaluate and compare changeable products according to their variates of sets are suggested in Chapter 4. These metrics are based on different approaches for preference elicitation in multiojbective optimization. The metrics are described and demonstrated, and the advantages and disadvantages of each approach are highlighted.

   One metric was presented in Salomon et al. (2015). The rest are presented in this thesis for the first time.

4. **Two Case Study applications**. To demonstrate the novel methodology, two applications from the field of engineering design were conceived and presented in Chapter 5:

   (a) *An optical table with relocatable supports*. This new design of an optical table has proven to better absorb floor vibrations than the existing design with fixed supports. The case study includes a mathematical model of the product, derived from first principles.

   The concept was first introduced in Salomon et al. (2014).

   (b) *Gearbox optimization for uncertain load scenarios*. The case study includes a novel perspective for gearbox optimization, where the varying load is treated as an uncertain entity with an estimated distribution. A model of the electric motor and gearbox was derived from first principles.

This work was partly published in Salomon et al. (2015) and Salomon et al. (2016a).

Additional contributions

1. **Review of Robust Multi-Objective Optimization**. Despite the increasing interest in robust multi-objective optimization, a systematic review of the existing approaches for modelling and solving uncertain multi-objective optimization problems (MOPs) cannot be found in the literature. Section 2.4.2 consists of a survey of the existing methods in which uncertain MOPs can be constructed, and the definitions of robustness that are used to solve this type of problem.

2. **Conceiving an elegant and simple problem to demonstrate all the issues arising in AROPs**. The analytic function that is used to guide the reader through the complexities of the framework is a very simple trigonometric expression. It consists of the smallest possible number of objectives, constraints, decision variables and uncertain parameters that can still be used to formulate an AROP. This enables the reader to recognise the role of every component of the problem, and to understand how changes in each component affect the performance of candidate solutions. Despite the problem's simplicity, it includes all the required features for observing the special properties of AROPs.

   The function is first presented to construct a single-objective AROP in Chapter 3 and is slightly modified to construct an ARMOP in Chapter 4.

3. **Introducing the Optimization of Adaptation Problem**. The AROP considers the performance of the changeable design after it has optimally adapted to the changing conditions, which are uncertain during the design stage. A related study (Salomon et al., 2013a), that was conducted alongside with the development of the AROP, addresses the following question: *What is the right way to adapt once the operating conditions have changed?* In other words, it searches for the optimal trajectory of configuration in time during the transition phase. In contrast to the *Optimal Control* approach that minimizes the difference from the new optimum in decision space, the approach presented in Salomon et al. (2013a) minimizes the difference from the new optimum in objective space. It is formulated as an optimization problem, termed *Optimization of Adaptation Problem* (OAP), that minimizes control effort and deviation from the new optimum in objective space.

The study was omitted from the scope of this thesis since the ARO framework is already complete without it. The OAP can be used within an AROP to evaluate an aspect of a candidate solution's performance, but an AROP can also be formulated without it.

Chapter 2

Background

The aim of this study is to suggest a framework for optimizing changeable products. Such a framework can be incorporated into the process of engineering design of products that need to adapt to changes during their normal service. At the stage of product design, the changing environment results in an uncertainty regarding the exact operating conditions. The foundations of this study are rooted in the fields of optimization and engineering design. The ways uncertainties and changeability are addressed in each of these disciplines serve as the starting point for the development of the methodology.

In order to properly position the proposed methodology within the existing literature, the relevant research fields that deal with one or more of the above topics need to be reviewed. Figure 2.1 provides an overview of these research fields, and positions them within the context of engineering design, optimization, uncertainties and changeability.

In this chapter, relevant literature on each of these related research areas is surveyed. The gap in the current art is identified, and the research required for filling this gap is highlighted.

2.1 Uncertainties in Engineering Design

Engineering design is the process in which a product is developed to achieve a desired functionality. The result of the process is a detailed set of instructions for product manufacturing. The process usually includes the following stages: identifying the need, specifying the requirements, suggesting several concepts and choosing the most promising one, detailed design, choosing parameters, simulations, experiments and possibly – if the results are not satisfactory – redesign. Throughout the design process, the designer has to consider uncertainties of several types. These uncertainties may affect the quality of the design and its cost, as well as the design process itself. As a result, they have an

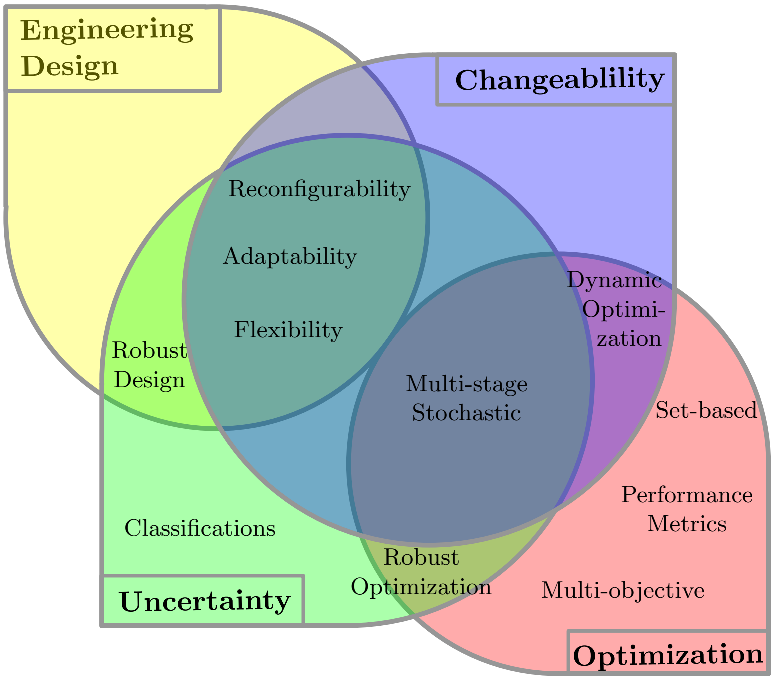


Figure 2.1: Research fields related to this study.

impact on product profitability and customer satisfaction. To survive in a competitive market, manufacturers must be able to address the uncertainties involved and reduce their potentially negative effects. A wide range of approaches to deal with uncertainties during the design process can be found in the literature.

According to Thunnissen (2005), uncertainty is defined as *"the difference between an anticipated or predicted value and a future actual value"*. The properties of uncertainties in the context of engineering design are classified in Section 2.1.1. Then, in Section 2.1.2, another classification is given according to the sources of uncertainty, i.e., the sources for the discrepancy between the actual future value and the one predicted by using a mathematical model.

2.1.1 Types of Uncertainties

The following classification of uncertainties to different types is adopted from Thunnissen (2005) who studied uncertainties in complex system design:

Epistemic uncertainty is any lack of knowledge during the modelling process about the product and the environment. It includes model simplifications, misunderstanding of the real system, human errors and unforeseen behaviour that could not be anticipated until the actual product is first tested. An example of such unforeseen behaviour is when some properties, well defined on their own, interact among themselves in an unpredictable manner.

Aleatory uncertainty refers to an inherent variation of a property's value. While some information such as boundaries and probability might be known, the actual value will vary by chance from unit to unit or time to time. It is usually described by a probability distribution function when included within a mathematical model.

Ambiguity is the type of uncertainty resulting from the usage of spoken language to describe system properties. It might be caused by a misinterpretation of a described property or by linguistic imprecisions that lead to a vague description. Fuzzy sets (Zadeh, 1965) and fuzzy logic (Zadeh, 1968) are commonly used when a mathematical model has to be based on ill-defined properties.

2.1.2 Sources of Uncertainties

To classify the sources of uncertainties in the process of engineering design, a terminology from Beyer and Sendhoff (2007) is adopted. Before the product is produced, the designer has to rely on models and simulations to assess the performance of a potential design. As illustrated in Figure 2.2, the model provides an estimation of the product's performance based on the design variables and the environmental parameters that are given as inputs. This can be mathematically described as follows:

$$\mathbf{z} := \mathbf{z}(\mathbf{x}, \mathbf{p}) \, , \tag{2.1}$$

where $\mathbf{z}$ is a vector of performance measures computed by using a model, $\mathbf{x}$ is a vector of design variables, and $\mathbf{p}$ is a vector of environmental parameters that cannot be determined by the designer. Uncertainties are usually present at every node of the presented scheme. In their review of robust optimization, Beyer and Sendhoff (2007) classified the sources of uncertainties into three types:

Type A: Uncertain environmental conditions. These uncertainties are a result of incomplete information about the requirements and operating conditions. They might also occur due to expected changes in parameter values during a system's operation. This type of uncertainty is modelled by using random values to describe the uncertain $\mathbf{p}$ parameters in Equation (2.1).

Type B: Production tolerances and deterioration. These uncertainties are present when the actual values of design variables differ from their nominal values. The deviation might occur at the production stage (due to manufacturing tolerances)

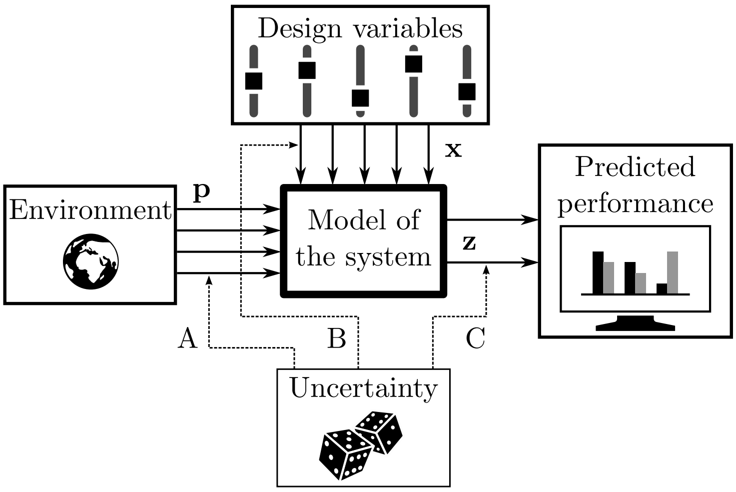


Figure 2.2: Sources of uncertainties during the design process.

or during operation (as a result of deterioration). Here, the design variables, $\mathbf{x}$, in Equation (2.1) are the source of uncertainty.

Type C: Uncertainties in the system output. The actual values of the performance vector usually differ from their simulated values due to model inaccuracies. Model inaccuracies are a result of an incorrect or simplified description of the relationship between variables within the model. Every simplification, and therefore every model, is inaccurate to some extent. The amount of inaccuracy varies from one model to another. Type C uncertainties are caused due to poor understanding of the physical phenomena described by the model, or due to intentional simplifications (such as linearisation) to reduce the model's complexity in order to accelerate its computation. Considering a model described by Equation (2.1), the modelled performance measures $\mathbf{z}$ are the source of uncertainty.

2.2 Design Methods for Coping With Uncertainties

Since the introduction of Taguchi's *Robust Design* methodology (Taguchi, 1987), a wide variety of methods have been developed in order to account for uncertainties during product design. Following Taguchi's approach, these methods aim at products that are less affected by the uncertainties, instead of trying to reduce the uncertainties themselves. A very effective way to improve the robustness of a product to uncertainties is to enable it to react to changes in real-time. Introducing this ability into the product is a useful approach to enhance performance and meet the requirements in the face of uncertainties. Many terms are used in the engineering design literature for the product's ability to change, and different authors use different terminologies to support their design methods. As a result, the same term may have different meanings, or two terms may describe the same attribute.

Chalupnik et al. (2013) suggested a structured framework based on the existing literature to define some terms relating to engineering design under uncertainties, and to highlight the differences and relationships between them. The following attributes were considered in their study: reliability, robustness, versatility, resilience, adaptability and flexibility. These properties were collectively termed as "ilities". Their framework focuses on reliability, i.e. the means of minimising the risk of failure, but similar definitions for the above are also used in the literature in the context of maximising performance (see e.g., Phadke, 1989; Koren et al., 1999; Saleh et al., 2009; Beyer and Sendhoff, 2007).

The *ilities* are classified according to the type of uncertainties they come to suppress, and whether they do it in an active or a passive manner.

Robustness is defined in this context as *"the ability of a system, as built/designed, to do its basic job in an uncertain or changing environment"*.

Versatility is defined as a passive form of protection against uncertainties associated with uncertain requirements, or in other words, *"the ability of a system to perform several tasks without changing its configuration"*.

Resilience is defined as a passive form of protection against uncertainties associated with both uncertain requirements and environmental conditions. Formally, *"the ability of a system to perform several tasks in uncertain or changing environment without changing its configuration"*.

Adaptability is defined as *"the ability of a system to be modified in order to do its basic job in an uncertain or changing environment"*. Both robustness and adaptability address uncertainties in the environmental conditions. Robustness does it without further action, while adaptability enables an active response to environmental changes.

Flexibility is defined as *"the ability of a system to change its states to meet new requirements or to operate in a new environment"*. It is an active form of protection against uncertainties associated with both uncertain requirements and environmental conditions.

Reconfigurability was not classified by Chalupnik et al. (2013), although it appears in many related studies and can be defined in the nature of this list. Following

Table 2.1: A classification of conceptual approaches to system protection against uncertainty (adapted from Chalupnik et al., 2013).

"Ility"	**Active/ Passive**	**Changing environment**	**Changing requirements**
Robustness	Passive	$\surd$	
Versatility	Passive		$\surd$
Resilience	Passive	$\surd$	$\surd$
Adaptability	Active	$\surd$	
Reconfigurability	Active		$\surd$
Flexibility	Active	$\surd$	$\surd$

definitions of Koren et al. (1999); Siddiqi et al. (2006); Haldaman and Parkinson (2010), reconfigurability is defined as *"the ability of a system to change its configuration repeatedly and reversibly to meet multiple requirements"*. It can be perceived as the active counterpart of versatility. Instead of possessing several functionalities to address several requirements, each requirement is associated with a different configuration. It is noted that the extent of reconfigurability should be no more and no less than required to address the intended set of requirements (Koren et al., 1999).

Table 2.1 summarises the differences and similarities between the approaches above. The existing design methodologies for applying some of the above conceptual approaches are surveyed in the reminder of this chapter. Special attention is given to the active forms of protection against uncertainties: adaptability, reconfigurability and flexibility. The Robust Design methodology also appears in this survey due to its significant contribution to the field of engineering design under uncertainties. Since there is no unified terminology in the literature for terms such as adaptability and reconfigurability, each method is classified according to its aim, rather than the terminology used by its authors. For example, the "Adaptable Design" methodology is described under "Design for Reconfigurability" as it aims at products that satisfy multiple requirements.

2.2.1 Robust Design

Robust Design is the first structured methodology to incorporate protection against uncertainties into the engineering design procedure. The methodology aims at products that are robust to disturbances caused by uncontrollable parameters. Eliminating the source of uncertainties can be costly (e.g. minimising manufacturing variations) or impossible (e.g. when fluctuations in operating conditions are concerned). Hence, the underlying principle is to reduce the effects of these uncertainties without eliminating their source. In other words, the aim is a design that can accommodate the uncertainties involved.

Genichi Taguchi, “the father of Robust Design”, was the first to present a structured methodology to account for uncertainties in the search for high quality, low cost and robust products (Taguchi, 1987; Phadke, 1989). His seminal work contributed to Japan's industrial rehabilitation after World War II, when it faced a severe shortage in high quality materials and manufacturing equipment. The tools provided by Taguchi's methods enabled the Japanese industry to produce high quality products despite these conditions, leading Japan to become a dominant industry in the international markets in many fields such as automotive, photography and electronics.

The Robust Design methodology is, in fact, an optimization scheme, aiming at finding the optimal values for a set of design parameters, considering different scenarios of the noise factors. Noise factors are considered by Taguchi as alternating parameters that are impossible or too costly to control. They include Type A and Type B uncertainties (i.e. environmental parameters, manufacturing variations and deterioration). The two major contributions of the approach are the objective function and the search mechanism, namely *Signal to Noise Ratio* and *Orthogonal Arrays*, respectively.

Instead of using an automated optimization procedure, as used in common optimization approaches, Taguchi's method relies on *design of experiments* (DOE) in order to evaluate different designs. The DOE is efficiently set with the use of Orthogonal Arrays (Rao, 1947). Each variable is sampled by a small number of discrete values (typically 2-3), and a relatively small number of experiments is conducted, where a different combination of the variables is used for each experiment. The values of the variables are systematically changed at each experiment according to an orthogonal lattice. Every combination of values between every two variables exists in exactly one experiment. An example for an orthogonal array with four variables, each with three possible values, is shown in Table 2.2. Note that only nine experiments are required in order to examine all combinations between pairs of variables' values. For simulation-based robust design, Taguchi proposed to simulate each of the parameter settings with a similar set of values of the noise factors. The set is also constructed by assigning discrete values to each noise factor, and constructing an orthogonal array of scenarios. Assuming an array of n experiments and an additional array of k scenarios for the noise factor, the DOE should include nk simulations.

Prior to the introduction of the Robust Design methodology, quality engineering mainly relied on quality inspections, i.e., keeping the performance within the tolerance limits. Taguchi had a different notion of quality, as he focused on keeping the performance close to the target. His aim was to maximise the so-called *Signal to Noise Ratio* (SNR),

Table 2.2: An orthogonal array for conducting nine experiments with four variables, each has three possible values (adapted from Phadke, 1989).

	Experiment number								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
A	1	1	1	2	2	2	3	3	3
B	1	2	3	1	2	3	1	2	3
C	1	2	3	2	3	1	3	1	2
D	1	2	3	3	1	2	2	3	1

described below:
Let $\mathbf{x}$ be the design variables, $\mathbf{p}$ be the noise factors and $\phi(\mathbf{x}, \mathbf{p})$ be a quality characteristic of a product with a target value $\hat{\phi}$. For a certain combination of $\mathbf{x}$, the *Mean Square Deviation* (MSD) is defined:

$$\text{MSD} = \frac{1}{k} \sum_{i=1}^{k} \Big(\phi(\mathbf{x}, \mathbf{p}_i) - \hat{\phi} \Big)^2 \tag{2.2}$$

The SNR is defined as follows:

$$\text{SNR} = -10 \log_{10}(\text{MSD}) \tag{2.3}$$

The SNR metric was used by Taguchi as an objective function that needs to be maximised w.r.t $\mathbf{x}$. Once the various SNR values are calculated, *analysis of variance* techniques (Fisher, 1925, pp. 198–235) are used to decide which parameter setting $\mathbf{x}$ yields the most robust performance, i.e., with smallest deviations from the target value. Design variables that do not affect the SNR are then used to adjust the mean performance to the target.

Since they were first published, Taguchi's methods have been implemented in a wide range of engineering fields. Nevertheless, their efficiency and applicability have been widely criticised as well. In the context of this thesis, the method's most obvious flaw was best stated by Trosset (1996): *"Taguchi's methods attempt to optimize an objective function by specifying all of the values of* $\mathbf{x}$ *at which the objective function will be evaluated* prior to observing any function values. *Thus, the Taguchi approach violates a fundamental tenet of numerical optimization—that one should avoid doing too much work until one nears a solution."* An extensive overview and a debate about Taguchi's methods can be found in the panel discussion of Nair et al. (1992).

2.2.2 Design for Adaptability

The term adaptable or adaptive can be found in several studies in engineering design, but none of them addresses products that perform adaptation solely as a response to changing environmental conditions. The following two studies were identified to address issues of adaptability, as defined in the context of this work:

Siddiqi et al. (2006) suggested a framework to analyse adaptable systems using Markov models. The system states are modelled as nodes, and the states evolve according to the probabilities to adapt from one node to another. Probabilities depend on the difference in performance and on the cost of adaptation between states. The performance of the system depends on the configuration and time-varying environmental conditions, and so, as long as the cost of adaptation is not too high, the system tends to adapt to the optimal state for each environmental scenario.

To demonstrate the approach, a case study of a planetary exploration rover was used. The rover is capable of adjusting its wheel dimensions, both in diameter and width, in response to a change in the type of terrain. Its objectives are to maximise thrust and to minimise power consumption. In their example, Siddiqi et al. (2006) used a weighted objective function, and an ordered sequence of soil types. The simulation results showed that the Markov model of the system always converges to the optimal state (according to the weighted objective function) within a few time steps. It was also noticed that the optimal state was very sensitive to the weighting parameter between the objectives.

This study did not include a comparison between potential adaptable prototypes, but it did include a comparison between the suggested adaptable rover, and one with fixed wheel dimensions. For the chosen sequence of uncertain terrain and different values of the weighting parameter, the superiority of the adaptable rover was demonstrated. The study does not address the question of how to choose the adaptable properties and their limits, and a method to handle the entire set of uncertain conditions was not discussed either.

Ferguson and Lewis (2006) address an important issue of adaptable systems, namely the proper way to change the variables when adapting from one configuration to another. Since there is a correlation between the system's configuration and its performance, the adaptation trajectories should be considered both in design and objective spaces. They pointed out that an adaptation trajectory dictated by high performance may be a complex trajectory in design space. To follow this kind of trajectory a complicated control law is required, which is usually associated with a higher cost and longer adaptation periods. Another important distinction was made between planning the optimal adaptation trajectory, and implementing a controller to follow it.

Similar to other studies of reconfigurable systems in the context of multi-objective optimization (e.g. Denhart, 2013, in Section 2.2.5), Ferguson and Lewis (2006) failed to choose a proper multi-objective optimization problem to demonstrate their approach. The case study used was an adaptive race car that can adjust its centre of gravity, roll stiffness and aerodynamic downforce in order to maximise its velocity in corners with different radii. When choosing the above properties as constant values for the entire race, performance for different radii can be seen as competing objectives, and the designer needs to decide for each track what trade-off is the most suitable. On the other hand, an adaptable car can adjust to the optimal configuration that enables the highest velocity for each corner, and therefore the multi-objective domain can be reduced to a single one, namely maximum velocity.

2.2.3 Design for Reconfigurability

Reconfigurable systems are aimed at efficiently satisfying a set of predetermined requirements. Every configuration should be dedicated to satisfying a specific requirement. The following methodologies can be considered as “design for reconfigurability”:

Reconfigurable Manufacturing Systems

Koren et al. (1999) developed an approach to design *reconfigurable manufacturing systems* (RMS) that are built of modules. These modules, termed Reconfigurable Machine Tools (see also Landers et al., 2001), can be added, removed or replaced when a new product is to be produced. RMS combines the advantages of high accuracy and production rates associated with dedicated manufacturing systems, with the versatility of flexible manufacturing systems such as Computer Numerical Control (CNC). An RMS is designed to have the exact changeability to enable the production of a desired family of products. Six characteristics are required in a system in order for it to be classified as an RMS (Koren and Shpitalni, 2010):

Customisation- changeability is limited to part family.
Convertibility- design for functionality changes.
Scalability- design for capacity changes.
Modularity- modular components.
Integrability- interfaces for rapid integration.
Diagnosability- design for easy diagnostics.

The methodology does not include optimization of the system in order to achieve its goals, which are high production rates, low costs and fast reconfiguration. Instead, it is demonstrated how these goals can be taken into account as part of the design process.

A survey of recent advances in the field of RMS was conducted by Gadalla and Xue (2017). Recently, Koren et al. (2015) stated that the trend in manufacturing goes toward *mass-individualisation*, where manufacturers will produce open platforms, and consumers will be able to develop and use various modules. A similar process already exists in the software industry, where applications can be developed by any user to extend the functionality of smart-phones and tablets.

Adaptable Design

Gu et al. (2004) and Hashemian (2005) presented the *Adaptable Design* (AD) methodology as a design paradigm aiming for both business success and environmental protection. The methodology provides guidelines for considerations during the design process. It addresses adaptability as an extension of the product's utility when additional functionalities are required. These functionalities are not part of the product's normal operation mode. According to Gu et al. (2004), the source of uncertainty that requires structural change is a changing requirement, rather than a changing environment. Therefore, it is described here under reconfigurability and not adaptability. Adaptation is considered as the work invested in order to extend the utility of the product. The AD methodology aims at two types of adaptability: design adaptability and product adaptability.

Design adaptability is the producer's ability to perform minor changes to an existing design in order to design a new product. It can be achieved by creating a family of designs or modular products such that some modules are shared by different products. Incorporating design adaptability should expedite the development of new products, and reduce manufacturing costs when the same equipment is used to produce different products.

Product adaptability is the user's ability to modify the product to satisfy new requirements. Several forms of product adaptability are considered: versatility, modularity and upgrade. The first refers to satisfaction of several functions by the same product, the second by adding or replacing modules and the third by replacing modules with newer versions as technology advances. A measure for adaptability is given based on the money saved by adapting a product rather than producing a dedicated product for each required functionality. The design process should result in a product that can be adapted to various applications that can be foreseen *a priori*. Unforeseen adaptation should be accommodated by including modularisation, adaptable interfaces and functional independence between modules.

Since AD was presented, a variety of studies were conducted to demonstrate its applicability. Some examples are: a modular vehicle concept and a versatile home and gardening tool (Hashemian, 2005), a modular kitchen appliance that can perform as a mixer, a blender or a food processor (Li et al., 2008b), a gear-cutting machine (Xu et al., 2008) and a modular coating machine (Han et al., 2012). For a further review the interested reader is referred to Gu et al. (2009).

Xue et al. (2012) and later Martinez and Xue (2016) use optimization to find the best adaptable design for a set of requirements that change through the product's life cycle. The dynamic nature of the requirements is known during the design phase and, therefore, there is no uncertainty over the fitness of each candidate design. The fitness of a candidate design is considered by its performance over the entire life-cycle. At every time phase, the best configuration and associated adjustable design variables are searched for through optimization, and the optimal performance at every time step is used to evaluate the design.

A methodology for robust design and optimization of adaptable and reconfigurable products was developed by Zhang et al. (2013, 2014, 2015) as an extension to the AD paradigm. Since this study involves robust optimization of changeable products, please refer to Section 2.4.3 for further details.

2.2.4 Design for Flexibility

Among the *ilities* mentioned in this section, flexibility is the most powerful system attribute for protection against uncertainties. It enables the system to adapt to new environmental conditions or new requirements by changing its state or configuration. Studies on flexibility can be found in a variety of fields, including finance, manufacturing systems and engineering design.

In *Options theory* in finance, projects or investments plans are considered flexible if they include contingent decisions that respond to future market conditions (see e.g., Amram and Kulatilaka, 1999; Evans, 1991; Triantis and Hodder, 1990). The concept of *real options* quantifies the financial value of flexibility. When integrated within the engineering design process, the real options theory provides some insights about the financial value of the product's flexibility, although it does not guide the designer how to introduce this flexibility into the product. The following are examples for studies of real options in the context of engineering design: de Neufville (2003) explains how real options can be incorporated into the evaluation of engineering products and projects, and provides examples for industrial projects that follow the real options reasoning. de Neufville et al. (2006) present a detailed case study of real options in a multilevel parking lot design. Ford and Sobek (2005) demonstrate the advantages of real options by analysing a successful car design activity with delayed decisions. Buurman et al. (2009) incorporate real options into the evaluation function of a robust optimization algorithm, for designing a maritime protection system. Criticism of the applicability of real options to the field of engineering design is also expressed. Saleh et al. (2009), for example, highlight the difference between measuring the value of an attribute and measuring the attribute itself. They state that real options theory has limited contribution to engineering design since it cannot quantify flexibility, and therefore it cannot be used as a design specification.

Flexibility of manufacturing systems has received a lot of attention during the last three decades. It is considered as an attribute of a manufacturing system that is capable of changing in order to cope with different types of uncertainties. Many forms of flexibility are associated with manufacturing systems. Saleh et al. (2009) highlight in their review some important forms:
Volume flexibility is the ability of a system to accommodate varying product demands by efficiently changing the production volumes.
Routing flexibility is the ability of a system to produce the same product either in a different order of operations, or by different machines. It can provide a protection against breakdowns or an effective way to accommodate a variety of demands of different products.
Expansion flexibility is the ability of the system to be expanded in order to accommodate higher demands than originally intended. It considers the maximal capacity rather than fluctuations in demand as in volume flexibility.
Product mix flexibility is the ability to produce a variety of products with minor adjustments to the system.
The interested reader is referred to the reviews of Sethi and Sethi (1990) and Saleh et al. (2009).

In engineering design, product flexibility is considered to be the product's ability to respond to changes in requirements and operating conditions, during its normal operation. In order to achieve this attribute, the product should include some properties (e.g., modularity, redundancy, design margins) that do not necessarily contribute to the product's immediate requirements. However, these properties allow the product to adapt in a cost-efficient manner to changes in requirements or operating conditions (Saleh et al., 2009). To date, there is no accepted quantitative measure for product flexibility. This kind of measure can serve as a design specification that can be weighed against other product attributes such as cost or life-span. Existing measures are surveyed in the next section.

2.2.5 Evaluation Measures for Changeable Products

In this section, existing methods for the evaluation of the quality of changeable products are surveyed. A critique of their validity is also provided. *Changeable* refers to any product that can be changed during normal operation, and it is used as a general term for *adaptable, reconfigurable* and *flexible*.

Olewnik et al. (2004) and Olewnik and Lewis (2006) suggested a framework of design for flexibility, with an iterative search procedure that includes a measure of flexibility in a multi-objective domain. The aim of the framework is to support a decision about which variables should be made flexible, and to what extent. Candidate solutions are evaluated according to a single criterion, namely the *corporate utility* (e.g. expected profit), which is a function of the expected costs, demand and price. The values of these attributes should be acquired by surveys and mathematical models, and they all depend on the design variables. Flexibility is considered to raise the performance and attractiveness of the product, but also to increase its cost. Making a variable flexible is associated with a cost, and the more flexible it is, the more it costs.

Despite the use of a single evaluation measure, Olewnik and Lewis (2006) discuss the advantages of flexible systems to satisfy multiple objectives. Generally, this statement is true, but the reasoning provided in their study implies a lack of understanding of basic concepts in multi-objective optimization. They state that *"flexible systems have the ability to eliminate performance trade-offs by adapting to give optimal performance in predictable situations"* (p. 75). This statement is supported with an example of a flexible engine that eliminates the designer's need to compromise between power output and fuel consumption. Optimality in a conflicting multi-objective domain always presents a set of trade-offs. Flexibility merely allows the user of the product to decide which trade-off solution is favourable at a given moment, depending on information that was not available earlier. For the example above, by making a flexible engine, the designer chooses a set of trade-off alternatives between the conflicting objectives, and the customer is able to choose the one most suitable for his/her needs. The notion of optimality in a multi-objective domain is conceived by Olewnik and Lewis as *"the extreme points of the Pareto frontier, since they represent the optimal performance for the individual objective functions"* (p. 82). Of course this observation is not true, especially in cases when towards the extreme points, a slight improvement in one objective results in a drastic degradation in the others. Please refer to Section 2.3.3 for a basic introduction to multi-objective optimization.

The interpretation of optimality described above has led Olewnik and Lewis to the following measure of flexibility, q_f, which is based on the Euclidean distances between the

m extreme points of the Pareto front of an m-dimensional multi-objective optimization problem:

$$q_f = \sum_{i=1,j=i+1}^{m} \|\Delta \mathbf{f}_{i,j}\|_2 \tag{2.4}$$

where $\Delta \mathbf{f}_{i,j}$ is the vector in objective space between two consecutive extreme points i and j. For the aforementioned reasons, and other drawbacks rooted in the manner in which the extreme points are ordered, the measure suggested in Equation 2.4 is not sufficient for quantifying the added value of flexibility to the product's performance. Although the flexibility measure, q_f, is formulated, Olewnik and Lewis have failed to demonstrate how it can be used to support a decision, even with the toy example presented.

Denhart (2013) addressed the question of how to evaluate and compare reconfigurable systems in a multi-objective domain.[1] This question plays a significant role in this thesis as well, and it will be studied extensively in Chapter 4. Denhart used an exploration rover design with two possible configurations as a case study in order to suggest an answer to the question above. Unfortunately, the suggested problem formulation avoided the question, and actually posed a single-objective problem as a multi-objective one. The concern in this case study was the rover's manoeuvrability in rough terrain. A combination of three performance measures was used to quantify manoeuvrability. Some uncertain parameters were considered, represented by a discrete set of scenarios of different combinations. Every configuration of a candidate solution was simulated for all scenarios in order to assess the solution's performance.

The problem was posed as multi-objective by treating manoeuvrability in different operating conditions as different objectives (i.e. uphill, downhill or levelled). Since there is no conflict between the objectives (the rover cannot move uphill and downhill simultaneously), the best configuration for each scenario was chosen to represent the rover's performance. As a result, the set of performances could be represented by its ideal vector, the vector consisting of best values in each objective among the vectors in the set. When comparing between candidate solutions, the ideal vectors were used to determine dominance.

As stated earlier, the multi-objective problem was not formulated correctly. The real three objectives are the performance measures that were used to measure manoeuvrability: average speed, distance from intended path and the ratio between distance to obstacle and turning diameter. The original three objectives are in fact a part of the

[1]See Section 2.3.3 for information on multi-objective optimization

uncertain environmental conditions. When the reconfigurable solution is evaluated in the 'real' multi-objective domain, a trade-off exists between the objectives, and a single configuration could be superior in one objective but inferior in another.

Although Denhart did not provide a sufficient answer to this research question, some aspects of the work have significance for this study:

1. The necessity of an evaluation method for changeable systems in a multi-objective domain is highlighted.

2. It is noted that a changeable system can adapt to new environmental conditions, which are uncertain during the design phase, and perform in the configuration that yields the best performance. This ability allows the designer to evaluate the system according to its best operating mode for each scenario of the uncertain conditions.

A measure of adaptability was suggested by Gu et al. (2004) for the Adaptable Design methodology. It is based on the cost of a reconfigurable product that can serve several requirements, compared to the total costs of producing a dedicated product for each requirement. The cost of a reconfigurable product considers the following factors: initial production cost to the original requirement(s), probability of reconfiguration to each state, and the cost of reconfiguration to each state. Assuming a reconfigurable product has n states and the i^{th} state is denoted as S_i, the adaptability measure is formulated as follows:

$$A = \sum_{i=1}^{n} \Pr(R_i)\left(1 - \frac{COST(S_1 \rightarrow AS_i)}{COST(0 \rightarrow IS_i)}\right) \tag{2.5}$$

where $\Pr(R_i)$ is the probability of requirement, i, to occur, IS_i is the ideal state, i, if the product was designed to satisfy R_i alone, and AS_i is the actual state, i, achieved by reconfiguration. The arrows in Equation 2.5 denote reconfiguration of the product from one state to another. If the numerator is larger than the denominator, it implies that it is more expensive to adapt to a state than to produce a dedicated product to satisfy this requirement, and therefore adaptability is not advocated.

Fletcher et al. (2010) proposed a different quantification measure for the Adaptable Design methodology. Here, reconfigurable products are assessed based on their architecture and interconnectivity between components. The reasoning behind this measure is that modular and segregated products can be adjusted to different requirements more easily. The product is broken down into its functional units. Each unit is assigned with a weight according to its cost, and the interactions and interfaces between the units are evaluated. The complexity of the product architecture is quantified by multiplying each connection value with the cost of the connected components. Finally, the relative adaptability (RA) is defined as the ratio of the ideal architecture (fully segregated) and the actual architecture. Equation 2.6 defines the relative adaptability measure:

$$RA = \frac{\sum_{\text{segregated connections}} \sigma_{ij}(C_i + C_j)}{\sum_{\text{all connections}} \sigma_{ij}(C_i + C_j)} \tag{2.6}$$

where C_i is the cost of unit i and σ_{ij} is a measure for the complexity of the connection between component i and component j. It is worth mentioning that products are not evaluated according to their performance when using this measure. Instead, they are evaluated according to their architecture alone.

It can be concluded that a relatively small amount of studies exist in the literature on evaluation measures for changeable products. Some measures are based on the product's performance (Olewnik et al., 2004; Olewnik and Lewis, 2006; Denhart, 2013), while others are based on the product's architecture (Fletcher et al., 2010) or cost (Gu et al., 2004). Denhart (2013) is the only one to address the impact of changeability on product performance in an uncertain environment. None of the evaluation measures above was used within an optimization framework to search for high-quality changeable products. Two research gaps are identified from this survey:

1. A method to evaluate changeable products in both single-objective and multi-objective domains, to support a comparison between alternative designs. This method should include an evaluation measure for changeable products that considers various types of uncertainties.

2. A design methodology for robust changeable products that includes an optimization procedure.

2.3 Optimization

> *"Since the fabric of the universe is most perfect and the work of a most wise creator, nothing at all takes place in the universe in which some rule of maximum or minimum does not appear."*
>
> — Leonhard Euler (1707-1783)

Optimization, also known as Mathematical Programming, is the process of seeking and selecting the best alternative from a set of possibilities, with respect to a certain (or several) objective(s). Optimization problems may arise in many different fields such as economics, engineering, mathematics, computer science, logistics, physics and control. Without loss of generality, a minimisation problem is mathematically defined as follows:

$$\begin{aligned}
\text{minimise:} \quad & \psi(\mathbf{x})\,, & & \qquad (2.7a)\\
\text{subject to:} \quad & h_j(\mathbf{x}) = 0, & j = \{1, \ldots, n_j\}\,, & \qquad (2.7b)\\
& g_k(\mathbf{x}) \leq 0, & k = \{1, \ldots, n_k\}\,, & \qquad (2.7c)\\
& x_{i,l} \leq x_i \leq x_{i,u}, & i = \{1, \ldots, n\}\,. & \qquad (2.7d)
\end{aligned}$$

A solution $\mathbf{x} \in \Omega$ is a vector of n decision variables: $\mathbf{x} = [x_1, \ldots, x_n]$, where Ω is the design space, typically consisting of real or binary values. Each decision variable x_i is subject to a lower bound $x_{i,l}$, and an upper bound $x_{i,u}$. The objective function is $\psi : \Omega \to \mathbb{R}$, and n_j and n_k are the number of equality and inequality constraints, respectively.

For the sake of clarity, the feasible domain is denoted as $\mathcal{X} \subseteq \Omega$, where a solution $\mathbf{x} \in \mathcal{X}$ is considered as a solution that satisfies Equations (2.7b–2.7d). Following this notation, Equation 2.7 can be written as follows:

$$\min_{\mathbf{x} \in \mathcal{X}} \psi(\mathbf{x})\,. \qquad (2.8)$$

This study combines several topics within the wide research field of optimization. The basic background for understanding each topic is provided below, together with references to relevant literature for a deeper understanding. In order to address these subjects, and to understand their differences and similarities, a structured notation is presented first.

2.3.1 Nomenclature Explained

The nomenclature used within this thesis is presented in the preface. To better understand the differences between the type of variables, and the manner they are used to describe different classes of optimization problems, the following explanations are provided.

Grouped Variables

A scalar value is marked with a normal weight font, while a vector, consisting of several scalar values is marked with a bold font and/or its elements within square brackets (e.g., $\mathbf{a} = [a_1, a_2, a_3]$).

A set of elements of the same type is denoted with an underline and/or its elements within curly brackets. For example: $\underline{\mathbf{b}} = \left\{\mathbf{a} \in \mathbb{R}^2 \mid a_1 = a_2\right\}$ is an infinite set of vectors, and the set $\underline{b} = \{b_1, b_2\}$ is a discrete set consisting of two scalar values.

Random vs. Deterministic Variables

A deterministic variable (either a scalar, a vector or a set) is denoted with a lower-case letter. When the same variable is subject to uncertainties, it is treated as a random variate. The corresponding upper-case letter is used to denote it. For example, consider the function $f(\mathbf{a})$, where $f : \mathbb{R}^2 \to \mathbb{R}$. If the values of the elements in $\mathbf{a}$ are uncertain, then it is denoted as $\mathbf{A}$. The function value is also uncertain, and therefore it is also assigned with an upper-case letter: $F(\mathbf{A})$.

Often, a random variate is repeatedly sampled and represented by a set of sampled deterministic values. This kind of set is denoted here differently than other sets, with a bar over a capital letter. If the random variate $\mathbf{A}$ is sampled k times, then $\bar{\mathbf{A}} = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$. The function variate $F(\mathbf{A})$ is also represented by a sampled set, i.e., $\bar{F}\left(\bar{\mathbf{A}}\right) = \{F(\mathbf{a}_1), \ldots, F(\mathbf{a}_k)\}$.

Types of Variables

Since this study deals with changeable products, a distinction is made between variables that must be fixed during the design stage, and others that can be adjusted by the user. Another distinction is made between design variables and other parameters that affect the objective functions and cannot be controlled. The following notation is used:

- The vector $\mathbf{x} = [x_1, \ldots, x_{n_x}] \in \mathcal{X}$ represents an adaptive design, where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is the feasible domain. The variables in $\mathbf{x}$ include all the properties that define the design, and cannot be intentionally altered once the product goes into service.

- The vector $\mathbf{y} = \left[y_1(\mathbf{x}), \ldots, y_{n_y}(\mathbf{x})\right] \in \mathcal{Y}(\mathbf{x})$ represents a possible configuration of the design $\mathbf{x}$. It includes all the properties that can be changed during the product's service. $\mathcal{Y}(\mathbf{x}) \subseteq \mathbb{R}^{n_y(\mathbf{x})}$ is the domain of adjustable variables of the design $\mathbf{x}$, and it includes all possible configurations of the design. It is referred to as the design's *adaptability*.

- The vector $\mathbf{p} = \left[p_1, \ldots, p_{n_p}\right]$ includes the environmental parameters, which are independent from the design variables, and cannot be controlled. Parameters are explicitly considered in this study when uncertainties over their values are present.

Types of Objective Functions

Different notations are used for three types of objective functions:

- $\psi(\mathbf{x})$ – a deterministic objective that does not depend on uncertain variables and cannot be affected by adaptation.

- $\phi(\mathbf{x}, \mathbf{p})$ – a stochastic objective that depends on uncertain variables and cannot be affected by adaptation. $\Phi(\mathbf{x}, \mathbf{P})$ is the variate of ϕ that corresponds to the variate $\mathbf{P}$.

- $\gamma(\mathbf{x}, \mathbf{y}, \mathbf{p})$ – a stochastic objective that can be changed by adaptation. Objectives of this type are inherently affected by uncertainties. Even if a changeable objective is not directly influenced by the uncertainties, i.e. $\gamma(\mathbf{x}, \mathbf{y})$, the configuration $\mathbf{y}$ varies according to the realisation of the uncertain parameters (to optimize other objectives), and therefore the value of this objective is affected as well. $\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{y}, \mathbf{P})$ is the variate of $\boldsymbol{\gamma}$ that corresponds to the variate $\mathbf{P}$.

2.3.2 Common Optimization Methods

Optimization problems can be tackled in many ways. Wolpert and Macready (1997) have shown in their seminal *no free lunch theorems* that any two algorithms are identically efficient when averaged over all classes of optimization problems. This means that a single optimization method cannot be suitable for every problem, and the algorithm needs to be tailored to the specific problem class.

Calculus-based iterative methods for local optimization such as *gradient methods, Newton methods* or *conjugate methods* were already studied back in the 18^{th} century. These methods are very useful when the objective function can be analytically derived, and derivatives information can be used (Gill et al., 1981).

Linear programming and the *Simplex* algorithm were proposed by Dantzig in 1947 to solve optimization problems that can be formulated as a set of linear inequalities and equations (Dantzig and Thapa, 1997). With the increasing availability of computers, optimization algorithms could be developed for solving more difficult problems such as stochastic, discrete, non-convex and non-linear problems. For example, *branch and bound* (Lawler and Wood, 1966) is used for discrete problems, *cutting plane* was introduced by Gomory (1958) for solving mixed integer linear problems and can also be used for non-linear programming (Avriel, 2003, pp. 477-482), and *dynamic programming* (Bellman, 1957), which is a recursive method that is used as a basis for many optimization algorithms.

The methods mentioned above are very useful for solving problems that adhere to a specific structure (e.g., linear/convex/combinatorial). However, many optimization problems are formulated in a general form and include a combination of challenges such as multi-modality, discontinuity and non-convexity. *Population-based heuristics* are commonly used for global optimization of difficult problems of this sort. These methods use a population of agents to simultaneously explore different areas of the domain. In order to focus the search in promising areas, successful individuals attract the other agents towards their area. Randomness is introduced into the process to avoid convergence to local optima, and therefore two consecutive runs of the same algorithm will not necessarily produce the same results. Among these methods are *genetic algorithms* (Holland, 1975; Goldberg, 1989), *differential evolution* (Storn and Price, 1997), *particle swarm optimization* (Kennedy and Eberhart, 1995) and *ant colony optimization* (Dorigo et al., 1996; Dorigo and Blum, 2005). Please refer to Giagkiozis et al. (2013b) for a survey of population-based optimization methods. The survey includes a very useful introduction to each of these approaches.

2.3.3 Multi-objective Optimization

Many optimization problems can be classified as *multi-objective optimization problems*, and involve the simultaneous optimization of two or more objectives. An MOP is formulated similarly to the single-objective problem in Equation (2.7), with the slight distinction that the objective $\psi(\mathbf{x})$ is replaced with a vector of m objective functions $\boldsymbol{\psi}(\mathbf{x}) = [\psi_1(\mathbf{x}), \ldots, \psi_m(\mathbf{x})]$. Objectives in real-world MOPs are often in conflict, i.e. an improvement of one objective results in a degradation of another. When this is the case, there is no single solution that minimises all objectives. Therefore, with the absence of known preferences between the objectives, the solution to an MOP is usually a set of solutions that provide different trade-offs between the various objectives.

Dominance is a fundamental concept in multi-objective optimization (MOO), which commonly defines the notion of optimality. This type of optimality is known as *Pareto optimality.* Since these terms are regularly used within this thesis, their formal definitions are given below:

Definition 2.1 (Pareto Dominance)**.** *A vector* $\mathbf{a} = [a_1, \ldots, a_n]$ *is said to Pareto dominate another vector* $\mathbf{b} = [b_1, \ldots, b_n]$ *(denoted as* $\mathbf{a} \prec \mathbf{b}$*) if and only if* $\forall i \in \{1, \ldots, n\} : a_i \leq b_i$ *and* $\exists i \in \{1, \ldots, n\} : a_i < b_i$

Definition 2.2 (Pareto Optimality)**.** *A solution* $\mathbf{x}^\star \in \mathcal{X}$ *is said to be Pareto-optimal in* $\mathcal{X}$ *if and only if* $\neg\exists\hat{\mathbf{x}} \in \mathcal{X} : \boldsymbol{\psi}(\hat{\mathbf{x}}) \prec \boldsymbol{\psi}(\mathbf{x})$

Definition 2.3 (Pareto-Optimal Set)**.** *The Pareto-optimal set* $\underline{\mathbf{x}}^\star$ *is the set of all Pareto-optimal solutions:* $\underline{\mathbf{x}}^\star = \{\mathbf{x} \in \mathcal{X} \mid \neg\exists\hat{\mathbf{x}} \in \mathcal{X} : \boldsymbol{\psi}(\hat{\mathbf{x}}) \prec \boldsymbol{\psi}(\mathbf{x})\}$

Definition 2.4 (Pareto-Optimal Front)**.** *The Pareto-optimal front (PF) is the set of objective vectors corresponding to the solutions in the Pareto-optimal set, i.e., PF* $\equiv \underline{\boldsymbol{\psi}}(\underline{\mathbf{x}}^\star)$

The global solution of an MOP, the Pareto-optimal set, may contain an infinite number of trade-off solutions. A multi-objective optimizer should provide the decision-maker (DM) with a finite set of solutions, known in the literature as an *approximation set* (AS), which is a representation of the true Pareto-optimal set. The objective vectors corresponding to the solutions of the AS are referred to as the *approximated front* (AF). According to Purshouse (2003), the AS and its associated AF should fulfil four requirements:

Proximity. The AF should be as close as possible to the true PF.

Pertinence. The AF should only contain vectors within the DM's region of interest (ROI), which is usually a subspace of the entire objective space.

Extent. The AF should be stretched across the entire range of the PF, within the ROI.

Distribution. The objective vectors of the AF should be evenly distributed along the trade-off surface.

The ideal AF to a bi-objective optimization problem is depicted in Figure 2.3. It can be seen that all of the objective vectors are evenly distributed on the true Pareto front, over its full extent within the ROI.

Setting preferences between conflicting objectives is an essential task within an MOO procedure. Ultimately, it is the role of a DM to determine which of the Pareto-optimal solutions will be the outcome of the optimization procedure. A common classification of MOO approaches can be made according to the stage in which DM preferences are introduced into the search (Zitzler, 1999; Purshouse, 2003):

***A priori* decision-making:** The objectives are aggregated to form a single-objective function, whose optimum is the preferred optimal solution. By setting *a priori* preferences, an MOP is reformulated as a single-objective optimization problem (SOP) that can be solved by a wide variety of algorithms. However, it requires a profound knowledge about the trade-offs between the objectives.

Decision-making during search: The DM is interactively involved in the search procedure. Preferences can be incorporated into the search to focus it towards

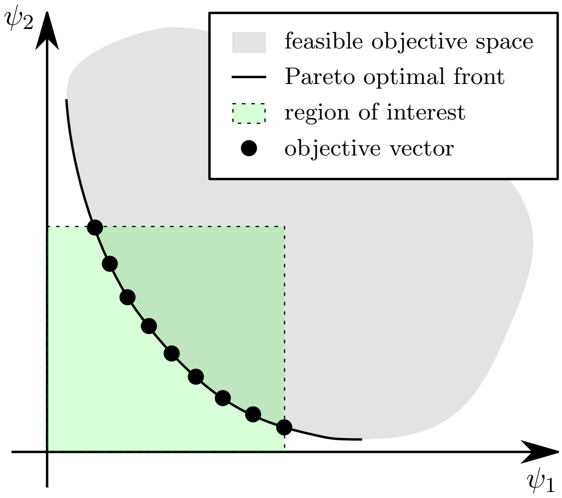


Figure 2.3: The ideal approximated front.

interesting regions, as new information becomes available. This approach does not require initial knowledge, but it does require effective visualisation tools to allow for efficient involvement of the DM.

***A posteriori* decision-making:** The optimizer returns an approximated set of the PF, and the DM chooses one preferred solution from the set. The main disadvantage of this approach is the potential waste of resources on finding solutions that are not in the DM's ROI.

Most of the non-population-based methods for MOO are based on the first approach, i.e., "scalarisation" of the MOP. Once the problem is formulated as an SOP, classic optimization methods can be used in order to search for the optimal solution. If more than one Pareto-optimal solution is sought, multiple SOPs are formulated by using different combinations of the objectives. Please refer to Steuer (1986) and Jahn (1986) for surveys of methods of this type for linear MOPs and to Miettinen (1999) and Marler and Arora (2004) for non-linear MOPs.

Nowadays, evolutionary multi-objective optimization algorithms (EMOAs) are the most popular approach for solving MOPs. Ten years after the first genetic algorithm was presented by Holland (1975), Schaffer (1985) introduced the first *vector evaluated genetic algorithm* (VEGA) to optimize multiple objectives in a single run. It was found that evolving a population of solutions simultaneously is highly suitable to MOO, where a set of optimal solutions is desired. During the three decades since the introduction of VEGA, the field of evolutionary multi-objective optimization has been constantly growing.

The large amount of studies on EMOAs is summarised in a number of review papers.

The following are recommended for further reading:
Fonseca and Fleming (1995) provided the first overview on EMOAs, where they highlighted the superiority of EAs that incorporate dominance relations and niching in their selection mechanism. Based on a conceptual algorithm proposed by Goldberg (1989), the first algorithms of this class were MOGA (Fonseca and Fleming, 1993), NPGA (Horn et al., 1994) and NSGA (Srinivas and Deb, 1994).

Coello (1999) conducted a comprehensive survey of evolutionary multi-objective optimization techniques. Every method reviewed in the survey is described in detail, followed by an extensive list of applications and a discussion on its strengths and weaknesses. More recent surveys include the work of Zhou et al. (2011), that provides a thorough investigation of the latest developments in the field as well as an extensive list of applications. Some of the topics covered by this survey are decomposition-based EMOAs, memetic EMOAs, co-evolutionary EMOAs, multi-modal MOPs and many-objective problems. Giagkiozis et al. (2013b) surveyed the differences and commonalities among various population-based optimization methods used for MOO. In addition to references to the relevant literature, Giagkiozis et al. explain the principles of each method. Therefore, this work can be very useful for researchers and practitioners who wish to solve MOPs, but are not necessarily experts in the field. The strengths and weaknesses of the methods are compared against each other, to support a proper choice of heuristic according to the type of MOP.

In addition to EAs, other population-based methods were adapted to solve MOPs. These methods include *evolutionary strategies* (Knowles and Corne, 2000), *particle swarm optimization* (Reyes-Sierra and Coello, 2006), *ant colony optimization* (Doerner et al., 2004), *differential evolution* (Das and Suganthan, 2011) and *artificial immune systems* (Coello and Cortes, 2005).

2.3.4 Evaluation Measures for Sets

One of the goals of this study is to answer the question *"how to evaluate changeable products in a multi-objective setting?"*. A changeable product is associated with a set of performance vectors, as it can be adjusted by its user to satisfy different preferences. Some methods for the assessment of sets were developed in the fields of *EMOA evaluation* and *set-based optimization.* These methods can be adopted and implemented to evaluate changeable products. In this section, popular quality measures for evaluation of sets and their underlying principles are surveyed, followed by an overview of set-based optimization.

Table 2.3: Dominance relations between two point vectors **a** and **b**, and between two sets of vectors $\underline{\mathbf{a}}$ and $\underline{\mathbf{b}}$ (adapted from Zitzler et al., 2003).

Relation	Vectors		Sets	
Strictly dominates	$\mathbf{a} \prec\prec \mathbf{b}$	**a** is better than **b** in all objectives.	$\underline{\mathbf{a}} \prec\prec \underline{\mathbf{b}}$	Every $\mathbf{b} \in \underline{\mathbf{b}}$ is strictly dominated by at least one $\mathbf{a} \in \underline{\mathbf{a}}$.
Dominates	$\mathbf{a} \prec \mathbf{b}$	**a** is not worse than **b** in all objectives and better in at least one objective.	$\underline{\mathbf{a}} \prec \underline{\mathbf{b}}$	Every $\mathbf{b} \in \underline{\mathbf{b}}$ is dominated by at least one $\mathbf{a} \in \underline{\mathbf{a}}$.
Better			$\underline{\mathbf{a}} \lhd \underline{\mathbf{b}}$	Every $\mathbf{b} \in \underline{\mathbf{b}}$ is weakly dominated by at least one $\mathbf{a} \in \underline{\mathbf{a}}$ and $\underline{\mathbf{a}} \neq \underline{\mathbf{b}}$.
Weakly dominates	$\mathbf{a} \preceq \mathbf{b}$	**a** is not worse than **b** in all objectives.	$\underline{\mathbf{a}} \preceq \underline{\mathbf{b}}$	Every $\mathbf{b} \in \underline{\mathbf{b}}$ is weakly dominated by at least one $\mathbf{a} \in \underline{\mathbf{a}}$.
Non-dominated/ Incomparable	$\mathbf{a} \parallel \mathbf{b}$	Neither **a** weakly dominates **b** nor **b** weakly dominates **a**.	$\underline{\mathbf{a}} \parallel \underline{\mathbf{b}}$	Neither $\underline{\mathbf{a}}$ weakly dominates $\underline{\mathbf{b}}$ nor $\underline{\mathbf{b}}$ weakly dominates $\underline{\mathbf{a}}$.

Quality Indicators for Approximation Sets

Over the last two decades evolutionary-based approaches for multi-objective optimization have gained increasing popularity, leading to a variety of newly-developed EMOAs. As a result, assessment methods were required to compare alternative EMOAs and decide which algorithm is the most suitable for a given application (Fonseca and Fleming, 1996; Zitzler, 1999; Van Veldhuizen, 1999; Knowles and Corne, 2002; Zitzler et al., 2003, 2010). This type of comparison is not a trivial task, since the result of an EMOA is usually a set of non-dominated solutions, rather than a single scalar value. Several quality indicators to compare and evaluate non-dominated sets were developed. These indicators are not only used for the assessment of algorithms, but also as a selection mechanism in *indicator-based* EMOAs (e.g., Zitzler and Künzli, 2004; Emmerich et al., 2005).

The common quality indicators can be classified into two main categories:
a *unary quality indicator* $q[\underline{\mathbf{a}}]$ is a function that assigns a scalar value to a set of vectors $\underline{\mathbf{a}} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$; and
a *binary quality indicator* $q[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ is a function that assigns a scalar value to an ordered pair of sets $\underline{\mathbf{a}}$ and $\underline{\mathbf{b}}$. Some of the binary indicators are symmetric, i.e. $q[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = c - q[\underline{\mathbf{b}}, \underline{\mathbf{a}}]$, where c is a constant (Knowles and Corne, 2002). Although symmetric indicators are easier to use, as only one comparison has to be conducted for each pair of sets, Zitzler et al. (2003) have shown that they provide less information regarding the relations between the sets.

A methodology to evaluate quality indicators was presented by Zitzler et al. (2003). In their study they extended the concept of vector domination to domination between sets of vectors, as summarised in Table 2.3. Based on these relations, Zitzler et al. examined each quality indicator to check whether it can be used to indicate each of the dominance relations. Both unary indicators and binary indicators were considered, as well as combinations of several indicators. They also provided proofs for the following statements:

- Unary quality indicators cannot indicate whether an approximated set is *better* than another.
- Some unary indicators are able to determine whether an approximated set is not worse than another.
- Binary indicators are able to determine that an approximated set is *better* than another.

Note that this study was confined to dominance relations, and did not considered other qualities of approximated sets such as diversity and pertinence.

In the following list, some common quality indicators are presented. Unless otherwise specified, all indicators consider the properties of the sets in the objective space. The list is ordered according to the indicators class (unary/binary) and the quality they measure.

Unary Indicators

Diversity (extent and distribution):

- A very simple indicator suggested by Schott (1995) is the number of members in the AS (i.e., number of non-dominated solutions found).

Distribution:

- Also suggested by Schott (1995), the *spacing* indicator is defined as follows:

$$q_S[\underline{\mathbf{a}}] = \left(\frac{1}{n-1} \sum_{i=1}^{n} \left(d_i - \tilde{d} \right)^2 \right)^{1/2}, \tag{2.9}$$

where d_i is the minimum *Manhattan* distance[1] of the i^{th} objective vector from other vectors in $\underline{\mathbf{a}}$, and $\tilde{d}$ is the average of all d_i values. The indicator only considers

[1] Note that all distance-based indicators require a proper normalisation of the objectives to produce meaningful values.

the distribution of the set (and not its extent), where a value of $q_S[\underline{\mathbf{a}}] = 0$ indicates that the vectors are evenly distributed.

Combined proximity and diversity:

- The *hypervolume* indicator, $q_{HV}[\underline{\mathbf{a}}]$, proposed by Zitzler and Thiele (1998) measures the hypervolume (HV) of the union of all objective vectors which are dominated by the set, $\underline{\mathbf{a}}$, and dominate a reference point, $\mathbf{r}$. The HV is a very popular indicator, nevertheless it suffers from two major drawbacks: **(a)** it is sensitive to the selection of $\mathbf{r}$, as demonstrated by Knowles and Corne (2002), and **(b)** it requires high computational effort and it suffers from the *curse of dimensionality.* Since the values of different objectives might vary radically, the HV values are often normalised either by the HV obtained by the true PF (Van Veldhuizen, 1999) or by a hyperbox confined between the best and worst known values for each objective (Zitzler, 1999). For an extensive overview on the HV indicator and its applications in the field of evolutionary multi-objective optimization (EMO), see Bradstreet (2011).

Unary Indicators Using a Reference Set

Many quality indicators that are considered in the literature as unary are, in fact, binary indicators, as they require two sets in order to produce a value. These indicators compute the quality measure of a set $\underline{\mathbf{a}}$, compared to a reference set. The most common reference set is the Pareto set (or Pareto front), but other sets can be used such as the set of all known non-dominated solutions (found by various algorithms). All of these indicators can be used as binary quality indicators as well. Considering an AS, $\underline{\mathbf{a}}$, and a reference set, $\underline{\mathbf{r}}$:

Proximity:

- The *error ratio* suggested by Van Veldhuizen (1999) measures the ratio of solutions in $\underline{\mathbf{a}}$ that are not members of $\underline{\mathbf{r}}$:

$$q_{ER}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = \frac{|\{\mathbf{a} \in \underline{\mathbf{a}} \,|\, \mathbf{a} \notin \underline{\mathbf{r}}\}|}{|\underline{\mathbf{a}}|}. \tag{2.10}$$

- The *generational distance* suggested by Van Veldhuizen (1999) is defined as follows:

$$q_{GD}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = \frac{1}{n}\left(\sum_{i=1}^{n} {d_i}^p\right)^{1/p}, \tag{2.11}$$

where d_i is the Euclidean distance in objective space of the i^{th} vector in $\underline{\mathbf{a}}$ from the nearest vector in $\underline{\mathbf{r}}$. A value of $q_{GD}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = 0$, with $\underline{\mathbf{r}}$ being a representative set of vectors from the true PF, indicates that $\underline{\mathbf{a}}$ is a subset of $\underline{\mathbf{r}}$, and therefore the algorithm has converged to the true PF.
Schütze et al. (2012) indicated that q_{GD} produces better values for larger approximated sets. For example, if a set $\underline{\mathbf{a}}$ consists of a single vector $\mathbf{a}$ with a distance $d = 1$ from the true PF, $\underline{\mathbf{r}}$, then $q_{GD}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = 1$. In case the set $\underline{\mathbf{a}}$ consists of n replicas of $\mathbf{a}$, the value of $q_{GD}[\underline{\mathbf{a}}, \underline{\mathbf{r}}]$ would be $\sqrt[p]{n}/n$. Therefore Schütze et al. suggested the *averaged generational distance* that is indifferent to the cardinality of the AS:

$$q_{GD_p}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = \left(\frac{1}{n} \sum_{i=1}^{n} {d_i}^p \right)^{1/p} . \tag{2.12}$$

- Zitzler et al. (2003) suggested the indicators $q_{\varepsilon}[\underline{\mathbf{a}}, \underline{\mathbf{r}}]$ and $q_{\varepsilon+}[\underline{\mathbf{a}}, \underline{\mathbf{r}}]$ that indicate how much all members of $\underline{\mathbf{r}}$ need to be scaled or translated, respectively, in order that $\underline{\mathbf{a}}$ would weakly dominate $\underline{\mathbf{r}}$.

Diversity:

- The *spread* indicator suggested by Deb et al. (2002) is defined as follows:

$$q_{\Delta}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = \frac{\sum_{j=1}^{m} d_j^e + \sum_{i=1}^{n} |d_i - \tilde{d}|}{\sum_{j=1}^{m} d_j^e + n\tilde{d}}, \tag{2.13}$$

where m is the number of objectives, d_i is the minimum Euclidean distance of the i^{th} objective vector from other vectors in $\underline{\mathbf{a}}$, $\tilde{d}$ is the average of all d_i values, and d_j^e is the minimal Euclidean distance of the best solution in $\underline{\mathbf{r}}$ w.r.t. the j^{th} objective from the solutions in $\underline{\mathbf{a}}$.

Combined proximity and diversity:

- The *inverted generational distance* suggested by Coello and Cortes (2005) is defined as follows:

$$q_{IGD}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = q_{GD}[\underline{\mathbf{r}}, \underline{\mathbf{a}}] . \tag{2.14}$$

Since the distances are measured from all vectors of $\underline{\mathbf{r}}$, regions of the PF not covered by $\underline{\mathbf{a}}$ result in an increased q_{IGD} value.
Schott (1995) and Czyzzak and Jaszkiewicz (1998) have suggested similar metrics.

For the same reasons as for the q_{GD_p} indicator, Schütze et al. (2012) proposed the *averaged inverted generational distance*:

$$q_{IGD_p}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = q_{GD_p}[\underline{\mathbf{r}}, \underline{\mathbf{a}}]\,. \tag{2.15}$$

- The *averaged Hausdorff distance* indicator suggested by Schütze et al. (2012) is defined as follows:

$$q_{\Delta_p}[\underline{\mathbf{a}}, \underline{\mathbf{r}}] = \max\left\{q_{GD_p}[\underline{\mathbf{a}}, \underline{\mathbf{r}}]\,, q_{IGD_p}[\underline{\mathbf{a}}, \underline{\mathbf{r}}]\right\}. \tag{2.16}$$

According to Schütze et al. (2012), this indicator serves as a more reliable metric than its components q_{GD_p} and q_{IGD_p}.

Binary Indicators

The indicators listed below are specifically designed to compare sets that do not dominate each other.

Proximity:

- Zitzler and Thiele (1998) proposed the *coverage* metric that measures the percentage of solutions in $\underline{\mathbf{b}}$ that are dominated by solutions in $\underline{\mathbf{a}}$:

$$q_C[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = \frac{|\{\mathbf{b} \in \underline{\mathbf{b}} \,|\, \exists \mathbf{a} \in \underline{\mathbf{a}} : \mathbf{a} \preceq \mathbf{b}\}|}{|\underline{\mathbf{b}}|}. \tag{2.17}$$

The q_C indicator does not provide information as to "how much" solutions in one set are dominated by the solutions of the other set. Therefore the relation $q_C[\underline{\mathbf{a}}, \underline{\mathbf{b}}] > q_C[\underline{\mathbf{b}}, \underline{\mathbf{a}}]$ does not necessarily imply that $\underline{\mathbf{a}}$ is better than $\underline{\mathbf{b}}$.

Combined proximity and diversity:

- Zitzler (1999) had used the hypervolume measure to suggest a binary indicator that measures the hypervolume covered by one set, but not covered by the other:

$$q_D[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = q_{HV}[\underline{\mathbf{a}} \cup \underline{\mathbf{b}}] - q_{HV}[\underline{\mathbf{b}}]\,. \tag{2.18}$$

- Zitzler et al. (2003) presented the binary q_{ϵ_+} indicator according to the concept of ϵ_+ *dominance* (Laumanns et al., 2002). A vector $\mathbf{a}$ is said to ϵ_+ dominate another vector $\mathbf{b}$, denoted as $\mathbf{a} \preceq_{\epsilon+} \mathbf{b}$, iff $\mathbf{a} \preceq \mathbf{b} + \epsilon$, where ϵ is a real number. The value of ϵ defines the dominance relation; a positive value allows a vector to ϵ_+

dominate another non-dominated vector, while a negative value requires stronger domination than the common definition.

For two sets of vectors $\underline{\mathbf{a}}, \underline{\mathbf{b}} \in \mathbb{R}^n$, the binary measure $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ is defined as the smallest value of ϵ required for every vector $\mathbf{b} \in \underline{\mathbf{b}}$ to be ϵ_+ dominated by at least one vector $\mathbf{a} \in \underline{\mathbf{a}}$. The symmetric indicator $q_{\epsilon_+}[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ is composed of the difference between $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ and $\epsilon_+[\underline{\mathbf{b}}, \underline{\mathbf{a}}]$. A formal definition of the q_{ϵ_+} indicator and a detailed example showing how it can be calculated appear in Appendix A.

Set Evaluation in Concept-based Optimization

Exploring alternative concepts and properly selecting the most suitable one has a great impact on the success of an engineering design process. The significance of this problem has been reflected in an increasing effort to develop methodologies to support concept selection (for a review, see Okudan and Tauhid, 2008). Among the various approaches, some studies can be found on *concept-based optimization*. In concept-based optimization, the design space consists of different concepts, where each concept is associated with a set of similar designs. Each design is mapped to a different objective vector, and therefore every concept is mapped to a set of alternative objective vectors. When comparing concepts, the comparison is made between sets of solutions.

Mattson and Messac (2003) proposed the *s-Pareto frontier* as a tool for concept selection. In this framework, the PF of each concept is identified separately, and the s-Pareto frontier is the global optimum, consisting of the non-dominated solutions among all concepts. Once the s-Pareto frontier is identified, the DM should choose one of the concepts according to additional unmodelled knowledge such as a preferred ROI within either the design or the objective space. Mattson and Messac (2005) suggested that a concept with a large surface area on the s-Pareto frontier is preferred, since it potentially offers more *design flexibility* for detailed design. They also considered a robust s-Pareto frontier to incorporate uncertainties, by shifting the expected objective values by $k\boldsymbol{\sigma}$, where k is a scalar and $\boldsymbol{\sigma}$ represents the standard deviations of the marginal objective distributions.

Lewis et al. (2010) suggested a conceptual design approach for modular products that involves an MOO procedure. They aimed at products that can be upgraded from one concept to another, such that every concept allows for a different trade-off between performance and cost. Their methodology includes optimization of several concepts to identify Pareto-optimal solutions for each concept. Then a modular product is designed by identifying parts that can be used in all concepts. Lewis et al. (2010) addressed the loss of optimality caused by constraining the different concepts to use identical parts.

The approach was validated by Wood et al. (2012) who constructed physical models based on the optimization results.

Avigad and Moshaiov (2009a) proposed an interactive evolutionary algorithm (EA) for set-based concept optimization. In this approach, candidate solutions that belong to different concepts are evaluated in a common objective space, while every concept is associated with a different design space. The DM assigns a preference value for each concept, with the ability to change the preferences as the search progresses. The resulting solution's fitness is determined according to its objective vector and the DM's preferences.

Although the three studies above suggested approaches to support concept selection, none of them used an evaluation measure to assign a grade to each concept. The only two studies found in the literature to do so are the following:

Avigad and Moshaiov (2009b) addressed the drawbacks of the s-Pareto approach. They demonstrated that concepts with a large variety of near-optimal solutions might be more preferable than concepts with a narrow PF that is a part of the s-Pareto frontier. Instead, they suggested that the entire PF of each concept should be used to compare different concepts. Two qualities were considered: *optimality* and *variability*. Optimality is defined by using a binary quality indicator between every two concepts, and grading each concept according to the number of successful comparisons. The quality indicator is based on the distances of solutions from a pre-defined vector that expresses the DM's ROI. Variability is measured by the hypervolume indicator, using the ROI vector as the reference point. Different concepts are then compared in the bi-objective domain of optimality-variability, to support a selection of one of them.

Avigad et al. (2011) introduced a different approach for concept-based optimization. In this study they considered the versatility of a family of designs to satisfy several requirements, expressed as a set of ROIs. The performance of each set was also converted to an auxiliary bi-objective space, considering requirement satisfaction and proximity in design space. It was assumed that similar products are associated with lower manufacturing costs and easier adaptation from one product to another within the set. Unary quality indicators were used for both measures.

It is evident from the literature that when a set of vectors needs to be evaluated, only its PF should be considered, and non-optimal members should be ignored. Some of the quality indicators that were reviewed in this section can be used for evaluating the performance of changeable products for multiple objectives. This will be explored in Chapter 4.

2.3.5 Dynamic Optimization

Optimization problems that search for a solution to changing objective functions and constraints are known as *Dynamic Optimization Problems* (DOPs). Mathematically, a DOP is defined as follows:

$$\min_{\mathbf{y}\in\mathcal{Y}} \gamma(\mathbf{y}, \mathbf{p}) \,. \tag{2.19}$$

where $\mathbf{y}$ is an n_y-dimensional vector of adjustable decision variables from some feasible region $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$, and $\mathbf{p}$ is a vector of time varying parameters. In some cases, the feasible domain, $\mathcal{Y}$, might also change with time. For any given vector, $\mathbf{p}$, the solution of the DOP is the vector, $\mathbf{y}$, that minimises the objective function.

The unique feature of a DOP, which distinguishes it from other optimization problems, is that the design variables can be adjusted whenever the optimum changes within the design space. In the context of adaptive products, these design variables are therefore considered as type $\mathbf{y}$, i.e., the adjustable variables.

Some researchers consider dynamic optimization to be a special case of robust optimization (RO) (e.g., Jin and Branke, 2005). However, this study makes a clear distinction between the two fields. While in RO the solution needs to be found prior to the realisation of the uncertainties (see Section 2.4), in a DOP it is searched for once a particular environmental condition is realised. This distinction is very important when optimizing changeable products, where decisions need to be made both before and after the realisation of the uncertainties.

The fields of robust optimization and dynamic optimization have been comprehensively studied during the past two decades, though the synergy between these two optimization approaches has received scarce attention. The proposed AROP uses both robust and dynamic optimization: the properties that cannot change with time are optimized through RO, while the adaptation of adjustable properties to the changing environment is analysed by using dynamic optimization.

Currently, evolutionary algorithms are the predominant approach for solving this class of problems (Branke, 2002; Cruz et al., 2011), but variations of other optimization methods exist to cope with dynamic environments. Some of them are: particle swarm optimization (Blackwell and Branke, 2004; Du and Li, 2008), ant colony (Lee and Park, 2001; Guntsch et al., 2001), immune-based algorithms (Gasper et al., 1999; Trojanowski and Wierzchoń, 2009; Rezvanian and Meybodi, 2010). Commonly, evolutionary algorithms for DOPs consist of a mechanism for continuously tracking the optimum over time, and an additional mechanism for seeking a new optimum in other regions of the design space. Please refer to Cruz et al. (2011) and Nguyen et al. (2012) for comprehensive surveys of the existing methods for solving DOPs and their applications. A recent survey of evolutionary algorithms for solving multi-objective DOPs, and benchmarks for such problems, can be found in the work of Jiang and Yang (2017).

2.4 Robust Optimization

The term *robust optimization* is not uniquely defined in the literature, and is used to describe several classes of optimization problems. In the scope of this study, robust optimization is used to describe all optimization problems that include uncertainties. The source of uncertainty can be the environmental parameters (Type A), design variables (Type B), objective functions or constraints (both are Type C), or any combination of the above. Similar to *robust design*, RO is concerned with minimising the effect of uncertainties and variation without eliminating their source. The aim is to find *robust solutions* – solutions that perform well with respect to the uncertainties involved, even if they are not the optimum solutions for the nominal conditions.

This section presents the basic concepts of RO. It starts with the popular methods for quantifying robustness of candidate solutions through robustness indicators, then an overview of RO for multiple objectives is provided, and finally, the scarce literature available on RO of changeable systems is surveyed. For a general overview of RO, please refer to the surveys that are discussed below.

In the field of mathematical programming, a distinction is made between *stochastic* and *robust* optimization. The first considers the uncertain variables as probabilistic values with certain distribution functions, and the latter considers them as a deterministic set of values, where the robust solution needs to be optimal over the entire set (i.e., the worst-case).

Bertsimas et al. (2011a) considered the different types of RO problems addressed in the mathematical programming literature. They focused on the computational tractability and applicability of each approach, as well as their conservativeness when compared with stochastic optimization methods.

The textbooks of Birge and Louveaux (1997) and Kall and Wallace (1994) serve as a good base for understanding stochastic optimization.

Jin and Branke (2005) focused on evolutionary approaches for solving RO problems. In the evolutionary optimization community the distinction between robust and stochastic optimization is not made, and both cases are considered as RO. In their survey, Jin and Branke considered uncertainties of type A, B and C, as well as dynamic optimization as a type of optimization under uncertainty.

Beyer and Sendhoff (2007) provided a wide perspective on the various types of

RO problems and the different approaches for solving them, including mathematical programming and meta-heuristics such as evolutionary algorithms.

A general robust optimization problem can be formulated as follows:

$$\min_{\mathbf{x}\in\mathcal{X}} \Phi(\mathbf{x}, \mathbf{p}, \mathbf{U}), \tag{2.20}$$

where $\mathbf{x}$ are the decision variables that need to be optimized and $\mathbf{p}$ are the uncontrolled parameters. Here, $\mathbf{U}$ is a vector of random variables that includes all the uncertainties associated with the optimization problem. A single scenario of the variate $\mathbf{U}$ is denoted as $\mathbf{u}$. Since uncertainties are involved, the objective function, Φ, is also a random variate, where every scenario of the uncertainties, $\mathbf{u}$, is associated with an objective value, ϕ.

2.4.1 Robustness Indicators

In a robust optimization scheme, the random objective function is replaced with a robustness criterion, denoted by the indicator $I[\Phi]$. Several criteria are commonly used in the literature, which can be broadly categorised into three main approaches:

1. **Worst-Case Scenario.** The worst objective vector, considering a bounded domain in the neighbourhood of the nominal values of the uncertain variables.

2. **Aggregated Value.** An integral measure of robustness that amalgamates the possible values of the uncertain variables (e.g., mean value and variance).

3. **Threshold Probability.** The probability that the objective function would be better than a predefined threshold which is considered as "good enough".

Worst-Case Scenario

The robust indicator for the problem in Equation (2.20) considering a *worst-case* criterion is defined as follows:

Definition 2.5 (Worst-case robustness indicator)**.**

$$I_w\left[\Phi(\mathbf{x}, \mathbf{p}, \mathbf{U})\right] := \max_{\mathbf{u}\in\mathbf{U}} \phi(\mathbf{x}, \mathbf{p}, \mathbf{u}). \tag{2.21}$$

For example, consider a problem involving Type B uncertainties, where the values of $\mathbf{x}$ are bounded between $\mathbf{x} \pm \boldsymbol{\delta}$. Here $\boldsymbol{\delta}$ can represent a vector of specified manufacturing tolerances. For the worst-case criterion, the robust optimization problem is formulated as follows:

$$\min_{\mathbf{x}\in\mathcal{X}} I_w\left[\Phi(\mathbf{X},\mathbf{p})\right] \equiv \min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{x}\in\Delta} \phi(\mathbf{x},\mathbf{p})\,, \tag{2.22a}$$

$$\text{where:}\quad \Delta = \{\mathbf{x} \mid \mathbf{x}-\boldsymbol{\delta} \leq \mathbf{x} \leq \mathbf{x}+\boldsymbol{\delta}\}\,. \tag{2.22b}$$

The main deficiency of the worst-case indicator is that all possible scenarios must be considered. This implies that either an analytic description of the random function value is available, or all extreme cases can be evaluated. Typically, both are not possible, and finding the worst-case scenario might require an optimization search itself (see Branke and Rosenbusch, 2008; Lu et al., 2016, for example). Therefore, in applications where the worst-case performance must be considered, safety factors are commonly used to account for the fact that some scenarios cannot be foreseen.

Aggregated Value

The *aggregated value* approach is suitable for uncertainties of a probabilistic nature. It includes expectancy measures of the function value, or its variance (or possibly both). The expected value of a random variate V is defined as follows:

$$\mathrm{E}(V) := \int_{-\infty}^{\infty} v \cdot f(v)\,\mathrm{d}v, \tag{2.23}$$

where $f(v)$ is the probability density function for the random value V.

The expected value indicator for the problem in Equation (2.20) is defined as follows:

Definition 2.6 (Expected value robustness indicator)**.**

$$I_E\left[\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\right] := \mathrm{E}\left[\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\right]. \tag{2.24}$$

The variance indicator for the problem in Equation (2.20) is defined as follows:

Definition 2.7 (Variance robustness indicator)**.**

$$I_v\left[\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\right] := \mathrm{var}\left[\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\right]. \tag{2.25}$$

When the probability density functions for the uncertainties $\mathbf{f}(\mathbf{u})$ are available, the expectancy measure can be derived by the integral:

$$\mathrm{E}\big(\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\big) = \int_{\mathbf{U}} \phi(\mathbf{x},\mathbf{p},\mathbf{u})\,\mathbf{f}(\mathbf{u})\,\mathrm{d}\mathbf{u}. \tag{2.26}$$

The variance indicator can be similarly derived:

$$\mathrm{var}\big(\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\big) = \int_{\mathbf{U}} \big(\phi(\mathbf{x},\mathbf{p},\mathbf{u}) - \mathrm{E}\left[\Phi(\mathbf{x},\mathbf{p},\mathbf{U})\right]\big)^2 \mathbf{f}(\mathbf{u})\,\mathrm{d}\mathbf{u}. \tag{2.27}$$

Commonly in real-world problems, the distribution of the uncertain variables is not known, and information is extracted by using Monte Carlo simulations to produce a set of values $\bar{\mathbf{U}}$. The integral measures in Equations (2.26) and (2.27) then become:

$$\mathrm{E}\Big(\bar{\Phi}\big(\mathbf{x},\mathbf{p},\bar{\mathbf{U}}\big)\Big) = \frac{1}{\left|\bar{\mathbf{U}}\right|}\sum_{\mathbf{u}\in\bar{\mathbf{U}}} \phi(\mathbf{x},\mathbf{p},\mathbf{u}) \tag{2.28}$$

and

$$\mathrm{var}\Big(\bar{\Phi}\big(\mathbf{x},\mathbf{p},\bar{\mathbf{U}}\big)\Big) = \frac{1}{\left|\bar{\mathbf{U}}\right|}\sum_{\mathbf{u}\in\bar{\mathbf{U}}} \Big(\phi(\mathbf{x},\mathbf{p},\mathbf{u}) - \mathrm{E}\left[\bar{\Phi}\big(\mathbf{x},\mathbf{p},\bar{\mathbf{U}}\big)\right]\Big)^2, \tag{2.29}$$

respectively, where $\left|\bar{\mathbf{U}}\right|$ is the cardinality of the sampled set, $\bar{\mathbf{U}}$.

When both mean performance and variance are of interest, the robust optimization problem can be formulated as an MOP:

$$\min_{\mathbf{x}\in\mathcal{X}} \left[I_E, I_v\right] \tag{2.30}$$

To address different robustness criteria other than the indicators in Equations (2.24) and (2.25), the objective function ϕ can be replaced with a utility function $\Upsilon(\phi)$. For example, in Equation (2.24), using the utility function $\Upsilon(\phi) = \phi^a$ will result in different criteria, according to the value of a:
$a = 1$ produces the expected function value,
$a > 1$ amplifies the effect of outliers, and
$0 < a < 1$ dampens the effect of outliers.

Threshold Probability

It is possible to address the probability of the objective function directly as a robustness measure by setting a performance target. A threshold, q, is considered as a satisficing performance for the objective value ϕ. When ϕ is uncertain, denoted by the random variable Φ, the probability for ϕ to satisfy the threshold level can be seen as a confidence level c. For a minimization problem this can be written as follows:

$$c(\Phi, q) = \Pr(\Phi \leq q)\,. \tag{2.31}$$

For the discrete version of Equation 2.31, consider the binary function

$$\beta(\phi, q) = \begin{cases} 1, & \text{for } \phi \leq q, \\ 0, & \text{otherwise.} \end{cases} \tag{2.32}$$

Given a set of samples with equal probability for the uncertain variables, $\overline{\mathbf{U}}$, and a corresponding set of objective values, $\bar{\Phi}$, Equation 2.31 can be replaced with:

$$c\big(\bar{\Phi}, q\big) = \frac{1}{|\bar{\Phi}|} \sum_{\phi \in \bar{\Phi}} \beta(\phi, q)\,. \tag{2.33}$$

Equation (2.31) can be used for two different robustness indicators:

1. Maximization of the confidence level c for a given threshold q, denoted as $I_q[\Phi, q]$. This measure can be used when the target for performance is known, and the emphasis is on meeting this target, rather than performing as well as possible.

2. Optimization of the threshold q for a pre-defined confidence level c, denoted as $I_c[\Phi, c]$. This is useful when there is no specific target for performance, but the confidence in the resulting performance can be specified. The preferred solution is the one that guarantees the best performance with the specified confidence.

The target-based robustness indicator I_q describes the confidence of the objective function to be better than a threshold, q. It is defined as follows:

Definition 2.8 (Target-based robustness indicator)**.** *Let Φ be a random objective function with a cumulative distribution function $F(\phi)$, and let q be a desired target for ϕ. If Φ is to be minimized, then*

$$I_q[\Phi, q] = Pr(\Phi \leq q) = F(q)\,. \tag{2.34}$$

If Φ is to be maximized, then

$$I_q[\Phi, q] = Pr(\Phi \geq q) = 1 - F(q)\,. \tag{2.35}$$

Using the target-based robustness indicator, the problem in Equation (2.20) becomes:

$$\max_{\mathbf{x} \in \mathcal{X}} I_q[\Phi(\mathbf{x}, \mathbf{p}, \mathbf{U})\,, q]\,. \tag{2.36}$$

For a sampled representation of the uncertain variables, $\overline{\mathbf{U}}$, and the objective function, $\overline{\Phi}$, the target-based indicator is defined as follows:

Definition 2.9 (Discrete target-based robustness indicator)**.**

$$I_q\left[\overline{\Phi}, q\right] = \frac{1}{\left|\overline{\Phi}\right|} \sum_{\phi \in \overline{\Phi}} \beta(\phi, q)\,, \tag{2.37}$$

The confidence-based robustness indicator, I_c, can be used when the objective function is to be optimized, while a pre-defined confidence in the obtained value needs to be assured. It is defined as follows:

Definition 2.10 (Confidence-based robustness indicator)**.** *Let Φ be a random objective function with a cumulative distribution function, $F(\phi)$, and let c be a desired confidence level. If Φ is to be minimized, then*

$$I_c[\Phi, c] = \underset{\phi}{\operatorname{argmax}}(F(\phi) \leq c)\,. \tag{2.38}$$

If Φ is to be maximized, then

$$I_c[\Phi, c] = \underset{\phi}{\operatorname{argmin}}(F(\phi) \geq 1 - c)\,. \tag{2.39}$$

In other words, there is a confidence level of c that a realisation of Φ would be better than $I_c[\Phi, c]$. The worst-case indicator is, in fact, a special case of the confidence-based indicator, where a confidence level of $c = 100\%$ is required.

For a sampled representation of the uncertain variables, $\overline{\mathbf{U}}$, and the objective function, $\overline{\Phi}$, the confidence-based indicator, $I_c\left[\overline{\Phi}, c\right]$, is defined as the c^{th} percentile of the set, $\overline{\Phi}$.

2.4.2 Robust Multi-Objective Optimization

Recently, the presence of uncertainties in multi-objective optimization problems is gaining increasing attention. Whenever uncertainties are considered in an MOP, every candidate solution is associated with a random objective vector and/or constraint vector. Finding a set of robust solutions to an uncertain MOP is a challenging task, affected by the type of uncertainties involved, and the manner in which they propagate to the objective functions and constraints.

Uncertain MOPs can be constructed in different ways to resemble situations that may arise in real-world optimization problems. Most studies on robust multi-objective optimization transform deterministic MOPs into uncertain MOPs by adding uncertainty factors to different aspects of the problem formulation. Adding noise to the objective functions is the most common practice (Teich, 2001; Hughes, 2001; Buche et al., 2002; Fieldsend and Everson, 2005, 2014; Goh and Tan, 2007; Knowles et al., 2009; Syberfeldt et al., 2010; Shim et al., 2013). Noise can also be added to the decision variables to resemble inaccuracies in the manufacturing process or deterioration (Deb and Gupta, 2006; Gaspar-Cunha et al., 2013; Mirjalili and Lewis, 2015; Meneghini et al., 2016). Uncertainties in uncontrolled parameters are also considered (Gunawan and Azarm, 2005; Mattson and Messac, 2005; Avigad and Branke, 2008; Hu et al., 2013), as well as a combination of the above (Basseur and Zitzler, 2006).

A general description for the stochastic features in uncertain MOPs can be found in the studies of Goh et al. (2010) and Salomon et al. (2016b). Goh et al. (2010) have developed a generic method that can transform any deterministic MOP into a stochastic one by injecting a parametric configurable noise function to various parts of the problem formulation. Salomon et al. (2016b) have presented a toolkit to generate uncertain MOPs that allows for direct control over the stochastic properties of the problem.

The definition of robustness varies according to the manner in which uncertainty is considered in the problem, and the algorithms for solving uncertain MOPs are designed accordingly. *Probabilistic dominance* was defined by Teich (2001) to search for candidate solutions that have the highest probability to be non-dominated. It was used to replace the standard domination relation within a *strength Pareto* approach (SPEA, Zitzler and Thiele, 1999). *Probabilistic ranking* was considered by Hughes (2001) for a set of candidate solutions, according to the probability every solution has for dominating the other solutions in the set.

The *'true' objective vector* is a straightforward robustness measure when the uncertainty is generated by adding noise to the objective values. The robust solution to uncertain MOPs of this kind is the same Pareto front as the one without the noise. The motivation in studies that use this measure is to suggest an efficient algorithm that can "filter" the noise in objective functions to find the same set of solutions as if there was no uncertainty. The 'true' objective values are assumed to be the *expected values* of the noisy functions. Some examples are the studies of Fieldsend and Everson (2005, 2014), Knowles et al. (2009), Goh and Tan (2007), Syberfeldt et al. (2010) and Shim et al. (2013).

The *expected value* of the variate objective vector is also used when the decision variables are the source of uncertainty. Deb and Gupta (2006) aimed at solutions that are less sensitive to variations from the nominal decision variables. Robustness was defined in this context as the expected value of the variate objective vector mapped from a neighbourhood around the nominal decision vector.

Aggregated measures may also consider the variance in addition to the expected function values. For example, Mattson and Messac (2005) replaced every stochastic objective with its expected value plus k standard deviations. The expected value was also used by Basseur and Zitzler (2006) in an indicator-based optimization framework. They considered the expected indicator value to assess a set of candidate solutions. While aiming at finding the set that optimizes the expected indicator value, for complexity reasons, either the best-case, worst-case or average indicator value was calculated based on a sample.

Sensitivity was used by Gunawan and Azarm (2005), Barrico and Antunes (2006) and Gaspar-Cunha et al. (2013) as a measure for robustness. It is measured according to the changes in objective values due to changes in decision variables or parameters (depending on the type of uncertainty under consideration). Gunawan and Azarm (2005) defined an acceptable sensitivity of the objective vector to variations of uncontrolled parameters. This sensitivity was later used as a constraint when optimizing the nominal objective functions. Barrico and Antunes (2006) applied a penalty to the nominal objective values according to the sensitivity of a candidate solution. Gaspar-Cunha et al. (2013) considered the average sensitivity to changes in decision variables in a neighbourhood of a candidate solution.

Worst-case optimization is applied in several studies on multi-objective optimization in the presence of uncertainties. When considering the marginal distributions of the objectives, each uncertain objective value can be replaced with its worst-case (Kuroiwa and Lee, 2012; Fliege and Werner, 2014). Avigad and Branke (2008) considered the irregular shape of the random objective vector due to uncertain parameters. The worst-case of a candidate solution is represented by the Pareto front of a reversed problem achieved by maximizing over the uncertainty domain (e.g., finding the scenario of uncertain parameters that maximizes the objectives of a minimization problem). To find the robust set of solutions, a nested EA was used, where the inner algorithm searched for the worst-case scenarios and the outer for the best solutions. The notion of *set dominance* was used to find the robust set of solutions. Meneghini et al. (2016) used a co-evolutionary algorithm to find the robust set of solutions for a worst-case problem. Together with the population of candidate solutions, a population of scenarios for the uncertain variations is evolved. This approach enables the worst-case scenario to be found together with the least sensitive solutions.

2.4.3 Robust Optimization of Changeable Systems

Until recent years, there has been very little study conducted on the robust optimization of changeable systems. The relevant studies that could be identified from the scarce literature on this topic are listed below.

Multi-stage stochastic optimization problems have been studied in the field of mathematical programming (Pflug and Pichler, 2014; Bertsimas et al., 2011b, and references therein). These problems consider sequential decision-making under dynamic uncertainties, where the decision variables can be constantly changed according to the realisation of the uncertainties. A decision must be taken at each stage by considering accumulated knowledge about the uncertainties and the ability of future decisions to overcome a 'wrong' decision.

For example, consider the following inventory control problem presented by Bertsimas et al. (2011a): A product should be produced at a changeable rate to satisfy a time-varying demand. A wrong decision at Stage i, that results in not satisfying the demand at Stage $i+1$, can be recovered by buying the product from a competitor at a higher price. Another example, presented by Pflug and Pichler (2014) considers the operation of a hydro generation system, consisting of a series of reservoirs. The aim is to maximize the profit by selling energy at peak prices, while maintaining the water capacity in each reservoir throughout the year. Both the energy demand and the rainfall are uncertain, and therefore decisions that are taken at the beginning of the year have an impact on the yield at the end of the year. Note that multi-stage stochastic optimization aims at an optimal strategy for adapting a system's adjustable attributes, rather than optimizing the system itself. For example, the problems mentioned above do not optimize the warehouse infrastructure or the architecture for the system of reservoirs. The scope of studies on multi-stage stochastic optimization is usually restricted to single-objective linear problems.

Ben-Tal et al. (2004) introduced the *adjustable robust optimization* methodology for uncertain linear programming problems. It distinguishes between decision variables that need to be determined before and after the realisation of the uncertainties. Adjustable robust optimization problems can be formulated as multi-stage problems, where a sequence of decisions needs to be made over a period of time (e.g., Ben-Tal et al., 2009), or as bi-level problems, where the optimization includes two stages (e.g., in circuit design where the hardware is designed on the first stage and tuned according to the realisation of the uncertainties on the second stage, see Mani et al., 2006; Yao et al., 2009).

Bertsimas and Caramanis (2010) studied the properties of multi-stage problems with limited adaptability, and compared them against problems with complete adaptability and against static robust optimization. Adjustable robust optimization problems with complete adaptability are defined by Bertsimas and Caramanis (2010) very similarly to the formulation of the AROP, as presented in this work (See Chapter 3). The formulation is given below, using the notations from Bertsimas and Caramanis (2010):

$$\begin{aligned}\text{CompAdapt}(\Omega) &:= \left[\begin{array}{ll}\min: & \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y}(\boldsymbol{\omega}) \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\,\mathbf{x} + \boldsymbol{B}(\boldsymbol{\omega})\,\mathbf{y}(\boldsymbol{\omega}) \le \mathbf{b}, \quad \forall \boldsymbol{\omega} \in \Omega\end{array}\right] \\ &= \min_{\mathbf{x}} \max_{\boldsymbol{\omega}\in\Omega} \min_{\mathbf{y}} \left[\begin{array}{ll} & \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\,\mathbf{x} + \boldsymbol{B}(\boldsymbol{\omega})\,\mathbf{y} \le \mathbf{b}\end{array}\right],\end{aligned} \tag{2.40}$$

where $\boldsymbol{\omega}$ is an uncertain scenario from an uncertainty set Ω, the model parameters, $\boldsymbol{A}$ and $\boldsymbol{B}$, depend on the uncertainties, $\mathbf{x}$ are the first stage decision variables, that do not depend on $\boldsymbol{\omega}$ and $\mathbf{y}$ are the second stage variables that can be decided according to the realisation of $\boldsymbol{\omega}$.

The problem in Equation (2.40) can indeed be considered as an active robust optimization problem. However, it does not focus on changeable products, as described in this study, i.e., how to optimize the basic properties of products that can react to changing conditions. Additionally, Problem (2.40) is restricted to applications that can be modelled as linear systems.

Avigad and Eisenstadt (2010), followed by Lara et al. (2013), demonstrated how active control can promote robustness to physical deterioration in the context of multi-objective optimization. They considered two types of decision variables, similar to those defined in Section 2.3.1, to demonstrate how adjustable variables can compensate for the degradation caused by the deterioration of fixed variables. Robustness was considered as the distance in objective space from the original performance, prior to deterioration. Note that the optimization scheme suggested by Avigad and Eisenstadt (2010) is not an RO scheme. The MOP was formulated and solved without considering the foreseen deterioration. Nevertheless, the robustness achieved by active control was examined, in order to support a decision as to which of the obtained Pareto-optimal solutions should be selected.

Avigad et al. (2010) considered the solution's ability to adapt to changing environmental conditions or requirements in a multi-objective optimization framework. They proposed a methodology for optimizing changeable solutions according to their best performance over the combined domains of adjustable design variables and uncertain environmental parameters. Since the problem setting is an MOP, the best performance of a changeable solution was conceived by its Pareto front. The optimization aim is to find all solutions that do not set-dominate each other.[1] They termed the combined set of Pareto frontiers a "Pareto layer". The methodology of Avigad et al. suffers from

[1]see Section 2.3.4 for information on set-domination.

a crucial unidentified issue that results in an unreal representation of the solution's abilities. The Pareto front of every solution considers its best performance over all possible configurations, as well as over all scenarios of the uncertain parameters. Since the parameters cannot be controlled, they must be treated by robust optimization approaches. This was not investigated by Avigad et al. (2010).

Basseur and Zitzler (2006) proposed an aggregated evaluation measure for sets of uncertain multi-objective vectors, based on a binary indicator. They demonstrated the usage of the measure in a multi-objective robust optimization framework. Although changeable systems were not addressed in this study, the suggested approach can be adopted to evaluate solutions of this type, since they are inherently associated with a set of objective vectors.

Zhang et al. (2013); Zhang (2014); Zhang et al. (2014, 2015) presents a methodology for robust design optimization of adaptable and reconfigurable products. This work was conducted in parallel to the one presented in this thesis and many similarities exist between this study and the one of Zhang et al.[1] In his study, Zhang describes an adaptable design as a product that possesses non-changeable and changeable design variables. The configuration of changeable design variables is determined according to the requirement for the product's functionality. A distinction is also made between design variables and other parameters that affect the functional properties. Uncertainties are considered for the parameters, design variables and requirements. Optimization is applied to the non-changeable design variables while considering the 'right' configuration of the changeable variables given the state of the uncertainty factors.

Optimality in Zhang's framework is measured by the deviation of the functional requirements from their required values. These deviations occur since the nominal values of changeable design variables are calculated to meet the requirements, but the uncertainties in parameters and design variables propagate to the product's simulated performance. Zhang et al. use the term *robustness measure* to describe the expected deviation, which is calculated by integrating over the entire uncertainty domain. A weighted sum approach is adopted to deal with multiple functional requirements.

Zhang's methodology is an important development towards the establishment of a framework for robust optimization of changeable products. However, it suffers from several crucial issues that need to be addressed in order to provide a reliable methodology for optimizing such products. These issues are highlighted in the next section, and additional research required in the field of robust optimization of changeable products is listed.

[1]The terms used by Zhang are modified here to fit the terminology of this study.

2.5 Research Gaps

Considering the state-of-the-art that was surveyed in this chapter, the following research still needs to be conducted in the field of robust optimization of changeable systems:

1. The most basic requirement from a robust optimization scheme is the ability to evaluate and compare alternative solutions in respect of the uncertainties involved. The evaluation consists of two main features:

   (a) Approximate the distribution of random performance metrics, given the distribution of the uncertainty factors.

   (b) Quantify the random performance according to desired robustness criteria. The choice of robustness criteria should comply with the application and the designer's attitude to risk.

   The current state-of-the-art does not include a study that provides such a generic framework for evaluating changeable products. The main challenge in optimizing changeable products is to properly consider the design's adaptability when approximating its random performance.

2. In order to use optimization as part of the design process of changeable products, a new class of robust optimization problem needs to be formulated. The adaptability of the candidate solutions to various types of uncertainty should be an integral part of the problem formulation.

3. The current state-of-the-art does not include a study that addresses the complexities of multi-objective optimization of changeable products. Most studies on design practices of changeable products consider a single performance criterion. The few studies that do consider multiple objectives do not exploit the added value of finding a set of trade-off solutions for *a posteriori* decision-making. Instead, the problem is transformed into a single-objective problem using a weighted sum approach or other utility functions.

   When a changeable product is optimized for multiple objectives, its evaluation becomes a very challenging task. The main challenge is that the product can be adjusted to several configurations by its user to satisfy different preferences between competing objectives. For any realisation of the uncertainties, the available trade-off might be different, and user preferences may change during the life-time of the product.

From all the literature surveyed in this chapter, the methodology presented by Zhang (2014) is the most relevant to this study. However, it does not provide the answers to the research gaps described above. The following points highlight the reasons:

1. In Zhang's study, the reason for adaptation is not to improve a performance objective or objectives, but to satisfy requirements for the functional properties. The requirements are treated somewhat like constraints by defining set-points for the system performance. Although it is not explicitly stated, the aim is to minimize the deviation from these set-points.

2. As stated in the summary of Zhang's dissertation, the method requires a closed-form expression that maps the current requirement and state of the uncertain parameters to the 'correct' values of the changeable variables. A very strong assumption is made that the requirements can always be satisfied, thanks to adaptation.

3. The state of changeable variables is calculated according to nominal values. Some uncertainty factors, such as deviation of non-changeable decision variables from their intended values or uncertainty regarding the requirements, do not exist any more when the optimal configuration needs to be selected. However, this is not exploited, and these factors remain uncertain when calculating the robustness measure.

4. The functional requirements in the study are either uncertain or time-dependent, and therefore a robustness metric is used for optimization. However, the metric choice does not adhere to the common practice behind robust optimization, whereby a robustness criterion is used to quantify a random objective value (Beyer and Sendhoff, 2007). Instead, the robustness metric is taken as the variance, or as a combination of the expected value and variance, of a utility function. This utility function measures how well the product satisfies the different functional requirements in a normalised fashion. Since it is assumed that adaptation can bring the function to its maximum value based on a closed-form expression, its variance (i.e., the robustness metric) is calculated by propagating the variances of the various uncertain parameters to the utility function.

5. When an adaptable product has the ability to change its architecture during operation (i.e, a combinatorial choice between some modular components is an adjustable decision variable), this choice is considered as an uncontrolled parameter with a predicted distribution rather than an actual choice of the user to improve performance (Zhang et al., 2014).

6. The framework aims at finding a single robust adaptable design, by solving a single-objective optimization problem. When multiple requirements exist, the objective function is composed of a weighted sum of the deviations from the different requirements.

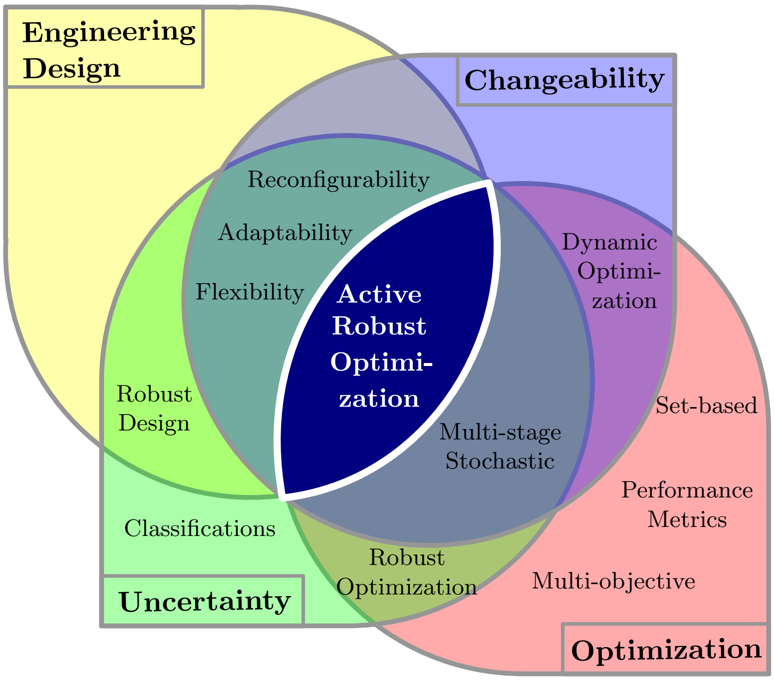


Figure 2.4: The scope of this study.

The review conducted in this chapter examined how uncertainties and changeability are treated in the fields of engineering design and optimization. Relating to Figure 2.1 once again, the existing research does not include studies that combine all four elements in a single framework.

While all relevant studies discussed in this review relate to either two or three of the above, the proposed *Active Robust Optimization* framework, presented in this thesis will combine all four. In Figure 2.4, the scope of the Active Robust Optimization is positioned among the associated fields. The framework uses concepts from the fields of robust, dynamic and multi-objective optimization to conduct optimization of changeable products. The requirements from changeable products and the basic assumptions by which they should be evaluated are burrowed from studies on engineering design of changeable products.

In the following chapter, the framework for *Active Robust Optimization* will be described in details. The *Active Robust Optimization Problem*, that lies in the core of the framework, will be formulated, demonstrated on a simple analytic function and solved for a variety of robustness metrics. A framework for *Active Robust Multi-Objective Optimization*, that handles all the issues raised in this chapter, will be presented in Chapter 4.

Chapter 3

Active Robust Optimization

3.1 Introduction

This chapter establishes the foundations of the new *Active Robust Optimization* framework to evaluate and optimize changeable products. The most basic *Active Robust Optimization Problem* is defined and its features are analysed and discussed using a simple analytic example.

The chapter is organised as follows: Formal definitions of the different types of variables and objective functions are given in Section 3.2. Then, the AROP is defined in Section 3.3.

An example AROP is formulated in Section 3.4. The function characteristics are discussed, and its analytic solution is presented for conditions where no uncertainties exist over any aspect of the problem formulation. Then, the uncontrolled parameter is considered as a random variable, and the stochastic nature of the objective function is examined by propagating the uncertainties from the uncertain parameter.

In Section 3.5 several robustness indicators are used to describe optimality for the uncertain objective function. The AROP is solved for each of the presented indicators, considering different definitions of robustness. The difference between robustness and active robustness is demonstrated through a comparison between the obtained optimal adaptive solutions and their non-adaptive counterparts.

Section 3.6 describes the differences in robustness assignment when sampling methods are used to approximate the uncertain parameters instead of the true underlying distribution. The effects of the sample size on estimation of the true robustness are examined for the various robustness indicators.

Throughout this thesis, the AROP is demonstrated using Type A uncertainties, i.e., uncertain environmental parameters. Section 3.7 presents the effects of uncertainties of other types on the problem formulation and the obtained solutions. The differences and similarities between AROPs with different sources of uncertainty are discussed.

3.2 Definitions

Before the Active Robust Optimization Problem can be properly formulated, the basic terminology is defined in this section.

3.2.1 Variables

A distinction is made between three types of variables: *design variables*, *adjustable variables* and *uncertain parameters*. Their definitions and associated terminology are described in the following.

Definition 3.1 (Design variable)**.** *A property of the product that can be determined by the designer. Denoted by the letter x. Once the product is realised, the value of x cannot be modified.*

Definition 3.2 (Candidate design)**.** *The minimum set of design variables required to describe a product. Denoted by the vector $\mathbf{x} = [x_1, \ldots, x_{n_x}] \in \mathbb{R}^{n_x}$.*

Definition 3.3 (Feasible design space)**.** *The set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ of all feasible candidate designs. Defined by the upper and lower bound of each design variable, and a set of equality and inequality constraints:*

$$l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n_x, \tag{3.1}$$

$$g_j(\mathbf{x}) = 0, \quad j = 1, \ldots, n_g, \tag{3.2}$$

$$h_k(\mathbf{x}) \leq 0, \quad k = 1, \ldots, n_h. \tag{3.3}$$

Definition 3.4 (Adjustable variable)**.** *A property that can be constantly modified during normal operation, i.e., after the product has been realised. Denoted by the letter y.*

Definition 3.5 (Configuration)**.** *A unique combination of values for all adjustable variables, denoted by the vector $\mathbf{y} = \left[y_1, \ldots, y_{n_y}\right] \in \mathbb{R}^{n_y}$. The configuration is determined by the user of the product, either manually or automatically.*

Definition 3.6 (Adaptation)**.** *A change of the adjustable variables from one configuration to another.*

Definition 3.7 (Adaptability)**.** *The set of all possible configurations of a candidate design* $\mathbf{x}$*, denoted as* $\mathcal{Y}(\mathbf{x})$*. The product's adaptability may differ from one candidate design to another.*

Definition 3.8 (Environmental parameter)**.** *A variable that affects the performance of the candidate design and can neither be influenced by the designer nor the user. Denoted by the letter* p*.*[1]

Definition 3.9 (Environmental scenario)**.** *A unique combination of values for all environmental parameters, denoted by the vector* $\mathbf{p} = \left[p_1, \ldots, p_{n_p}\right] \in \mathbb{R}^{n_p}$*.*

Definition 3.10 (Uncertain parameter)**.** *An environmental parameter whose value cannot be determined during the design process. Instead it is described by a random variable* P*. A realisation of the uncertain parameter* P *is denoted by the letter* p*.*

Definition 3.11 (Environmental space)**.** *A vector random variate* $\mathbf{P} \in \mathbb{R}^{n_p}$ *describing all possible scenarios of* $\mathbf{p}$ *and their probabilities.*[2]

3.2.2 Objective Functions

Definition 3.12 (Objective function)**.** *A mapping from the design and environmental spaces to the objective space* $z : \mathbb{R}^{n_x+n_y+n_p} \rightarrow \mathbb{R}$*.*

Within the framework of Active Robust Optimization, three types of objective functions are considered. The type of function is determined according to its sensitivity to uncertainties and whether or not it can be changed by adaptation.

Definition 3.13 (Deterministic function)**.** *A function* $\psi(\mathbf{x}) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ *depends only on the decision variables, and is not affected by uncertain parameters.*

For example:

$$\psi(\mathbf{x}) = x_1 \cos(x_2)\,. \tag{3.4}$$

Definition 3.14 (Stochastic function)**.** *A function* $\phi(\mathbf{x}, \mathbf{p}) : \mathbb{R}^{n_x+n_p} \rightarrow \mathbb{R}$ *depends on uncertain parameters and cannot be affected by adaptation.* $\Phi(\mathbf{x}, \mathbf{P})$ *is the variate of* ϕ *that corresponds to the variate* $\mathbf{P}$*.*

For example:

$$\Phi(\mathbf{x}, P) = x_1 \cos(P - x_2)\,. \tag{3.5}$$

[1] The word 'environmental' is often discarded, and it is simply referred to as 'parameter'.

[2] Without loss of generality, the AROP is presented with Type A uncertainties (i.e., with uncertain environmental parameters). AROPs with different sources of uncertainties will be discussed in Section 3.7.

Definition 3.15 (Adaptable function). *A function* $\gamma(\mathbf{x}, \mathbf{y}, \mathbf{p}) : \mathbb{R}^{n_x+n_y+n_p} \to \mathbb{R}$ *is a stochastic function that can be affected by adaptation.* $\Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P})$ *is the variate of* γ *that corresponds to the variate* $\mathbf{P}$.

For example, in the function

$$\Gamma(x, y, P) = x \cos(P - y)\,, \tag{3.6}$$

the design variable x needs to be determined during the design phase, when the parameter p is considered as a random variable P, while the adjustable variable y can be determined according to the realisation of p.

The functions in Equations (3.4)–(3.6) will be used in Section 3.4 to analyse the various aspects of the AROP.

3.3 Problem Formulation

An *optimal adaptive solution* is the solution to the following Robust Optimization Problem:[1]

$$\min_{\mathbf{x} \in \mathcal{X}} \Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P})\,, \tag{3.7}$$

where $\Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P})$ is a random variate of function values $\gamma(\mathbf{x}, \mathbf{y}, \mathbf{p})$ that correspond to the variate $\mathbf{P}$, according to the design, $\mathbf{x}$, and the configuration, $\mathbf{y}$.

The problem in Equation (3.7) is a robust optimization problem, since the optimal solution should be robust to the uncertainties in $\mathbf{P}$. The fact that $\mathbf{x}$ is an adaptive solution distinguishes this problem from the common RO problem (as explained in Section 2.4), and makes it an *active* RO problem. For every scenario of the uncertainties in $\mathbf{P}$, the performance of a solution can be affected by changing the $\mathbf{y}$ configuration within the solution's adaptability. As a result, whenever the environmental parameters change, the solution's performance can be improved by adaptation. For a proper evaluation of an adaptive solution, it has to be assessed for each scenario with its best possible performance. This performance is achieved by the optimal configuration for that scenario.[2,3] In order to find the optimal configuration $\mathbf{y}^\star$ in a changing environment,

[1] The AROP is arbitrarily formulated as a minimization problem. The same formulation holds for maximization.

[2] Previous studies in the field also assumed that changeable products should be evaluated according to the optimal configuration in every scenario (e.g., Siddiqi et al., 2006; Ferguson and Lewis, 2006; Olewnik and Lewis, 2006; Denhart, 2013).

[3] The base assumption of the ARO methodology is that the user is rational, and has the will and ability to use the product in its optimal configuration. Please refer to Section 6.3 for a discussion on optimizing changeable products while considering sub-optimal configurations.

one must solve the following dynamic optimization problem:

$$\mathbf{y}^\star = \underset{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}{\operatorname{argmin}}\, \gamma(\mathbf{x}, \mathbf{y}, \mathbf{p})\,. \tag{3.8}$$

Note that in the above formulation, the values of the environmental parameters $\mathbf{p}$ are known at the time of this search. The values of $\mathbf{x}$ are constant (the evaluated design does not change) and therefore, in Equation (3.8), the $\mathbf{x}$ variables are also treated as parameters. However, one or more values of $\mathbf{p}$ can change (which makes this problem a DOP) and so, for best performance, the above DOP should be solved whenever $\mathbf{p}$ changes, and $\mathbf{y}$ should be adapted to the new $\mathbf{y}^\star$. The optimization can be done either on-line or off-line, depending on how rapid the response should be.

Considering the entire environmental uncertainty, a one-to-one mapping between the scenarios in $\mathbf{P}$ and the optimal configurations in $\mathcal{Y}(\mathbf{x})$ can be defined as follows:

$$\mathbf{Y}^\star = \underset{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}{\operatorname{argmin}}\, \Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P})\,. \tag{3.9}$$

In order to transform the RO problem in Equation (3.7) to an active RO problem, $\mathbf{y}$ should be replaced with $\mathbf{Y}^\star$.

Following the above, an *Active Robust Optimization Problem* (AROP) is formulated:

Definition 3.16 (Active Robust Opimization Problem)**.**

$$\min_{\mathbf{x}\in\mathcal{X}} \Gamma(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P})\,, \tag{3.10}$$

$$\textit{where:} \quad \mathbf{Y}^\star = \underset{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}{\operatorname{argmin}}\, \Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P}). \tag{3.11}$$

It is a multi-stage optimization problem. In order to compute the objective function Γ in Equation (3.10), the DOP in Equation (3.11) has to be solved for every solution $\mathbf{x}$ with the entire environment universe $\mathbf{P}$.

3.4 Analytic Example

The active robust optimization problem has some very unique characteristics that do not exist in other optimization problems. In order to observe these special features and analyse them, an analytic example is presented in this Section. The problem is constructed in the simplest possible way to include all features of an AROP. This makes it possible to isolate the effects of each of these features, and study them separately. The example AROP consists of a single decision variable, a single adjustable variable

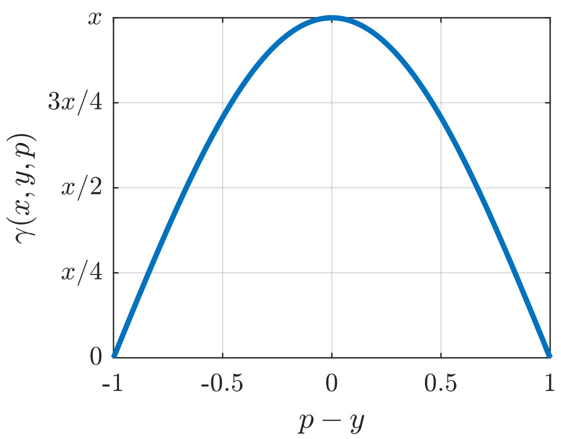


Figure 3.1: Visual interpretation of the objective function in Equation (3.12).

and a single parameter. Owing to the problem's simplicity, its analytic solution can be derived, and various aspects of the active robust optimization problem can be studied.

3.4.1 Problem Formulation

Consider the following objective function that needs to be maximized:

$$\gamma(x, y, p) = x \cos\left(\tfrac{\pi}{2}(p - y)\right), \tag{3.12}$$

where x is a design variable and y is an adjustable variable that can respond to variation in the uncontrolled parameter p. Figure 3.1 depicts the relations between x, y, p and γ. The upper bound for the objective function γ is x, and therefore, the larger x is, the larger the value that can be achieved for the best case. This best case occurs when $p - y = 4k, \quad k = 0, \pm 1, \pm 2, \ldots$

First, the problem is formulated without considering uncertainties in the parameter p:

$$\max_{x \in \mathcal{X}} \gamma(x, y^\star, p)\,, \tag{3.13}$$

$$\text{where:} \quad \gamma(x, y^\star, p) = x \cos\left(\tfrac{\pi}{2}(p - y^\star)\right), \tag{3.14}$$

$$y^\star = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} \gamma(x, y, p)\,, \tag{3.15}$$

$$\mathcal{X} = (0, 1]\,, \tag{3.16}$$

$$\mathcal{Y}(x) = [x - 1, 1 - x]\,. \tag{3.17}$$

Equation (3.17) recognises the case where the product's adaptability may differ from one candidate design to the other. Assuming p can only take values within the interval

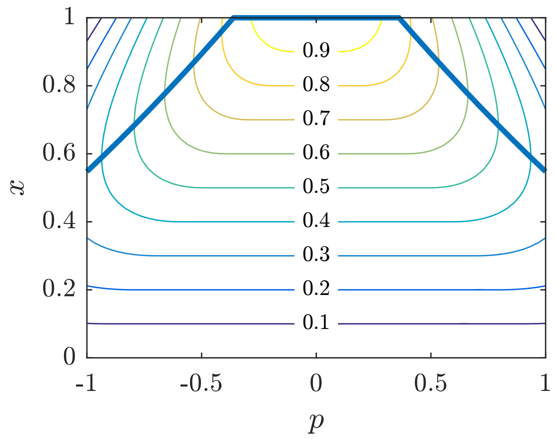


Figure 3.2: Function values with optimal configuration, according to Equation (3.19), for different combinations of x and p. Solutions with maximum function value for every value of p are marked with a solid line.

$-1 \leq p \leq 1$, the unconstrained optimal configuration of y is $y^\star = p$. Since the solution's adaptability depends on the value of x—the adjustable variable y is constrained between $\pm(1-x)$—the optimal configuration cannot be achieved for every combination of x and p.

The following is the closed form expression for the constrained solution of Equation (3.15):

$$y^\star(x,p) = \begin{cases} x-1, & \text{for } p < x-1, \\ 1-x, & \text{for } p > 1-x, \\ p, & \text{otherwise.} \end{cases} \tag{3.18}$$

The objective function in Equation (3.14) then becomes:

$$\gamma(x,y^\star,p) = \begin{cases} x\cos\left(\frac{\pi}{2}(p-x+1)\right), & \text{for } p < x-1, \\ x\cos\left(\frac{\pi}{2}(p+x-1)\right), & \text{for } p > 1-x, \\ x, & \text{otherwise.} \end{cases} \tag{3.19}$$

The function values of $\gamma(x,y^\star,p)$ can be seen in Figure 3.2 as contour lines.

3.4.2 Solution for Deterministic Problem

The Active Robustness methodology deals with optimization problems with some level of uncertainty. Before analysing the effects of uncertainties on the problem formulation, a very simple case where no uncertainties exist is examined. This provides an upper bound for the performance of an adaptive solution.

In the case where p is a known constant parameter, there is no need to change the value of y, and both x and y can be determined simultaneously. The solid line in Figure 3.2 shows the optimal value of x for every p between -1 and 1. It is evident from the figure that when $|p| < 0.36$, the optimal solution has no adaptability, i.e., $x^\star = 1, y^\star = 0$. When $|p| > 0.36$, it is better to 'sacrifice' the amplitude of the function in order to reduce the argument of the cosine. The worst optimal performance of $\gamma = 0.36$ is achieved when $p = \pm 1$, with the solution $x^\star = 0.54$ and the configuration $y^\star = \pm 0.46$.

3.4.3 Uncertainty Propagation to the Objective Function

Now let p be a realisation of a random variable P with a known (or estimated) distribution, defined by the probability density function (PDF), $f(p)$. The problem in Equations (3.13)–(3.17) is rewritten to accommodate the uncertainty:

$$\begin{aligned} &\max_{x \in \mathcal{X}} \Gamma(x, Y^\star, P)\,, \\ \text{where:}\quad &\Gamma(x, Y^\star, P) = x \cos\left(\tfrac{\pi}{2}(P - Y^\star)\right)\,, \\ &Y^\star = \operatorname*{argmax}_{y \in \mathcal{Y}(x)} \Gamma(x, y, P)\,, \\ &\mathcal{X} = (0, 1]\,, \\ &\mathcal{Y}(x) = [x - 1, 1 - x]\,. \end{aligned} \tag{3.20}$$

Note the optimal configuration $y^\star$ is replaced with a variate of optimal configurations $Y^\star$ that corresponds to the realization of the variate P. Consequently, the objective function also becomes a random variate Γ.

The probability density function for Γ can be obtained using the *probabilistic transformation method* (Walpole et al., 2007, pp. 211–219) defined in Theorem 3.1.

Theorem 3.1. *Suppose W and V are continuous random variables with probability density functions $f(w)$ and $g(v)$, respectively. Let $w = h(v)$ be a continuous bijective function between the values of V and W over the interval $[a, b]$, so that the inverse function $v = k(w)$ exists. Then the probability distribution of W in the interval $[k(a), k(b)]$ is*

$$f(w) = g(k(w)) \left| \frac{dk}{dw} \right|\,. \tag{3.21}$$

The probabilistic transformation method requires a continuous bijection between V and W. This implies that $h(v)$ needs to be either monotonically increasing or monotonically decreasing. In the case of P and Γ, this condition is not true for all values of p. As can be seen in Figure 3.2, $\gamma(x, p)$ increases monotonically for $p < x - 1$, decreases monotonically for $p > 1 - x$, and remains constant for $x - 1 \leq p \leq 1 - x$.

Theorem 3.2 extends the method to cases where the mapping between the random variables is not a bijective function.

Theorem 3.2. *Let $w = h(v)$ define a mapping between the values of V and W that is not a bijective function over the interval $[a, b]$. If the interval $[a, b]$ can be partitioned into n mutually disjoint intervals*

$$[a, b] = \{[a_1, b_1], [a_2, b_2], \dots [a_n, b_n]\},$$

such that a bijective inverse function

$$v_1 = k_1(w), \quad v_2 = k_2(w), \quad \dots \quad v_n = k_n(w)$$

exists for all n intervals, then the probability distribution of W in the interval $[k(a), k(b)]$ is

$$f(w) = \sum_{i=1}^{n} g(k_i(w)) \left| \frac{dk_i}{dw} \right|. \tag{3.22}$$

Using Equation (3.22), the PDF for $\Gamma(x, Y^\star, P)$ can be obtained for $\gamma \neq x$. The density function at $\gamma = x$ is undefined since it corresponds to all values in the range $x - 1 \leq p \leq 1 - x$. The two inverse functions to Equation (3.19) that correspond to $p < x - 1$ and $p > 1 - x$, respectively, are:

$$p_1(\gamma, x, y^\star) = x - 1 - \frac{2}{\pi} \arccos \frac{\gamma}{x}, \tag{3.23}$$

$$p_2(\gamma, x, y^\star) = 1 - x + \frac{2}{\pi} \arccos \frac{\gamma}{x}. \tag{3.24}$$

Note that $p_1(\gamma) = -p_2(\gamma)$ which is a result of $\gamma(p)$ being a symmetric function. The derivatives of Equations (3.23) and (3.24) are:

$$\frac{\mathrm{d}p_1}{\mathrm{d}\gamma} = -\frac{\mathrm{d}p_2}{\mathrm{d}\gamma} = \frac{2}{\pi x \sqrt{1 - \left(\frac{\gamma}{x}\right)^2}} \tag{3.25}$$

If P is bounded between $[a, b]$, then $\Gamma(x, Y^\star, P)$ is bounded by the following interval:

$$\min\left[\gamma(x, y^\star, a)\ ,\ \gamma(x, y^\star, b)\right] \ \leq\ \Gamma(x, Y^\star, P) \ \leq\ x. \tag{3.26}$$

Otherwise it is bounded between $-x \leq \Gamma(x, Y^\star, P) \leq x$.

Finally, the probability density function $f(\gamma(x, Y^\star, P))$ is defined for values of γ

inside the interval in Equation (3.26):

$$f(\gamma(x, Y^\star, P)) = \frac{2\big(g\big(x - 1 - \frac{2}{\pi}\arccos\frac{\gamma}{x}\big) + g\big(1 - x + \frac{2}{\pi}\arccos\frac{\gamma}{x}\big)\big)}{\pi x\sqrt{1 - \big(\frac{\gamma}{x}\big)^2}}, \tag{3.27}$$

where $g(p)$ is the PDF of the random variable P.

In the following, the distribution of $\Gamma(x, Y^\star, P)$ is analysed for two common distributions of P.

Uniform Distribution

Let p be a random variable that follows a uniform distribution $P \sim \mathrm{U}(-1, 1)$. $g(p)$ in Equation (3.27) can be simply written as

$$g(p) = \begin{cases} 0.5, & \text{for } -1 \leq p \leq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.28}$$

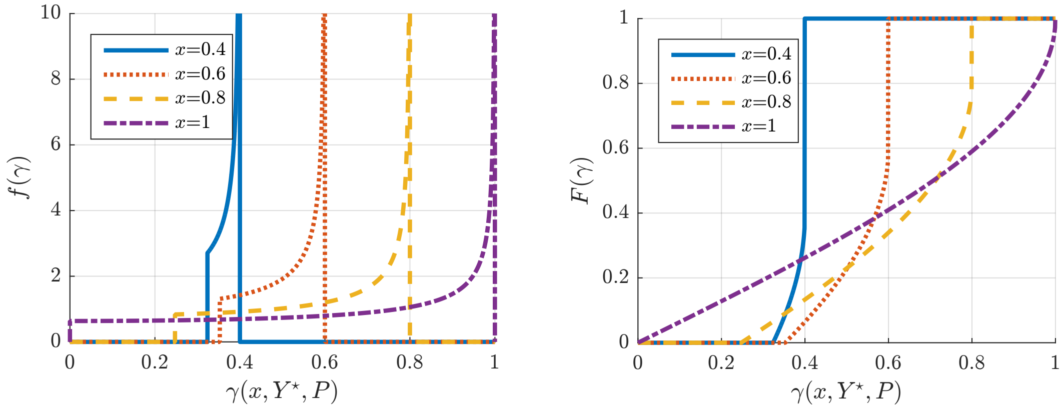
The lower bound for $\Gamma(x, Y^\star, P)$, which corresponds to the extremes of P, is $x \cos x$. Using Equation (3.27), the PDF for $\Gamma(x, Y^\star, P)$ can be obtained:

$$f(\gamma(x, Y^\star, P)) = \begin{cases} \frac{2}{\pi x\sqrt{1-\left(\frac{\gamma}{x}\right)^2}}, & \text{for} \quad x\cos x \leq \gamma < x, \\ 0, & \text{otherwise.} \end{cases} \tag{3.29}$$

Figure 3.3 depicts the PDF, $f(\gamma(x, Y^\star, P))$, and the corresponding cumulative distribution function (CDF), $F(\gamma(x, Y^\star, P))$, calculated for different values of x according to Equation (3.29). It is evident that the variance of Γ increases for larger values of x, due to reduced adaptability. On the other hand, other properties such as selected percentiles, or expected value may decrease. This issue will be discussed in Section 3.5.

Normal Distribution

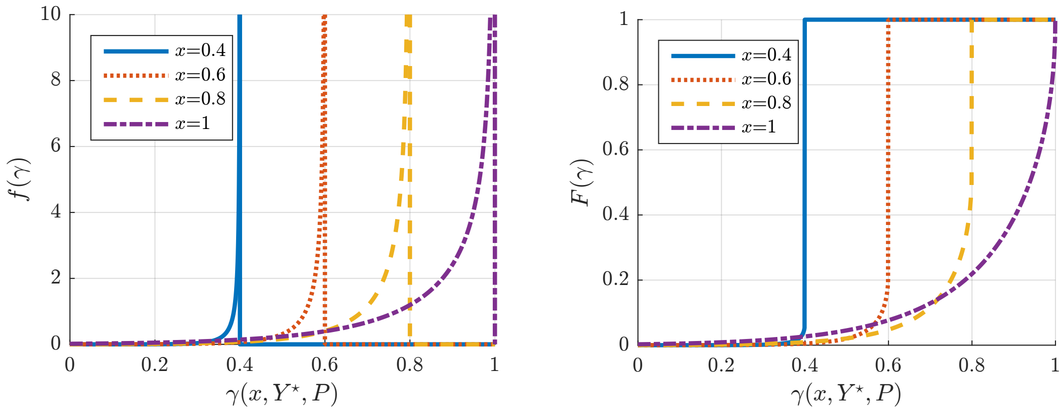
Now let p be a random variable that follows a normal distribution $P \sim \mathrm{N}\big(0, \frac{1}{3}\big)$. A similar exercise can be performed to derive the expression for $f(\gamma(x, Y^\star, P))$ as for the uniform distribution. Figure 3.4 depicts the PDF $f(\gamma(x, Y^\star, P))$ and the corresponding CDF $F(\gamma(x, Y^\star, P))$ for the above distribution of P. Note that for this distribution of P, there is a higher probability for values of p close to $p = 0$ than for the uniform distribution. Therefore adaptability is less required, and as a result for every value of x there is higher density at larger values of γ. On the other hand, P is unbounded, which means that it is possible for $\gamma(x, y^\star, p)$ to have negative values. Although, there is a very small probability for this to happen.



(a) Probability density function.

(b) Cumulative distribution function.

Figure 3.3: Distribution functions of the random objective for different values of x when $P \sim \mathrm{U}(-1, 1)$.



(a) Probability density function.

(b) Cumulative distribution function.

Figure 3.4: Distribution functions of the random objective for different values of x when $P \sim \mathrm{N}(0, \frac{1}{3})$.

3.4.4 Recap

The problem presented in this section is a very simple analytic function. Nonetheless, it consists of several features that may be common in real-world active robust optimization problems.

1. The environmental parameter is uncertain during design phase, but is known during product operation.
2. The adjustable variable can react to changes in the uncertain parameter.
3. The adaptability of the product depends on the design itself.
4. Robustness can be achieved either by enhancing the solution's adaptability or its permanent features. A trade off exists between the two alternatives as they compete for the same resources.

Working with such a simple function has several benefits:

1. The distribution of the objective value can be derived analytically. This means that approximation methods for the solutions' performance are not required, and the true optimal robust solution can be found.
2. The objective function can be calculated very quickly. As a result, billions of function evaluations can be conducted in a reasonable time and there is no need for efficient algorithms to solve the AROP. This enables the study to focus on the aspects of the problem, rather than on algorithmic issues.
3. Having only one decision variable, one adjustable variable and one parameter, makes it easy to modify the problem in order to analyse the effects of different problem features.

In the following, the AROP presented in this section will be used to highlight various aspects of the methodology. The optimization of the random objective function will be addressed as part of a robust optimization scheme. A comparison with the conventional robust optimization approach will be conducted, by altering the above problem to not consider adaptation. Other aspects such as sampling the objective function and considering different sources of uncertainty will also be addressed.

3.5 Optimizing for Robustness

In optimization, candidate solutions are compared against each other in order to promote convergence towards the optimal solution. To do so, every candidate solution is assigned a fitness value, where the optimal solution has the best fitness. In the previous section, the performance of a candidate solution was described as the random variable $\Gamma(x, Y^\star, P)$, where the distribution of Γ depends on the solution of x, the distribution of P and the ability to adapt through y. When optimizing for robustness, the random performance is assigned with a fitness value, according to the manner in which robustness is defined. The definition of robustness is expressed via an indicator $I[\Gamma]$ that quantifies the random performance variable Γ with a scalar value.

The problem in Equation (3.30) describes the AROP in Equation (3.20) as a robust optimization problem, where the aim is to optimize the robustness indicator $I[\Gamma]$. Without loss of generality, the problem is formulated as a maximization problem. The actual choice whether to maximize or minimize depends on the robustness criterion.

$$
\begin{aligned}
& \max_{x\in\mathcal{X}} I[\Gamma(x, Y^\star, P)]\,, \\
\text{where:}\quad & \Gamma(x, Y^\star, P) = x\cos\left(\tfrac{\pi}{2}(P - Y^\star)\right), \\
& Y^\star = \underset{y\in\mathcal{Y}(x)}{\operatorname{argmax}}\, \Gamma(x, y, P)\,, \\
& \mathcal{X} = (0, 1]\,, \\
& \mathcal{Y}(x) = [x-1, 1-x]\,.
\end{aligned}
\tag{3.30}
$$

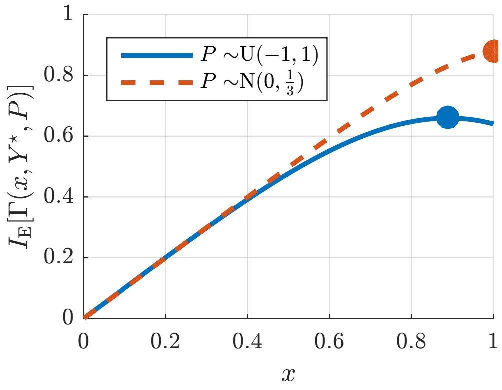
The optimal solution for the AROP in Equation (3.30) depends on the indicator used for describing robustness. In the following, the above AROP is solved for four robustness indicators, previously defined in Section 2.4. It is shown that the optimal solution depends not only on the objective function and the distribution of uncertainties, but also on the choice of robustness criterion.

3.5.1 AROP Solution for Different Definitions of Robustness

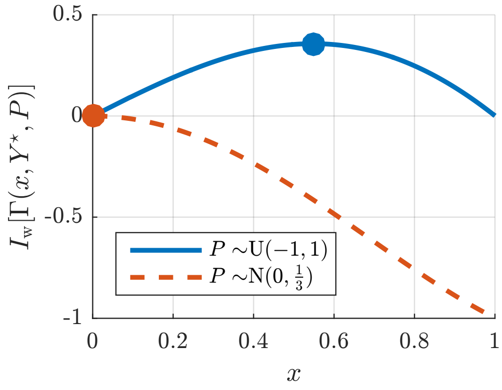
Expected Value

The expected value indicator, defined in Equation (2.24), is a reasonable choice as a robustness criterion when an adaptive (and also non-adaptive) product operates over a long period in a changing environment, and the average performance over time is important. For example, when optimizing a product to yield maximum daily profit in a changing market, the product with the maximum expected value is the one that is most probable to yield the highest profit over an entire year.

Figure 3.5(a) depicts the expected γ value for different values of x. Both indicator values for uniform and normal distributions of P are shown. The optimal robust solution for each distribution is marked with a circle. Note that the optimal solution when P



(a) Expected function value.



(b) Worst function value.

Figure 3.5: Common robustness metrics values as functions of x.

follows a normal distribution is not adaptive at all, i.e., $x = 1, \mathcal{Y}(\mathbf{x}) = 0$.

Worst-Case

For a random objective function, Γ, that needs to be maximized, the worst-case scenario is the lower bound of Γ. Similarly, the upper bound of Γ is the worst-case scenario for minimization. The worst-case robustness indicator $I_w[\Gamma]$ is used when the most conservative design is desired.

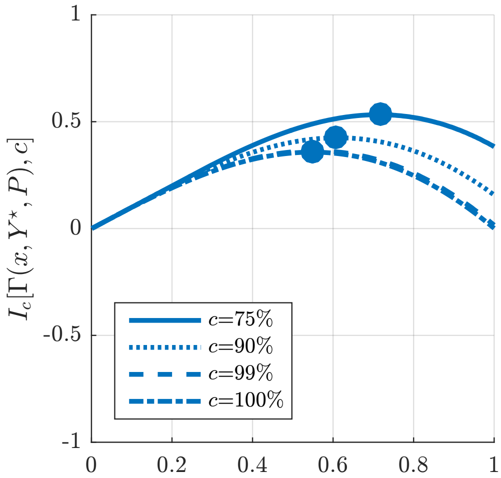
The indicator values for uniform and normal distributions of P are depicted in Figure 3.5(b). Note that the optimal value of x for the uniform distribution, when P is bounded between -1 and 1, is $x = 0.54$, which is the optimal value observed in Figure 3.2 for $p = \pm 1$.

For the normal distribution, the solution that has the highest worst-case performance is $x \to 0$ (x must be larger than 0). This is a good example for illustrating the conservativeness of the worst-case criterion. By minimising the potential damage caused by any possible scenario of the uncertain parameter, the 'optimal' solution also prevents any possible gain in other scenarios. The performance of $x \to 0$ is simply $\gamma \to 0$ regardless of the value of the parameter p. Note that the probability for $|p| > 1$, and therefore $\gamma < 0$ is less than 0.3%. Nevertheless, as long as this probability exists, it needs to be considered for $I_w[\Gamma]$.

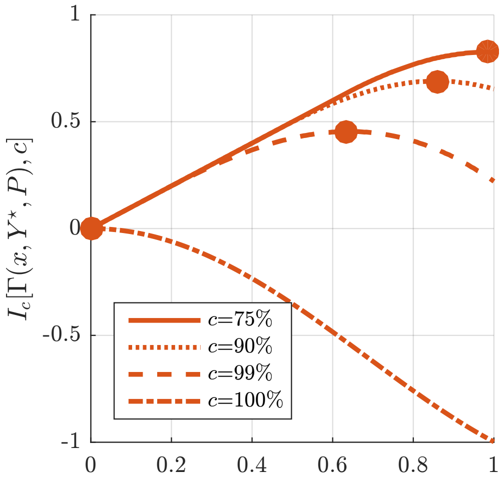
Confidence-Based

The confidence-based robustness indicator can be used when the objective function is to be optimized, while a pre-defined confidence in the obtained value needs to be assured.

Figures 3.6(a) and 3.6(b) depict the values of $I_c[\Gamma(x, Y^\star, P), c]$ when P follows a



(a) $P \sim \mathrm{U}(-1, 1)$.



(b) $P \sim \mathrm{N}(0, \frac{1}{3})$.

Figure 3.6: Confidence-based indicator values as functions of x for different confidence levels c.

uniform and normal distribution, respectively. Four levels of confidence are considered in every figure. Note that the curve for $c = 100\%$ is identical to the curves of the worst-case indicator in Figure 3.5(b). From a comparison between the function of $I_c[\Gamma(x, Y^\star, P), c]$ for the two distributions, several observations can be made:

1. There exists an inverse correlation between the required confidence level and the maximum function value that satisfies this confidence.

2. The value of $I_c[\Gamma, c]$ is sensitive to the distribution of Γ.

3. The solution x with the highest $I_c[\Gamma, c]$ value is sensitive to:

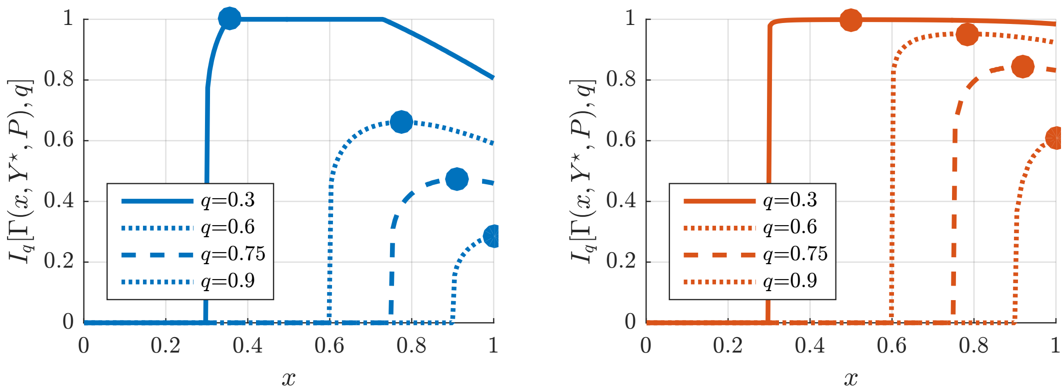
   (a) the distribution of Γ.

   (b) the required confidence level c.

4. While there is almost no difference between $I_c[\Gamma, 99\%]$ and $I_c[\Gamma, 100\%]$ when P follows a uniform distribution, there is a great difference between the two when P follows a normal distribution.

Target-Based

The target-based robustness indicator describes the confidence of the objective function to be better than a predefined target q.



(a) $P \sim \mathrm{U}(-1, 1)$.

(b) $P \sim \mathrm{N}(0, \frac{1}{3})$.

Figure 3.7: Target-based indicator values as functions of x for different goals q.

Figures 3.7(a) and 3.7(b) depict the values of $I_q[\Gamma(x, Y^\star, P), q]$ when P follows a uniform and normal distribution, respectively. The following observations can be made based on Figure 3.7:

1. There exists an inverse correlation between the required target q and the confidence in achieving this target.

2. The confidence for achieving a goal q with a solution $x < q$ is zero.

3. For the above example, the value of $I_q[\Gamma, q]$ is sensitive to the distribution of Γ.

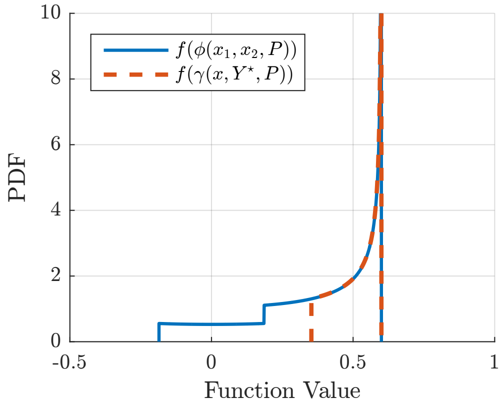
4. However, the solution x with the highest $I_q[\Gamma, q]$ value is not sensitive to the distribution of Γ. The optimal solutions for the uniform distribution have very similar indicator values at the optimum of the normal distribution, and vice versa.

5. For values of $0.36 \le x \le 0.73$, there is a complete confidence for the value of γ for the uniform distribution to be larger than 0.3. This can be verified by examining the functions of $F(\gamma)$ in Figure 3.3(b). $F(\gamma) = 0$ for $x = 0.4$ and $x = 0.6$, while it is larger than zero for $x = 0.8$ and $x = 1$.

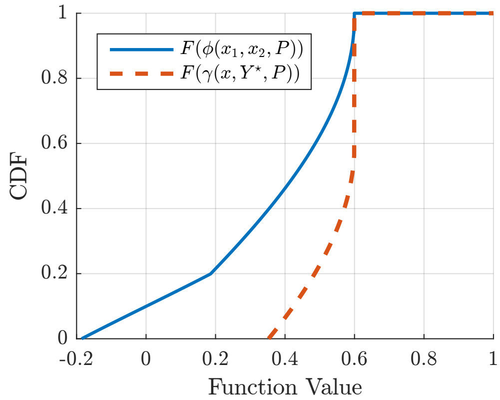
The Target-based robustness indicator can serve as a very useful measure of robustness when targets can guide the optimization process. For example, when designing a product in a competitive market, outperforming the competitor's product by 10% can serve as the target q.

3.5.2 Comparison with a Non-Adaptive Robust Solution

This section demonstrates the difference between robustness and *active* robustness. Considering the four robustness criteria described above, a search for a non-adaptive



(a) Probability density function



(b) Cumulative distribution function

Figure 3.8: Distribution of the objective function for a non-adaptive solution $[x_1, x_2] = [0.6, 0.2]$ and an adaptive solution $x = 0.6$, for $P \sim \mathrm{U}(-1, 1)$.

robust solution is undertaken. A non-adaptive solution for the problem in Equation (3.30) is a solution that cannot adapt the variable y to the realisations of P. Therefore both decision variables are considered as type x, i.e., fixed decision variables. Since the objective function cannot be affected by adaptation, it is denoted as $\Phi(\mathbf{x}, P)$.[1]

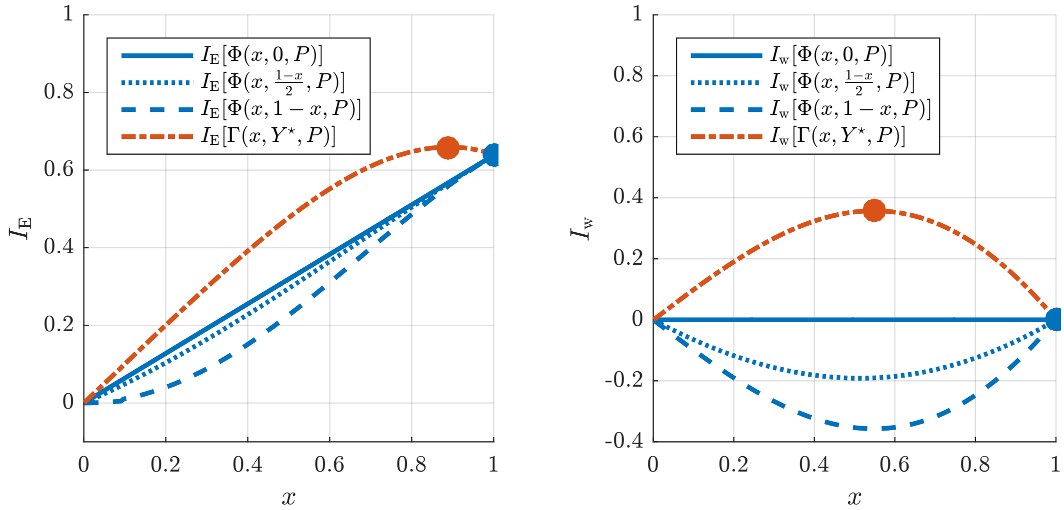
The robust optimization counterpart of the AROP in Equation (3.30) is the following:

$$\begin{aligned} & \max_{\mathbf{x}\in\mathbb{R}^2} I[\Phi(\mathbf{x}, P)]\,, \\ \text{subject to:}\quad & 0 < x_1 \leq 1, \\ & x_1 + |x_2| \leq 1, \\ \text{where:}\quad & \Phi(\mathbf{x}, P) = x_1 \cos\left(\tfrac{\pi}{2}(P - x_2)\right). \end{aligned} \tag{3.31}$$

The PDF and CDF of $\Phi(\mathbf{x}, P)$ can be seen in Figure 3.8 for a solution $\mathbf{x} = [0.6, 0.2]$, where $P \sim \mathrm{U}(-1, 1)$. The distribution of $\Gamma(x, Y^\star, P)$ with $x = 0.6$ (taken from Figure 3.3) is also displayed for comparison. From the CDFs in Figure 3.8(b) it is evident that the adaptive solution outperforms the non-adaptive one for any conceivable robustness criterion. The adaptive solution has a lower (or equal) probability to be smaller than any value of the objective function, compared to the non-adaptive solution. According to the definition of *first-order stochastic dominance*, the performance of the adaptive solution $\Gamma(x, Y^\star, P)$ dominates that of the non-adaptive solution $\Phi(\mathbf{x}, P)$.

When analysing the PDFs in Figure 3.8(a), a disturbing fact can be observed; while the area under the $f(\phi)$ is equal to one, the area under $f(\gamma)$ is smaller than one. The reason for this behaviour is the piecewise description of $\gamma(x, y^\star, p)$, as described in

[1]Please refer to Section 2.3.1 for further details on the differences in notation between adaptive and non-adaptive solutions.



(a) Expected value

(b) Worst-case performance

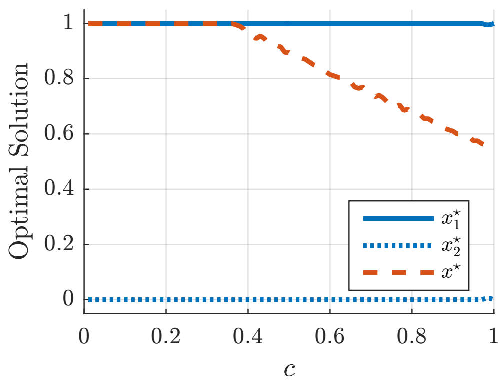
Figure 3.9: Expected value and worst-case performance of adaptive and non-adaptive solutions as functions of x. For the non-adaptive solutions, $x_1 = x$ and three alternatives for x_2 are depicted.

Equation 3.19. For every scenario of the uncertain parameter in the range $-0.4 \le p \le 0.4$, the value of the adaptive objective function is $\gamma = 0.6$. A 'jump' of the CDF $F(\gamma)$ from approx. 0.6 to 1 can be observed in Figure 3.8(b). This means that for the given distribution of P, there is a 40% probability for $\Gamma(x, y^\star, P)$ to be exactly 0.6. This exact probability is the probability for the non-adaptive objective function to have values in the range $-0.2 \le \Phi(\mathbf{x}, P) \le 0.35$.

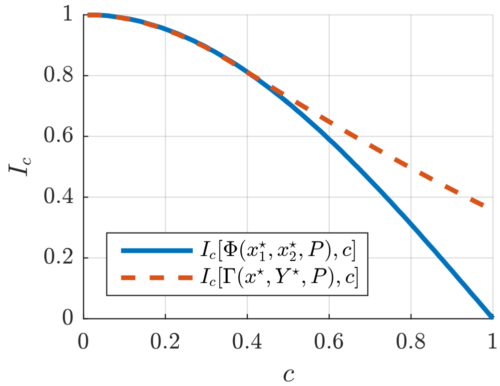
The optimal solution to Equation (3.31) is the vector $\mathbf{x}^\star = [x_1^\star, x_2^\star]$ that maximizes the robustness criterion $I[\Phi]$. In the following, the performance of the adaptive solution $\Gamma(x, Y^\star, P)$ and the non-adaptive solution $\Phi(\mathbf{x}, P)$ is compared against the four robustness indicators described above. In all examples the uncertain parameter follows a uniform distribution $P \sim \mathrm{U}(-1, 1)$.

Expected Value and Worst-Case

Figure 3.9 depicts the expected value and worst-case performance for various non-adaptive solutions and an adaptive solution. The indicator values are shown as a function of x for the adaptive solution and as a function of $x_1 = x$ for the non-adaptive ones. The three blue curves represent different strategies for choosing x_2: the solid line is for $x_2 = 0$; the dashed line is for the maximum value of x_2, i.e., $x_2 = 1 - x_1$; and the dotted line is for a value of an intermediate value, i.e., $x_2 = (1 - x_1)/2$. The solutions to Equations (3.31) and (3.30) are marked with blue and red dots, respectively.



(a) Optimal solution in Decision space



(b) Indicator value

Figure 3.10: Optimal solutions for the confidence-based indicator with different desired confidence levels.

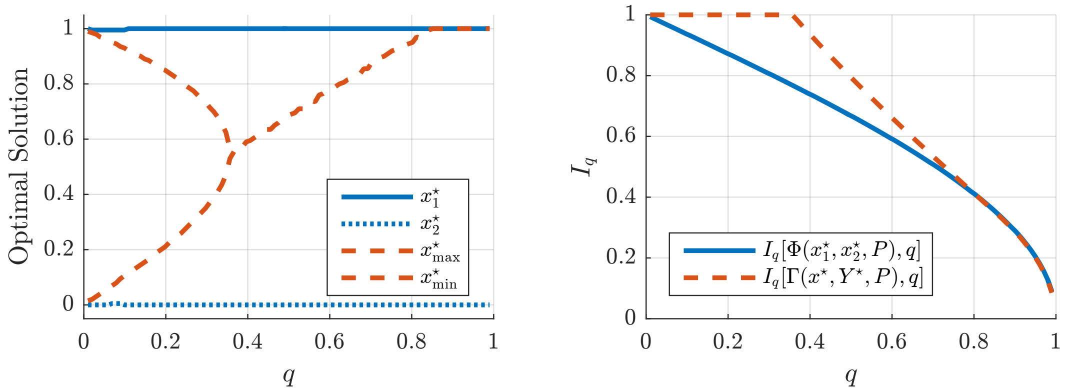
It is interesting to note that once the adjustable variable y becomes fixed, it can only degrade the solution's robustness, and highest robustness is achieved when $x_2 = 0$. On the other hand, when it is used as an adaptive variable, it improves the optimal worst-case significantly, and the optimal expected performance slightly. The optimal non-adaptive solution w.r.t. expected value is $\mathbf{x}^\star = [1, 0]$, and w.r.t. worst-case performance is $\mathbf{x}^\star = [x_1, 0]$, where x_1 can take any feasible value.

Confidence-Based and Target-Based

The robustness defined by the indicators I_c and I_q depends on the desired confidence and target. The effect of these parameters on the indicator value and the obtained robust solution is examined. Both the robust optimization problem in Equation (3.31) and the active robust optimization problem in Equation (3.10) are solved for all possible values of c and q between zero and one. The optimal solutions are found, to a precision of ± 0.005, using an enumeration, which is possible thanks to the low complexity of the objective function. The optimal adaptive and non-adaptive solutions, and their robustness values, are compared in the following.

Figure 3.10 depicts the optimal solutions for the confidence-based indicator with different confidence levels. From Figure 3.10(a) it can be observed that the optimal non-adaptive solution for every desired confidence level c is $\mathbf{x} = [1, 0]$. This solution is the same as the optimal non-adaptive solution for the expected value and the worst-case performance. The indicator values change with the confidence level from the best-case of $\phi = 1$ when $c = 0\%$ to the worst-case of $\phi = 0$, when $c = 100\%$.

In contrast, the optimal adaptive solution does change for different desired confidence levels c. When low confidence in the attained value is required, maximum performance



(a) Optimal solution in Decision space

(b) Indicator value (confidence in attaining the target)

Figure 3.11: Optimal solutions for the target-based indicator with different targets for the objective function.

should not be sacrificed for adaptability, and the optimal solution is $x^\star = 1$. As the desired confidence increases, adaptability is required to maintain higher function values, and therefore $x^\star$ decreases. In Figure 3.10(b) it can be seen that as the confidence level increases, the difference between the attainable value of the adaptive and the non-adaptive solutions increases.

The optimal solutions for the target-based indicator with different targets are shown in Figure 3.11. Figure 3.11(a) depicts the optimal adaptive and non-adaptive solutions, and Figure 3.11(b) depicts the confidence level that every solution can attain for each target.

The optimal non-adaptive solution is the same solution as for the other robustness criteria, i.e., $\mathbf{x}^\star = [1, 0]$. The confidence in attaining every target is maximized with the largest cosine amplitude, and when the cosine argument has a uniform distribution, centred at zero. It can be seen in Figure 3.11 that the optimal confidence level for every target decreases as the target increases, towards zero confidence when the target is $q = 1$.

Adaptability can ensure 100% confidence in attaining low function values (i.e., $q \leq 0.36$). In this range, as can be seen in Figure 3.11(a), a range of solutions can attain the optimal indicator level of 100% confidence. The boundaries of the optimal solutions $x^\star$ are marked with a dashed line in Figure 3.11(a). For example, 100% confidence of attaining the target $q = 0.25$ can be achieved with the solutions $0.27 \leq x^\star \leq 0.79$.

Since the same level of robustness can be achieved either by improved hardware (i.e., larger values of x) or enhanced adaptability (for small values of x), the preferred solution can be chosen based on additional criteria that were not specified in the original problem formulation. For example, if the cost of the product is relative to the value of x, the lower bound of the optimum solutions may be preferred. In this case, the optimal adaptive solution is superior to the non-adaptive counterpart not only for the original robustness metric, but also for the cost criterion. The cost of the non-adaptive solution is higher as it requires the maximum value of x (denoted x_1 in the non-adaptive problem formulation in Equation (3.31)).

For targets greater than $q = 0.36$, the amplitude must be increased, which means reducing the solution's adaptability. When the target for the function value goes above $q = 0.84$, the optimal adaptive solution is exactly the same as the non-adaptive one, i.e. $x^\star = 1$.

3.6 Sampled Representation of the Uncertainties

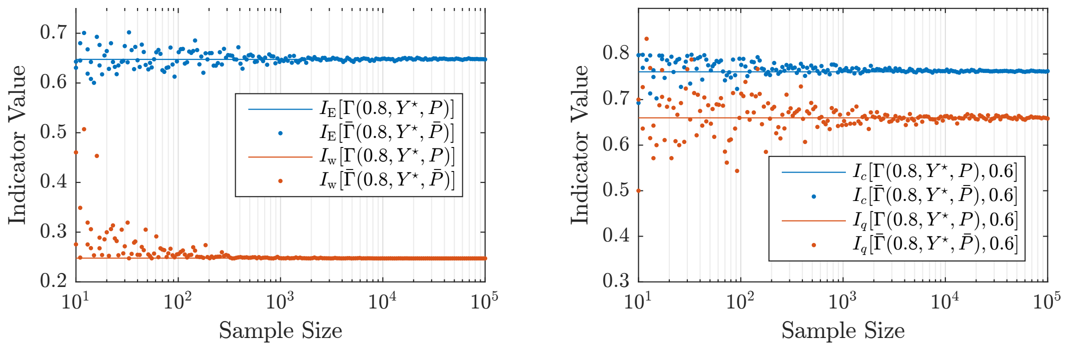
Commonly in real-world problems, the distribution of the uncertain objective function Γ cannot be analytically derived for the following reasons: i) the distribution of the uncertain parameters is not known and needs to be derived from empirical data, and/or ii) it is not feasible to analytically propagate the uncertainties to form the uncertain objective function.

An approximation of Γ, denoted $\bar{\Gamma}$, can be obtained using uncertainty quantification (UQ) methods. Monte-Carlo sampling is the most reliable UQ method and often used as a reference for the 'true' uncertainty. However, it requires a large number of function evaluations to converge (Poles and Lovison, 2009). An approximation of the uncertain objective function Γ can be obtained from repeated evaluations of the function using independent samples from the uncertain parameters.

The random variate $\mathbf{P}$ is represented by a set with k independent samples $\overline{P} = \left\{\mathbf{p}^1, \ldots, \mathbf{p}^k\right\}$ drawn from $\mathbf{P}$. The approximated variate of the objective function then becomes:

$$\bar{\Gamma}(\mathbf{x}, \mathbf{y}^\star, \bar{\mathbf{P}}) = \left\{\gamma(\mathbf{x}, \mathbf{y}^\star, \mathbf{p}^1), \ldots, \gamma\left(\mathbf{x}, \mathbf{y}^\star, \mathbf{p}^k\right)\right\}. \tag{3.32}$$

The example AROP that is used in this Chapter is composed of a very simple objective function, and therefore it can be evaluated many times without a substantial computational cost. In real-world optimization problems, the objective functions are typically predicted using complicated simulation models. A single function evaluation may take up to several hours or even days. When this is the case, Monte-Carlo methods are not feasible to evaluate the performance of every candidate solution.



(a) Expected value and worst-case indicators.

(b) Confidence-based and target-based indicators.

Figure 3.12: Convergence of indicator values of sampled uncertain function to the true indicator value, for different numbers of samples. All indicators are calculated for $x = 0.8$.

A variety of other UQ methods exist that require a smaller set $\bar{\mathbf{P}}$ for a reliable approximation of Γ through $\bar{\Gamma}$. Some examples are polynomial chaos (Wiener, 1938; Poles and Lovison, 2009), evidence theory (Shafer, 1976; Vasile et al., 2012), stochastic collocation (Eldred et al., 2011) and quadratic interpolation (Paenke et al., 2006). An alternative suggested by Branke et al. (2017) is to conduct a large number of samples, but to reduce the simulation runtime, which results in a less accurate approximation. The simulation runtime can then be progressively increased as the optimization progresses towards the robust solution.

Despite the computational issues associated with Monte-Carlo methods, they are used throughout this study to present and discuss the concepts of the Active Robustness framework. When sampling methods are required to approximate the uncertainty factors for problems with expensive objective functions, other UQ methods should be used.

In the following, the relation between the number of samples and the estimation of robustness is examined. The uncertain parameters are sampled into a set $\bar{P}$ and the robustness indicators are computed using their discrete definitions provided in Section 2.4.

Figure 3.12 depicts the convergence of various robustness indicators based on $\bar{\Gamma}$ with different sample size k. A solution $x = 0.8$ is evaluated for the expected-value, worst-case, confidence-based indicator with $c = 60\%$ and target-based indicator with $q = 0.6$, where $P \sim \mathrm{U}(-1, 1)$. The analytic indicator values are depicted as solid lines, and the approximated indicator for every sample size is depicted as a single dot. In agreement with the *law of large numbers*, as the sample size grows, the approximated indicator value converges towards the analytic value.

The convergence rate may differ from one indicator to another (even when calculating all indicators for the same sampled approximation $\bar{\Gamma}$). Assuming convergence is considered satisfactory when the approximated indicator value is within 2% of the true value, a comparison between the convergence rate for different indicators can be made. For the results shown in Figure 3.12, I_w converges with the smallest set of 322 samples, then I_E, I_c and I_q with 445, 850 and 5,416 samples, respectively. When a sampled representation of the uncertainty is used to solve an AROP, it is recommended to first find the smallest sample size that provides a reliable approximation of the robustness indicator used.

3.7 AROPs With Various Sources of Uncertainty

The active robust optimization problem in Definition 3.16 is formulated to cope with uncertain environmental conditions (Type A uncertainty). Adaptability can be used to compensate for other types of uncertainties as well. This section demonstrates how adaptability can be used when the realization of the design might differ from the nominal value (Type B uncertainty), and when the inaccuracy of the simulation method can be treated as an uncertain function evaluation (Type C uncertainty).

3.7.1 Type B Uncertainty

Consider an uncertainty over the value of the decision variable x in the optimization problem of Equations (3.13)–(3.17). This uncertainty can be conceived as inaccuracies in manufacturing processes, under acceptable tolerances. In engineering design, continuous variables cannot be manufactured to an exact dimension, and every desired dimension must be accompanied with an allowable tolerance. The manufactured product is acceptable as long as the actual dimension is within the tolerance. There is no guarantee for the distribution of the dimension for a batch of products, although some assumptions can be made according to the manufacturing process.

The realisation of x can take any value from the random variable X that is defined over the interval of accepted x values. Since the allowable tolerance and the manufacturing process are defined by the designer, he/she has some control over the boundaries and distribution of the random variable X. This control does not exist for Type A uncertainties, where the environmental parameters are treated as random numbers.

Manufacturing tolerances can be specified in many ways. In the following example, a symmetric tolerance of $\pm 20\%$ of the nominal value is used. A uniform distribution within the acceptable interval is assumed:

$$X \sim \mathrm{U}(0.8x_{nom}, 1.2x_{nom})\,, \tag{3.33}$$

where x_{nom} is the nominal value specified for x.

Assuming the parameter value p is deterministic and positive, an AROP is formulated:

$$\begin{aligned} &\max_{x_{nom}\in\mathcal{X}} I[\Gamma(X,Y^\star,p)]\,,\\ \text{where:}\quad &\Gamma(X,Y^\star,p) = X\cos\big(\tfrac{\pi}{2}(p-Y^\star)\big)\,,\\ &Y^\star = \operatorname*{argmax}_{y\in\mathcal{Y}(x)} \Gamma(X,y,p)\,,\\ &\mathcal{X} = \{x_{nom}|0 < X \le 1\}\,,\\ &\mathcal{Y}(x) = [x-1, 1-x]\,,\\ &X \sim \mathrm{U}(0.8x_{nom}, 1.2x_{nom})\,,\\ &p = 0.7. \end{aligned} \tag{3.34}$$

Note that the feasible domain $\mathcal{X}$ is smaller than the domain when there is no uncertainty over x. In Equation (3.34) all possible realisations of x must be smaller or equal to 1, and therefore the feasible values for x_{nom} are $0 < x_{nom} \le 0.833$. The adaptability of the solution $\mathcal{Y}(x)$ depends on the actual realisation of x and not on the requested nominal value. Hence, for every possible realisation of x there might be a different optimal configuration $y^\star$, and a one-to-one mapping exists between X and $Y^\star$.

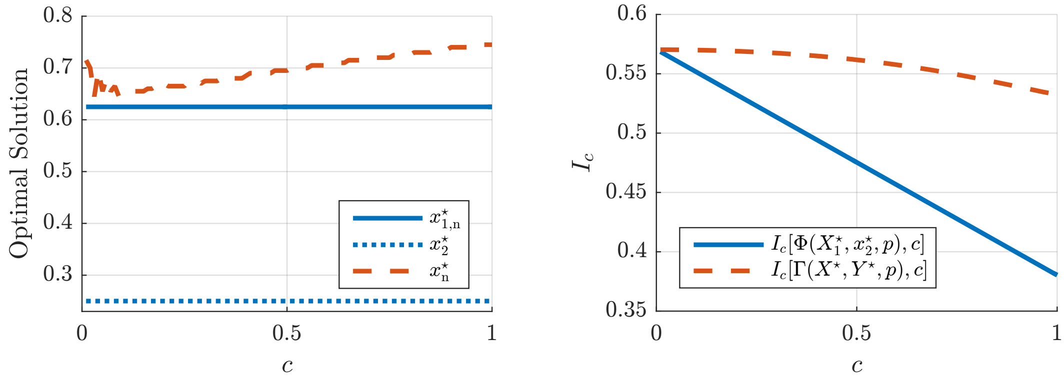
The optimal configuration is:

$$y^\star(x,p) = \begin{cases} p, & \text{for } x \le 1-p,\\ 1-x, & \text{for } x > 1-p, \end{cases} \tag{3.35}$$

and the optimal function value is:

$$\gamma(x,y^\star,p) = \begin{cases} x, & \text{for } x \le 1-p,\\ x\cos\big(\tfrac{\pi}{2}(p+x-1)\big)\,, & \text{for } x > 1-p. \end{cases} \tag{3.36}$$

The distribution function of Γ cannot be obtained analytically, since the inverse function $x(\gamma)$ does not exist. Since X follows a uniform distribution for every nominal value x_{nom}, it is easy to obtain the sampled approximation $\bar{\Gamma}\big(\bar{X},Y^\star,p\big)$ using Monte-Carlo sampling with $k = 5{,}000$ samples for every candidate solution. The indicator values for the sampled approximation of the uncertain objective function were calculated as explained in Section 3.6.



(a) Optimal solution in Decision space

(b) Indicator value

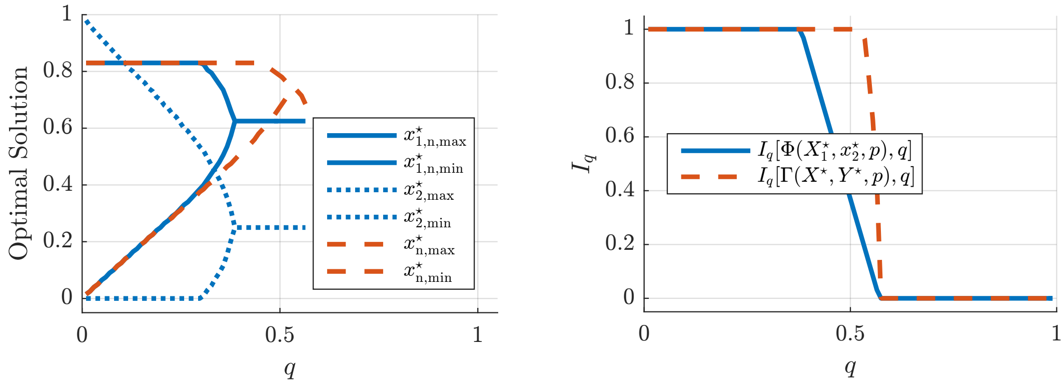
Figure 3.13: Optimal solutions for the confidence-based indicator with different desired confidence levels.

A non-adaptive solution with uncertainty of Type B is also considered. The robust optimization problem is similar to the one in Equation (3.31), but here the uncertainty over p is replaced with an uncertainty over x_1. Similar to the adaptive solution, x_1 follows a uniform distribution $X_1 \sim \mathrm{U}(0.8x_{1,nom}, 1.2x_{1,nom})$; x_2 is deterministic, but needs to be defined before the realisation of the solution, and therefore it is not adjustable.

The non-adaptive solution $\mathbf{x}^\star = \left[x^\star_{1,nom}, x^\star_2\right]$ is the solution to the following robust optimization problem:

$$
\begin{aligned}
& \max_{\mathbf{x}\in\mathbb{R}^2} I[\Phi(\mathbf{X}, p)]\,, \\
\text{subject to:}\quad & 0 < X_1 \leq 1, \\
& X_1 + |x_2| \leq 1, \\
\text{where:}\quad & \Phi(\mathbf{X}, p) = X_1 \cos\!\left(\tfrac{\pi}{2}(p - x_2)\right), \\
& X_1 \sim \mathrm{U}(0.8x_{1,nom}, 1.2x_{1,nom})\,, \\
& p = 0.7.
\end{aligned}
\tag{3.37}
$$

For a given candidate solution $\mathbf{x} = [x_{1,nom}, x_2]$, the PDF $f(x_1)$ is given, and it is straightforward to propagate it to the uncertain objective function Φ. Since both p and x_2 are deterministic, the term $\cos\!\left(\frac{\pi}{2}(p - x_2)\right)$ is a constant scalar for every realisation of x_1. Therefore, Φ follows the same distribution as X_1, simply scaled by the above scalar. In the example provided in Equation (3.37), it is a uniform distribution. The AROP in Equation (3.34) and the RO problem in Equation (3.37) are solved for two robustness indicators I_c and I_q.



(a) Optimal solution in Decision space

(b) Indicator value (confidence in attaining the target)

Figure 3.14: Optimal solutions for the target-based indicator with different targets for the objective function.

Confidence-Based. Figure 3.13 depicts the optimal adaptive and non-adaptive solutions for the confidence-based indicator with different confidence levels, c. The superiority of the adaptive solution over the non-adaptive one is evident from the figure. Both solutions achieve the same best-case performance (i.e., function value with 0% confidence), but as the required confidence level increases, the difference in the achieved performance grows significantly.

The maximum function value that can be achieved by either an adaptive or a non-adaptive solution is $\phi = 0.57$. If the solution is non-adaptive, this value is achieved for the best-case of the solution $\mathbf{x}^\star = [0.625, 0.25]$ (i.e., when $x_1 = 1.2x_{nom} = 0.75$). If the solution is adaptive, this value can be achieved by all solutions that have non-zero probability for $x = 0.75$. The nominal solution x_{nom} needs to be in the range $x^\star_{nom} \in [0.75/1.2, 0.75/0.8]$, that is, $x^\star_{nom} \in [0.625, 0.937]$. The curve for $x^\star_{nom}$ in Figure 3.13(a) does not appear as a smooth line because the maximum confidence can be achieved by a range of adaptive solutions for some confidence levels, c.

For higher confidence levels, an adaptive solution with nominal values larger than $x_{nom} = 0.625$ can maintain high function values while remaining feasible thanks to adaptation of y to the actual realisation of x. A non-adaptive solution must consider feasibility for all possible realisations x_1, and therefore solutions with $x_{1,\mathrm{n}} > 0.625$ cannot satisfy the constraint $x_2 < 0.25$. As a result, only lower values of x_1 can be identified as optimal, leading to poorer performance compared to the adaptive solution.

Target-Based. Figure 3.14 depicts the optimal adaptive and non-adaptive solutions for the target-based indicator with different targets, q. None of the solutions can achieve function values greater than 0.57, and therefore the indicator value for $q > 0.57$ is zero for both adaptive and non-adaptive solutions. For low targets, q, both types of solutions can assure 100% confidence in satisfying the target. A span of optimal solutions can provide this confidence. The ranges of optimal solutions are depicted in Figure 3.14(a) via their maximum and minimum values. For the non-adaptive optimal solutions, the maximum $x^\star_{1,\mathrm{n}}$ corresponds to the minimum $x^\star_2$, and vice versa.

The advantage of the adaptive solutions over the non-adaptive ones can be seen from Figure 3.14(b). While the confidence for targets larger than $q = 0.37$ drops for the optimal non-adaptive solutions, it remains 100% for the optimal adaptive solutions for all targets within $q \leq 0.52$. The optimal non-adaptive solution for the targets $0.37 \leq q \leq 0.57$ is the same optimal solution as for the confidence-based indicator $\mathbf{x}^\star = [0.625, 0.25]$.

3.7.2 Type C Uncertainty

Adaptability can be exploited to account for approximation errors in the simulation method. consider a property of the product that needs to be approximated by simulation. The property might not be an objective on its own, but may be used to calculate an objective. For example, a manufacturing line may be designed to produce at a desired rate ω'. The production rate depends on the product being manufactured (represented by a set of parameters $\mathbf{p}$), the machines used for production (the actual solution, $\mathbf{x}$) and the way the machines are operated (adjustable variables, $\mathbf{y}$). This can be denoted as $\omega(\mathbf{x}, \mathbf{y}, \mathbf{p})$. Since a desired rate is given, a quadratic objective function can be considered:

$$\gamma\big(\mathbf{x}, \mathbf{y}, \mathbf{p}, \omega'\big) = \big(\omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) - \omega'\big)^2 . \tag{3.38}$$

Now assume the production rate ω can be approximated by a simulation method. For a given set of inputs, the simulation produces a deterministic output, but in practice the actual production might differ from the simulated value. Depending on the fidelity of the simulation method, its output can be considered as a random variable with a prior based on the deterministic approximation. For example, the random production rate Ω may follow a normal distribution with mean, ω, and standard deviation, σ:

$$\Omega|\omega \sim \mathrm{N}(\omega, \sigma) . \tag{3.39}$$

Considering the uncertain function evaluation, and assuming perfect information over the other factors, the objective in Equation (3.38) becomes a random objective function:

$$\Gamma\big(\mathbf{x}, \mathbf{y}, \mathbf{p}, \omega'\big) = \big(\Omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) - \omega'\big)^2 . \tag{3.40}$$

For the uncertainty example in Equation (3.39), the uncertain function evaluation can be broken into the sum of a deterministic part and a random variable $U \sim \mathrm{N}(0, \sigma)$:[1]

$$\Omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) = \omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) + U. \tag{3.41}$$

The random objective function Γ then becomes

$$\Gamma\big(\mathbf{x}, \mathbf{y}, \mathbf{p}, U, \omega'\big) = \big(\omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) + U - \omega'\big)^2. \tag{3.42}$$

An active robust optimization problem can be formulated to minimize this random objective function:

$$\begin{aligned} & \min_{\mathbf{x}\in\mathcal{X}} \Gamma\big(\mathbf{x}, \mathbf{Y}^\star, \mathbf{p}, U, \omega'\big), \\ \text{where:} \quad & \mathbf{Y}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \Gamma\big(\mathbf{x}, \mathbf{y}, \mathbf{p}, U, \omega'\big), \\ & \Gamma\big(\mathbf{x}, \mathbf{y}, \mathbf{p}, U, \omega'\big) = \big(\omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) + U - \omega'\big)^2. \end{aligned} \tag{3.43}$$

The adjustable variables $\mathbf{y}$ are adapted to the optimal configuration to meet the required production rate, ω'. Feedback control needs to be incorporated to allow for optimal adaptation, based on the actual error between the estimated rate and the actual rate of production.

The variables U and ω' in Equation (3.43) cannot be controlled by the designer, and must be considered as additional parameters in the problem formulation. Therefore, an uncertain parameter vector $\mathbf{P}$ can be considered:

$$\mathbf{P} = \big[\mathbf{p}, U, \omega'\big], \tag{3.44}$$

and the AROP in Equation (3.43) can be rewritten as follows:

$$\begin{aligned} & \min_{\mathbf{x}\in\mathcal{X}} \Gamma(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}), \\ \text{where:} \quad & \mathbf{Y}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P}), \\ & \mathbf{P} = \big[\mathbf{p}, U, \omega'\big], \\ & \Gamma(\mathbf{x}, \mathbf{y}, \mathbf{P}) = \big(\omega(\mathbf{x}, \mathbf{y}, \mathbf{p}) + U - \omega'\big)^2. \end{aligned} \tag{3.45}$$

The above AROP has exactly the same structure as the AROP defined in Definition 3.16 for Type A uncertainties. Hence, it can be concluded that as long as the uncertainties

[1] Note that the structure $\omega + U$ and the assumption of a normal distribution for U are specific for the given example. Generally, other distributions for U are possible, as well as other relations between the deterministic and the random parts of the evaluation function (e.g., $\omega \cdot U$).

over the simulation methods can be elicited, Type A and Type C uncertainties can be treated using the same tools.

The following example is provided to support this statement. Consider the production rate, $\omega(x, y, p)$, can be estimated by the function and constraints used so far in this section:

$$\begin{aligned} & \omega(x,y,p) = x\cos\left(\tfrac{\pi}{2}(p-y)\right), \\ \text{subject to:}\quad & 0 < x \leq 1, \\ & x + |y| \leq 1. \end{aligned} \tag{3.46}$$

The AROP in Equation (3.45) then becomes:

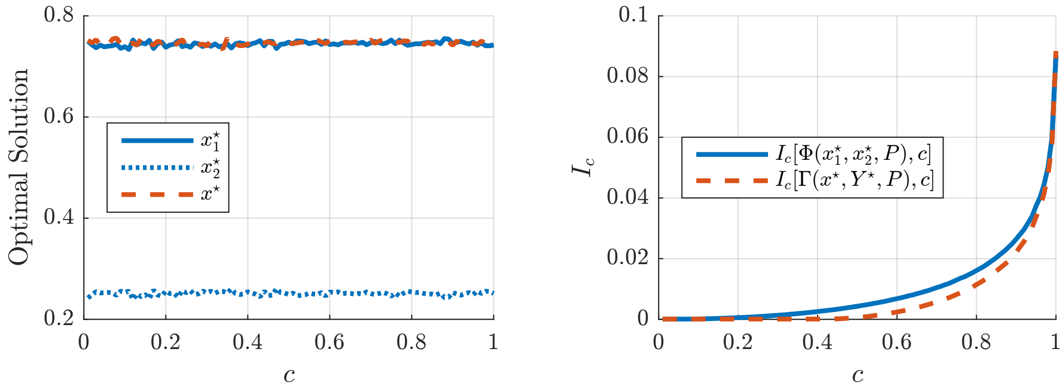
$$\begin{aligned} & \min_{x\in\mathcal{X}} \Gamma(x, Y^\star, \mathbf{P}), \\ \text{where:}\quad & Y^\star = \operatorname*{argmin}_{y\in\mathcal{Y}(x)} \Gamma(x, y, \mathbf{P}), \\ & \mathbf{P} = \left[p, U, \omega'\right], \\ & \Gamma(x, y, \mathbf{P}) = \left(x\cos\left(\tfrac{\pi}{2}(p-y)\right) + U - \omega'\right)^2, \\ & U \sim \mathrm{N}(0,\sigma), \\ & \mathcal{X} = (0, 1], \\ & \mathcal{Y}(x) = [x-1, 1-x]. \end{aligned} \tag{3.47}$$

A similar problem is defined for a non-adaptive solution that needs to set its y variable regardless of the actual performance of ω:

$$\begin{aligned} & \min_{\mathbf{x}\in\mathbb{R}^2} \Phi(\mathbf{x}, \mathbf{P}), \\ \text{subject to:}\quad & 0 < x_1 \leq 1, \\ & x_1 + |x_2| \leq 1, \\ \text{where:}\quad & \mathbf{P} = \left[p, U, \omega'\right], \\ & \Phi(\mathbf{x}, \mathbf{P}) = \left(x_1\cos\left(\tfrac{\pi}{2}(p-x_2)\right) + U - \omega'\right)^2, \\ & U \sim \mathrm{N}(0,\sigma). \end{aligned} \tag{3.48}$$

Both optimization problems were solved using an enumerative approach, and a sample-based representation of U. The following values were used for the parameters: $p = 0.7,\ \omega' = 0.6,\ \sigma = 0.1$. Exactly the same methods were used to solve this problem as in Section 3.6 where the source of uncertainty was in the parameters (Type A).

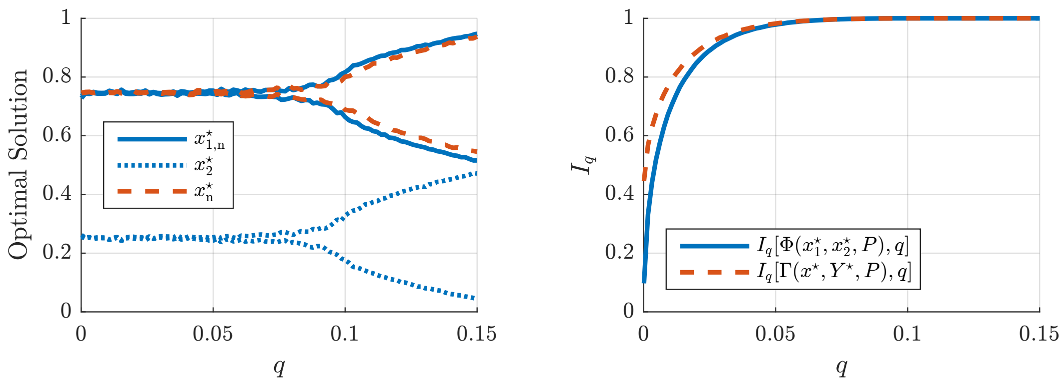
The results, depicted in Figures 3.15 and 3.16 for the I_c and I_q indicators, respectively, show that adaptation improves the solution's performance for both robustness indicators



(a) Optimal solution in Decision space

(b) Indicator value

Figure 3.15: Optimal solutions for the confidence-based indicator for an uncertain evaluation function.



(a) Optimal solution in Decision space

(b) Indicator value

Figure 3.16: Optimal solutions for the target-based indicator for an uncertain evaluation function.

for some values of c and q. The optimal adaptive solution $x^\star$ is identical to the first variable of the optimal non-adaptive solution $x_1^\star$. The oscillations in the optimal solutions' values are a result of approximation of the solution by Monte-Carlo methods, and the discretisation of the decision space when searching for the optimal solution, $x^\star$, or, $\mathbf{x}^\star$, and the optimal configuration, $y^\star$.

3.8 Summary

The active robust optimization problem for optimizing adaptive products was formulated in this chapter. The formulation takes into account the ability of the adaptive product to change some adjustable properties in response to different realisations of parameters that are considered as random during the design period.

The problem is a nested optimization problem – the evaluation of each candidate solution is conducted by solving a set of optimization problems to find the optimal configurations for different scenarios of the uncertainties. The performance with the optimal configuration is mapped against each scenario to construct a distribution of the random performance variable. Once the performance of the adaptive product can be described as a random variable, conventional methods for robust optimization can be used, through the use of robustness indicators.

In robust optimization, the performance in nominal cases of the uncertainties is commonly sacrificed for robustness to extreme cases. Adaptability can handle those extreme cases, and allow for better overall performance. The superiority of an adaptive design was demonstrated over an equivalent non-adaptive design for several variations of the problem formulation and different robustness metrics. The advantage of the adaptive design is rooted in the ability to decide the values for some of its variables after the uncertainty has been realised, in contrast to conventional robust optimization, where all variables are set during the design stage.

In order to analyse the properties of this new class of optimization problems, a very simple analytic function was used to formulate an AROP with minimum dimensionality. It consisted of a single design variable, a single adjustable variable, a single uncertain parameter and a single objective function. The low dimensionality of the problem and the simplicity of the objective function made it possible to propagate the uncertainties from the random parameter to the objective and to find the analytic solution to the AROP for several robustness criteria.

Despite the low dimensionality of the example AROP, the uncertain objective function could not be calculated analytically for some cases, and Monte-Carlo sampling had to be used to estimate it. Moreover, an enumeration was used to find the optimal robust solution when it could not be found analytically. These computationally expensive methods could be employed for such a simple function, but real optimization problems are never this simple. When the difficulty of the problem increases, more efficient methods for optimization and uncertainty approximation have to be used. Some examples for such methods are given in Chapter 5, where evolutionary algorithms are used to solve the case studies.

The basic AROP, presented in this chapter, is composed of a single objective function. Therefore, a one-to-one mapping between every scenario of the uncertainties and the optimal configuration exists. In the next Chapter the problem is extended to deal with situations where more than one objective exists. When this is the case, there might be a set of optimal configurations for every uncertain scenario, providing different trade-off between the objectives. The complexities of the problem as well as several approaches to solve it are presented.

Chapter 4

Active Robust Multi-Objective Optimization

4.1 Introduction

In this chapter a wider definition of the AROP is provided. It is extended to include problems that involve multiple objectives, which are very common in real-world design. Chapter 3 introduced AROPs with one objective function that may be sensitive to various types of uncertainties. Whenever the uncertain conditions change, a single-objective optimization problem needs to be solved in order to find the new optimal configuration. Therefore, a one-to-one mapping between the realisation of the uncertain variables and the optimal configuration exists.

The problem becomes much more complicated when more than one objective can be simultaneously improved by adaptation. Since the solution to a multi-objective optimization problem is usually a set of Pareto-optimal solutions, a set of optimal configurations can exist for every realisation of the uncertainties. This poses many challenges in evaluating and comparing candidate solutions.

For example, the brightness of a smartphone screen can be adjusted by the user according to the current lighting conditions. A brighter screen provides better visibility of the data displayed, but consumes more energy. Each brightness level provides a different trade-off between power consumption and the clarity of displayed data. For every lighting condition, a different set of trade-offs exists. When choosing the components of the smartphone, these need to be taken into account.

The Chapter is organised as follows: In Section 4.2 the Active Robust Multi-Objective Optimization Problem is defined and its notion of optimality is discussed.

An analytic example, based on the single-objective AROP of Chapter 3, is described in Section 4.3. The problem features and the relations between the objective functions are analysed, as well as the manner in which they are affected by uncertainties and adaptation.

A candidate solution to an ARMOP has a set of optimal configurations for every possible realisation of the uncertainties. As a result, its performance can be described as a variate of Pareto-optimal frontiers. In order to optimize adaptive products for a multi-objective problem, a means to evaluate and compare candidate solutions is required. Despite the recent progress in the field of multi-objective optimization under uncertainties, quality indicators for variate of sets cannot be found in the existing literature. In Section 4.4, the requirements from a robustness indicator for an ARMOP are characterised, and a number of strategies for fitness assignment and comparison between candidate solutions are suggested. Robustness indicators, that are based on these strategies are introduced.

Section 4.5 demonstrates how the suggested robustness indicators can be incorporated into search heuristics. High-level descriptions of evolutionary algorithms are provided. Each algorithm uses a different solution approach, based on the manner in which robustness is defined for the ARMOP.

To conclude the chapter, a review of the suggested methods for solving ARMOPs is presented in Section 4.6. It includes some guidelines for selecting the most suitable approach based on issues such as computing resources and flexibility for *a posteriori* decision-making. Section 4.7 summarises the chapter.

4.2 Problem Formulation

This section extends the definition of the Active Robust Optimization Problem in Section 3.3 to optimization problems with multiple objectives of type γ (i.e., adaptable objectives, see definition in Section 2.3.1).

As discussed in Section 2.1.2, uncertainties over the objective values of a candidate design can originate from the parameters $\mathbf{p}$ (Type A), the decision variables $\mathbf{x}$ and $\mathbf{y}$ (Type B) and/or the objective functions themselves (Type C). Without loss of generality, the aspects of the ARMOP are demonstrated using Type A uncertainties, i.e., uncertain environmental parameters. Other sources of uncertainties can be treated in a similar fashion, as demonstrated for the single-objective case in Section 3.7.

Definition 4.1 (Active Robust Multi-Objective Optimization Problem)**.** *Let $\mathbf{x} \in \mathcal{X}$ be a vector of design variables within a feasible domain $\mathcal{X}$, describing a changeable design. Let $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ be a vector of adjustable variables of the candidate design $\mathbf{x}$, where $\mathcal{Y}(\mathbf{x})$ is the adaptability of the design $\mathbf{x}$. Let $\mathbf{p}$ be a vector of uncertain environmental parameters,* *described by a vector variate* $\mathbf{P}$*. Let* $\boldsymbol{\gamma}(\mathbf{x}, \mathbf{y}, \mathbf{p})$ *be a vector of* $n_\gamma > 1$ *adaptable objective functions, and let* $\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{y}, \mathbf{P})$ *be a vector variate corresponding to the variate* $\mathbf{P}$*. An active robust multi-objective optimization problem is the following:*

$$\min_{\mathbf{x} \in \mathcal{X}} \underline{\boldsymbol{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}^\star}, \mathbf{P}) \,, \tag{4.1}$$

$$\textit{where: } \underline{\mathbf{Y}^\star} = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{y}, \mathbf{P}) \,. \tag{4.2}$$

The underline notation is used to distinguish a set of vectors from a single vector. Since the solution to (4.2) is a set of optimal configurations for every realisation of the uncertainties, $\underline{\mathbf{Y}^\star}$ is a variate of Pareto-optimal sets. As a result, $\underline{\boldsymbol{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}^\star}, \mathbf{P})$ is a variate of Pareto-optimal frontiers.

4.3 Analytic Example

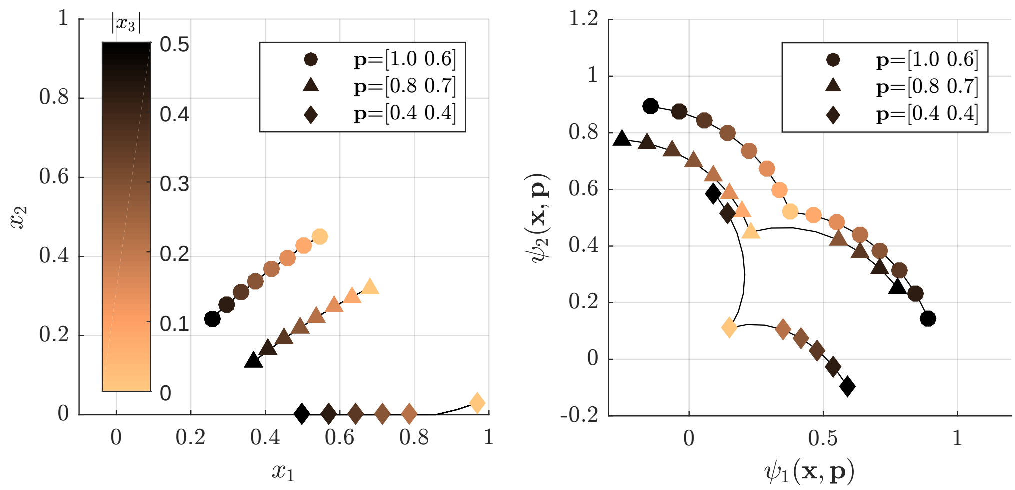
A simple bi-objective optimization problem is used in this section to demonstrate the unique properties of ARMOPs. Consider the following functions that need to be simultaneously minimized:

$$\begin{aligned}
\underset{x}{\text{minimize:}} \quad & \gamma_1(x, \mathbf{y}, \mathbf{p}) = r(x, y_1, p_1) \cdot \cos(\theta(y_2, p_2)) \,, \\
& \gamma_2(x, \mathbf{y}, \mathbf{p}) = r(x, y_1, p_1) \cdot \sin(\theta(y_2, p_2)) \,, \\
\text{where:} \quad & r(x, y_1, p_1) = 1 - x \cos\big(\tfrac{\pi}{2}(p_1 - y_1)\big) \,, \\
& \theta(y_2, p_2) = \tfrac{\pi}{2}(p_2 + y_2) \,, \\
\text{subject to:} \quad & 0 < x \leq 1, \\
& |y_2| \leq \tfrac{1}{2}, \\
& x + |y_1| + |y_2| \leq 1.
\end{aligned} \tag{4.3}$$

The functions are composed of a distance term $r(x, y_1, p_1)$ and a direction term $\theta(y_2, p_2)$. For values of θ between zero and $\pi/2$ both sine and cosine terms are positive. Therefore, an increase in r increases both γ_1 and γ_2, while an increase in θ increases γ_2 and decreases γ_1. The distance term, $r(x, y_1, p_1)$, is based on the objective function used in Chapter 3. The direction term, θ, determines the ratio between the objectives.

4.3.1 Functions Analysis

Before the problem in Equation (4.3) is framed as an ARMOP, let us take a step back and assume no uncertainties exist over any part of the problem formulation. When this



(a) Optimal solution in Decision space (b) Optimal solutions in Objective space

Figure 4.1: Optimal solutions for the problem in Equations (4.4)-(4.9) for different parameter settings.

is the case, the parameters p_1 and p_2 are constant terms, and all decision variables can be considered as type x, i.e., they only need to be found once. The problem in (4.3) can then be posed as a simple multi-objective optimization problem:

$$\underset{\mathbf{x}}{\text{minimize:}} \quad \psi_1(\mathbf{x},\mathbf{p}) = r(x_1,x_2,p_1)\cdot\cos(\theta(x_3,p_2))\,, \tag{4.4}$$

$$\psi_2(\mathbf{x},\mathbf{p}) = r(x_1,x_2,p_1)\cdot\sin(\theta(x_3,p_2))\,, \tag{4.5}$$

$$\text{where:} \quad r(x_1,x_2,p_1) = 1 - x_1\cos\left(\tfrac{\pi}{2}(p_1 - x_2)\right), \tag{4.6}$$

$$\theta(x_3,p_2) = \tfrac{\pi}{2}(p_2 + x_3)\,, \tag{4.7}$$

$$\text{subject to:} \quad 0 < x_1 \le 1, \quad |x_2| \le 1, \quad |x_3| \le \tfrac{1}{2}, \tag{4.8}$$

$$\sum_{i=1}^{3}|x_i| \le 1. \tag{4.9}$$

Figure 4.1 depicts the solution to Equations (4.4)-(4.9) in Decision and Objective spaces, for three combinations of the parameters. For each parameter setting, the problem was solved in three stages:

1. Generate a discrete set $\underline{x_3}$ spanning the feasible range of x_3.
2. For each variable in $\underline{x_3}$:
   (a) Calculate the direction $\theta(x_3,p_2)$.
   (b) Minimize the distance function $r(x_1,x_2,p_1)$.
3. Remove dominated solutions.

The markers in Figure 4.1 depict the non-dominated solutions for each parameter setting, and the connecting lines show the optimal solutions for all values of θ, including dominated solutions. For every direction θ, the optimization in Stage 2b follows the procedures described for the single-objective problem in Chapter 3.

The value of $\theta(x_3, p_2)$ defines the ratio between the objectives. The 'neutral' ratio is defined by the value of the environmental parameter p_2. If the decision-maker is not satisfied with this ratio and prefers a different trade-off between the objectives, the value of x_3 can be increased or decreased accordingly. The colour of the markers relates to the magnitude of x_3, i.e., to the amount of deviation from the 'neutral' direction. Due to the constraint in Equation (4.9), a smaller magnitude of x_3 allows for a smaller value of $r(x_1, x_2, p_1)$. As a result, the Pareto frontiers in Figure 4.1(b) appear as arcs that are pulled towards their centre at the 'neutral' direction.

The curves of Pareto sets in Figure 4.1(a) depend on the value of p_1[1] As demonstrated in Chapter 3, higher values of p_1 require higher ratios of x_2/x_1 in order to remain optimal. Additionally, as the value of p_1 decreases, the problem becomes 'easier', and the arcs can be brought closer to the origin and improve performance in both objectives.

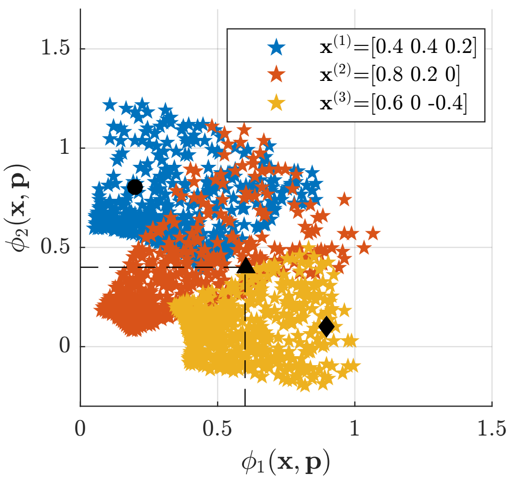
4.3.2 Introducing Uncertainties

Now that the properties of the objective functions are captured, this section examines how uncertainties over the parameters affect the problem. Adaptability is not considered in this section, which means that all decision variables are of type x, similar to Section 4.3.1. Assuming the uncertain parameters can be described by the vector variate $\mathbf{P}$, the problem in (4.3) can be posed as the following robust multi-objective optimization problem (RMOP):

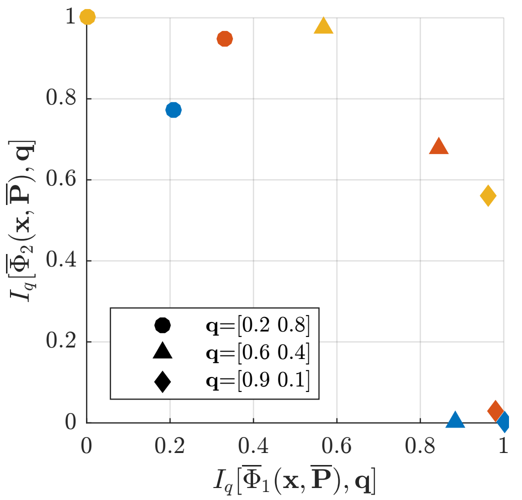
$$
\begin{aligned}
\underset{\mathbf{x}}{\text{minimize:}} \quad & \Phi_1(\mathbf{x}, \mathbf{P}) = R(x_1, x_2, P_1) \cdot \cos(\Theta(x_3, P_2))\,, \\
& \Phi_2(\mathbf{x}, \mathbf{P}) = R(x_1, x_2, P_1) \cdot \sin(\Theta(x_3, P_2))\,, \\
\text{where:} \quad & R(x_1, x_2, P_1) = 1 - x_1 \cos\!\big(\tfrac{\pi}{2}(P_1 - x_2)\big)\,, \\
& \Theta(x_3, P_2) = \tfrac{\pi}{2}(P_2 + x_3)\,, \\
\text{subject to:} \quad & 0 < x_1 \le 1, \quad |x_2| \le 1, \quad |x_3| \le \tfrac{1}{2}, \\
& \sum_{i=1}^{3} |x_i| \le 1.
\end{aligned}
\tag{4.10}
$$

Note that the uncertainty over the parameters propagates to the distance and direction terms, and eventually to the objective functions. For a given candidate solution $\mathbf{x}$, every

[1] However, the dominance relations between the solutions on the curve depend on p_2 as well.



(a) Objective values for each sample of $\mathbf{P}$. The goal vectors are shown with black markers.



(b) $\mathbf{I_q}$ indicator values for the three candidate solutions.

Figure 4.2: Performance of three candidate solutions when both objectives depend on uncertain parameters.

scenario $\mathbf{p}$ is mapped to an objective vector $\boldsymbol{\phi}$. The multivariate $\boldsymbol{\Phi}$ is the image of all scenarios of $\mathbf{P}$.

To illustrate the distribution of the objective functions, consider both parameters follow uniform distributions: $P_1 \sim \mathrm{U}(-1, 1)$ and $P_2 \sim \mathrm{U}(0.25, 0.75)$. Figure 4.2(a) depicts a sampled representation of the objective functions $\overline{\boldsymbol{\Phi}}(\mathbf{x}, \overline{\mathbf{P}})$ for three candidate solutions, using a set of $k = 500$ samples for $\overline{\mathbf{P}}$. As expected from the relation between θ and p_2, the random direction term Θ is uniformly distributed over an interval of $\pi/4$ radians. The lower and upper bounds depend on the variable x_3. Due to the cosine term in the distance function, R is right-skewed, where the lower bound depends on the value of x_1 and the upper bound on both x_1 and x_2.

In order to assess the random performance vector $\boldsymbol{\Phi} = \left[\Phi_1, \ldots, \Phi_{n_\phi}\right]$ for a candidate solution $\mathbf{x}$, robustness indicators need to be used. There are two possible approaches for using robustness indicators in RMOPs:

1. Indicators that quantify the multivariate $\boldsymbol{\Phi}$ with a single scalar value $I[\boldsymbol{\Phi}]$.

2. Using a separate indicator for each of the marginal distributions:

$$\boldsymbol{I}[\boldsymbol{\Phi}] = \left[I[\Phi_1], \ldots, I\left[\Phi_{n_\phi}\right]\right]. \tag{4.11}$$

When the first approach is used, the objective functions of the bi-objective problem in Equation (4.10) can be transformed into a single objective:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathcal{X}}{\text{optimize}}\, I[\boldsymbol{\Phi}(\mathbf{x},\mathbf{P})]\,, \\ \text{where:}\quad &\boldsymbol{\Phi}(\mathbf{x},\mathbf{P}) = [\Phi_1(\mathbf{x},\mathbf{P})\,,\Phi_2(\mathbf{x},\mathbf{P})]\,. \end{aligned} \tag{4.12}$$

When the second approach is used, it can be transformed into the following bi-objective problem:

$$\begin{aligned} &\underset{\mathbf{x}\in\mathcal{X}}{\text{optimize}}\, \boldsymbol{I}[\boldsymbol{\Phi}(\mathbf{x},\mathbf{P})]\,, \\ \text{where:}\quad &\boldsymbol{I}[\boldsymbol{\Phi}(\mathbf{x},\mathbf{P})] = [I[\Phi_1(\mathbf{x},\mathbf{P})]\,, I[\Phi_2(\mathbf{x},\mathbf{P})]]\,. \end{aligned} \tag{4.13}$$

To illustrate the difference between the two approaches, consider the problem in Equation (4.10), where a goal vector $\mathbf{q} = [0.6, 0.4]$ exists for the objective functions. The threshold indicator I_q can be used in two ways. On the first, the confidence in the random objective vector to satisfy the goal vector $\mathbf{q}$ is interpreted as

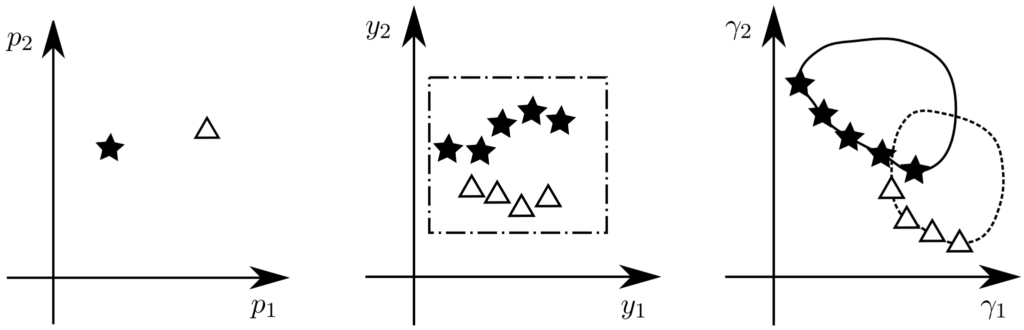
$$I_q[\boldsymbol{\Phi}(\mathbf{x},\mathbf{P})] = \Pr(\boldsymbol{\Phi}(\mathbf{x},\mathbf{P}) \preceq \mathbf{q})\,, \tag{4.14}$$

where $\mathbf{a} \preceq \mathbf{b}$ denotes that vector $\mathbf{a}$ weakly dominates vector $\mathbf{b}$. The dotted region at the bottom left corner of Figure 4.2(a) include all the samples that dominate the goal vector (which is marked with a black triangle). It can be seen in the figure that $\mathbf{x}^{(1)}$ does not satisfy the goals for any sampled scenario of the uncertainties and therefore $I_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(1)}, \overline{\mathbf{P}}\big)\big] = 0$. $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(3)}$ do satisfy the goal for some of the sampled scenarios, and their indicator values are $I_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(2)}, \overline{\mathbf{P}}\big)\big] = 0.61$ and $I_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(3)}, \overline{\mathbf{P}}\big)\big] = 0.53$.

It is worth mentioning that using the above robustness indicator as a single measure to assess candidate solutions goes against the nature of MOO. While the aspiration in MOO is to find a set of solutions that offer a wide range for *a posteriori* decision-making, this robustness measure forces the decision-maker to set his/her preference prior to the optimization process. At this stage the DM might not be aware of the possible performance, and may find it difficult to set a proper goal.

It might be beneficial to use this type of scalar robustness metric in a decomposition-based optimization framework. A set of goal vectors $\underline{\mathbf{q}}$ can be used to decompose the MOP into multiple single-objective problems, where each sub-problem targets a different trade-off between the objectives.

The threshold indicator can also be used for the second approach to assess the marginal distributions Φ_1 and Φ_2. For the above goal vector $\mathbf{q} = [0.6, 0.4]$, the vector



(a) Uncertain parameter space (b) Adjustable design space (c) Objective space

Figure 4.3: Optimal configurations of an adaptive solution for two scenarios of the uncertain parameters.

robustness indicator $\boldsymbol{I}_q$ is defined:

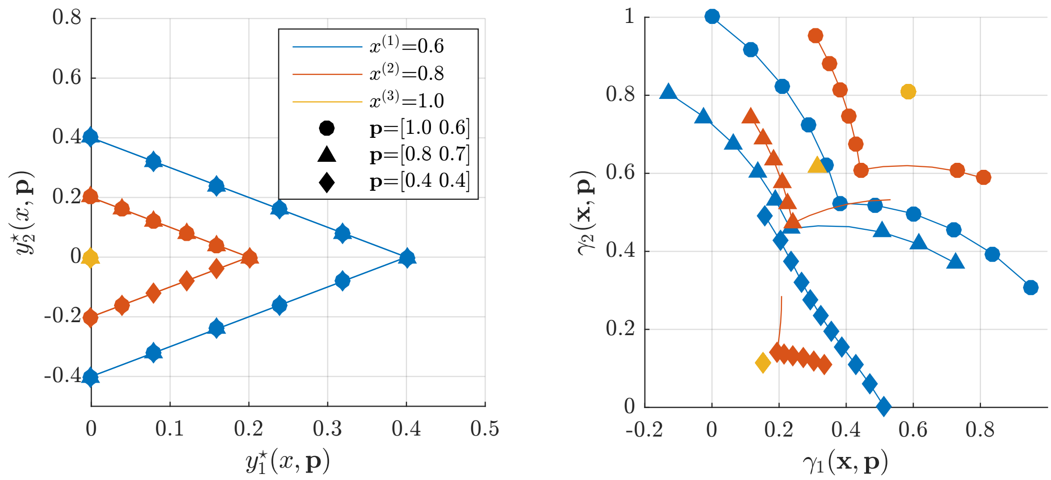
$$\boldsymbol{I}_q[\boldsymbol{\Phi}(\mathbf{x},\mathbf{P})] = [\Pr(\Phi_1(\mathbf{x},\mathbf{P}) \leq 0.6)\,, \Pr(\Phi_2(\mathbf{x},\mathbf{P}) \leq 0.4)]\,. \tag{4.15}$$

The vector indicator values of the three candidate solutions in Figure 4.2 are $\boldsymbol{I}_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(1)},\overline{\mathbf{P}}\big)\big] = [0.86, 0]$, $\boldsymbol{I}_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(2)},\overline{\mathbf{P}}\big)\big] = [0.83, 0.63]$ and $\boldsymbol{I}_q\big[\overline{\boldsymbol{\Phi}}\big(\mathbf{x}^{(3)},\overline{\mathbf{P}}\big)\big] = [0.53, 0.98]$. This approach, of using the marginal objective distributions, is the most common in existing studies on robust multi-objective optimization (see Section 2.4.2).

4.3.3 Introducing Adaptability

Now it is time to return to the definition of the active robust multi-objective optimization problem in Equations (4.1)–(4.2). While every scenario of the uncertain parameters in a conventional RMOP is associated with a single performance vector, adaptive solutions can offer a trade-off between the objectives whenever the parameters change. This is denoted with the underline notation for $\underline{\boldsymbol{\Gamma}}$ and $\underline{\mathbf{Y}^{\star}}$.

Figure 4.3 illustrates the performance of a candidate solution $\mathbf{x}$ for a hypothetical bi-objective ARMOP. Two scenarios of $\mathbf{P}$ are depicted in Figure 4.3(a). The configuration space, depicted in Figure 4.3(b) as the area bounded by the broken line, depends on the solution $\mathbf{x}$, and therefore it does not change from one scenario of $\mathbf{P}$ to another. However, for every scenario $\mathbf{p}$, a different mapping exists from configuration space to objective space, which is depicted in Figure 4.3(c). All possible objective values in Figure 4.3(c) are bounded by the solid and dashed contours for the star and triangle scenarios, respectively. The Pareto frontiers and corresponding Pareto sets $\underline{\mathbf{y}^{\star}}$ that minimize $\boldsymbol{\gamma}(\mathbf{x},\mathbf{y},\mathbf{p})$ are shown for both scenarios. Note that $\underline{\mathbf{y}^{\star}}$ is a set of the Pareto-optimal configurations and not a single optimal configuration, as in single-objective AROPs.



(a) Optimal configurations - can be described for all parameters by the line $y_1 + |y_2| = 1 - x$

(b) Objective values for optimal configurations

Figure 4.4: Pareto-optimal configurations for three solutions under three possible parameter settings.

Figure 4.4 depicts the optimal configurations of three candidate solutions to the ARMOP presented at the beginning of this section in Equation (4.3). When the problem is posed as an ARMOP, there is only a single decision variable x, and the other two variables are adjustable variables. y_1 improves the distance term in response to variations in p_1, and y_2 controls the direction (i.e., the ratio between the objectives) and can respond to variations in p_2. As discussed in Section 4.3.1, the distance term can be minimized by both increasing x or setting y_1 close to p_1. The "adaptability constraint" $x + |y_1| + |y_2| \leq 1$ introduces a trade-off between the minimum achievable distance term, and the ability to respond to variations in the uncertain parameters. Additionally, solutions that are more adaptive (i.e., having a smaller value of x) are able to span a wider front of Pareto-optimal configurations for each realisation of the uncertainties, thanks to a wider range for y_2.

Each colour in Figure 4.4 is associated with a different candidate solution and each marker with a single scenario of the uncertain parameters. The blue solution $x^{(1)} = 0.6$ is the most adaptive one, and the yellow solution $x^{(3)} = 1.0$ has no adaptability at all. The optimal configurations and their corresponding objective values are depicted in Figures 4.4(a), and 4.4(b), respectively. The following observations on the ARMOP can be made from Figure 4.4:

1. The adaptability constraint is active for all optimal configurations, regardless of the values of the uncertain parameters. This is evident from Figure 4.4(a).[1]

[1] The non-dominated solutions among the optimal trade-off curve appear with symbols. Since the

2. Adaptability allows the solution to maintain similar performance after a change of the uncertain parameters. It can be seen in Figure 4.4(b) that the difference in performance of $x^{(3)}$ between any two scenarios is much greater than the difference of $x^{(1)}$. The maximum deviation in performance for the above example is between the circle and diamond scenarios. For $x^{(3)}$, the Euclidean distance between the performance vector of each scenario is 1.13, whereas the minimum distance between performance vectors of $x^{(1)}$ for the two scenarios is 0.27.

3. Adaptive solutions span a wider Pareto front for every uncertain scenario than non-adaptive ones. This leaves more room for decision-making whenever a change occurs in the uncertain parameters.

In the next section, the above problem is used to demonstrate various approaches to evaluate and compare candidate solutions to an ARMOP.

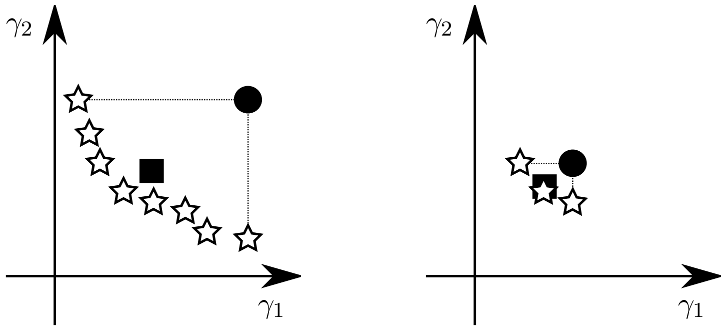
4.4 Evaluating Candidate Solutions for ARMOPs

In the previous section, the unique properties of an ARMOP were introduced by considering a small number of scenarios of the uncertain parameters, $\mathbf{P}$. It was demonstrated that for every scenario, $\mathbf{p}$, there might exist a set of optimal configurations, $\underline{\mathbf{y}}^\star(\mathbf{x}, \mathbf{p})$, and a corresponding front of optimal performance vectors, $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}, \underline{\mathbf{y}}^\star, \mathbf{p}\big)$. However, in order to properly evaluate a candidate solution, one should consider its Pareto front for all possible scenarios of $\mathbf{P}$, or at least a sufficiently large representative set of scenarios. According to the AROP methodology, the variate $\underline{\boldsymbol{\Gamma}}$ in Equation (4.1) should be replaced with a robustness criterion $I[\underline{\boldsymbol{\Gamma}}]$. The formulation of the ARMOP then becomes:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} I[\underline{\boldsymbol{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}}^\star, \mathbf{P})] \\ \text{where:} \quad &\underline{\mathbf{Y}}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \boldsymbol{\Gamma}(\mathbf{x}, \mathbf{y}, \mathbf{P})\,. \end{aligned} \tag{4.16}$$

Although there are existing robustness criteria for multi-objective random variates (see Section 2.4.2), a criterion that quantifies a variate of sets cannot be found in the existing literature. Figure 4.5 demonstrates why existing robustness criteria are not suitable for quantifying a Pareto-optimal front by reducing it to a single representative objective vector. Consider the set of Pareto-optimal performance vectors for a minimization problem, marked with stars in Figure 4.5(a). This set can be the Pareto-optimal front $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}^{(a)}, \underline{\mathbf{y}}^\star, \mathbf{p}\big)$ of a candidate solution $\mathbf{x}^{(a)}$ for a single scenario $\mathbf{p}$. Averaging these vectors will result in the performance vector marked as a black square. Considering the

same eleven configurations are used for all scenarios, the symbols in Figure 4.4(a) overlay at some points.



(a) solution $\mathbf{x}^{(a)}$ (b) solution $\mathbf{x}^{(b)}$

Figure 4.5: Unsuitable robustness criteria for a Pareto front. Pareto-optimal performances are marked with stars, the mean value is marked with a square and the worst-case with a circle.

worst-case could be interpreted as the vector marked with a circle, which is the nadir objective vector defined as follows:

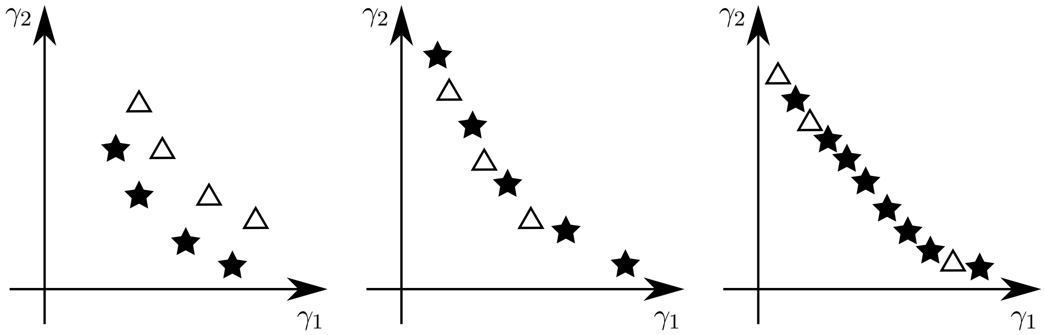
$$\begin{aligned} \boldsymbol{\gamma}^{\text{nad}} &= \left[\gamma_1^{\text{nad}}, \ldots, \gamma_j^{\text{nad}}, \ldots, \gamma_{n_\gamma}^{\text{nad}}\right] \\ \text{where:} \quad \gamma_j^{\text{nad}} &= \max_{\mathbf{y} \in \underline{\mathbf{y}}^\star} \gamma_j(\mathbf{x}, \mathbf{y}, \mathbf{p}) \ \forall\, j = \{1, \ldots, n_\gamma\}\,. \end{aligned} \tag{4.17}$$

Now consider the optimal performances of another candidate solution $\mathbf{x}^{(b)}$ for the same scenario $\mathbf{p}$, as depicted in Figure 4.5(b). It can be seen that $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}^{(b)}, \underline{\mathbf{y}}^\star, \mathbf{p}\big)$ is a subset of $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}^{(a)}, \underline{\mathbf{y}}^\star, \mathbf{p}\big)$, and therefore it provides the user with fewer alternatives for decision-making. Assuming no *a priori* preferences have been specified, the smaller extent provided by Solution $\mathbf{x}^{(b)}$ makes it less attractive than Solution $\mathbf{x}^{(a)}$ (according to this single scenario $\mathbf{p}$). Nevertheless, both robustness criteria mentioned above consider Solution $\mathbf{x}^{(b)}$ as superior to Solution $\mathbf{x}^{(a)}$. This can be inferred from the fact that both the mean performance vector and the nadir vector of Solution $\mathbf{x}^{(a)}$ are dominated by those of Solution $\mathbf{x}^{(b)}$.

4.4.1 Requirements from Robustness Indicators for ARMOPs

Before performance indicators for ARMOPs can be suggested, it is important to understand what properties make an adaptive solution preferable. The performance of an adaptive solution is a variate of Pareto-optimal frontiers, $\underline{\boldsymbol{\Gamma}}$. When these Pareto frontiers are evaluated by a performance indicator, they need to be evaluated according to the common notions of optimality in multi-objective optimization, as listed in Chapter 2:

1. **Dominance.** The portion of the objective space dominated by the Pareto front should be maximized.



(a) Dominance (b) Extent (c) Distribution

Figure 4.6: Differences in quality of Pareto frontiers for different criteria.

2. **Extent.** The range of possible values for each objective should be maximized. This allows the DM to incorporate higher-level knowledge that is not available during the design process to choose a configuration from different regions of the objective space.

3. **Distribution.** The distances between neighbouring vectors of the Pareto front should be minimized. This allows the DM a fine resolution in choosing the appropriate trade-off between objectives.

4. **Pertinence.** The relevance of the solutions in the Pareto front to the decision-maker's preferences. Whenever preferences can be described before or during the optimization process, the search can be accelerated by targeting only Pareto-optimal solutions that satisfy those preferences. Pertinence is not a quality of the Pareto-front *per se*, but it is necessary to check whether or not it can be considered when evaluating a front.

Figure 4.6 depicts the differences in quality between two candidate solutions for the first three criteria, based on their optimal performance for a single realisation of the uncertainties. The solution marked with stars is better than the one marked with triangles for the criterion mentioned in any panel.

An evaluation of a candidate solution $\mathbf{x}$ to an ARMOP consists of the following stages:

1. Evaluating the quality of the Pareto front, $\underline{\boldsymbol{\gamma}}(\mathbf{x}, \mathbf{p})$, for each scenario under consideration using a quality indicator, $q\big[\underline{\boldsymbol{\gamma}}(\mathbf{x}, \mathbf{p})\big]$.

2. Constructing the distribution (or an approximation) of the quality indicator, $Q[\underline{\boldsymbol{\Gamma}}(\mathbf{x}, \mathbf{P})]$, that corresponds to the distribution of the uncertain parameters, $\mathbf{P}$.

3. Quantifying the distribution with a robustness indicator, $I[Q]$.

Once the Pareto front, $\underline{\boldsymbol{\gamma}}(\mathbf{x},\mathbf{p})$, is replaced with the quality indicator, $q\big[\underline{\boldsymbol{\gamma}}(\mathbf{x},\mathbf{p})\big]$, the evaluation becomes similar to a common robust optimization procedure, where a random variate needs to be assessed. In the following, several quality indicators are introduced. The approach taken to solve the ARMOP is highly influenced by the choice of indicator. For example, a unary indicator that quantifies the PF with a scalar value leads to a single-objective RO, an indicator that results in a vector leads to a multi-objective RO, and an indicator based on goal vectors can be used in a decomposition-based framework.

4.4.2 Single-Objectivisation

A straightforward approach for an ARMOP is to identify a high-priority objective and to transform the ARMOP into a single-objective AROP. The other objectives are also taken into account when choosing a candidate design, but adaptation only takes place in response to changes in the high-priority objective. This kind of approach can be taken, for example, when a design need to be optimized for maximum functionality and minimum cost. Different candidate designs have a different degree of functionality and different cost. Additionally, the configuration in which the product is being used affects the operational costs. While the overall production cost and operational cost is taken into account when the solution is evaluated, during operation, the functionality of the product dictates the configuration. An example of an optical table optimization is provided in Chapter 5 to illustrate this approach.

Consider the ARMOP defined in Equations (4.1)–(4.2). Instead of considering the entire Pareto set of optimal configurations $\underline{\mathbf{y}^\star}(\mathbf{x},\mathbf{p})$ for every scenario $\mathbf{p}$, the objective vector is decomposed into a *master* objective γ_{m} and *slave* objectives $\boldsymbol{\gamma}_{\mathrm{s}}$:

$$\boldsymbol{\gamma} = \left[\gamma_{\mathrm{m}}, \gamma_{\mathrm{s},1}, \ldots, \gamma_{\mathrm{s},n_\gamma-1}\right] . \tag{4.18}$$

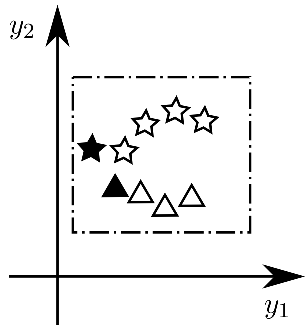
Given the uncertainty the variate $\boldsymbol{\Gamma}$ becomes:

$$\boldsymbol{\Gamma} = \left[\Gamma_{\mathrm{m}}, \Gamma_{\mathrm{s},1}, \ldots, \Gamma_{\mathrm{s},n_\gamma-1}\right] . \tag{4.19}$$

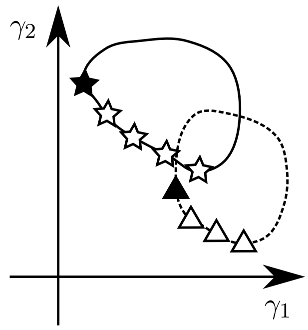
The ARMOP then becomes the following AROP:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\Gamma}(\mathbf{x},\mathbf{Y}^\star,\mathbf{P})\,, \\ \text{where: } &\mathbf{Y}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \Gamma_{\mathrm{m}}(\mathbf{x},\mathbf{y},\mathbf{P})\,. \end{aligned} \tag{4.20}$$

Note that the variate $\boldsymbol{\Gamma}(\mathbf{x},\mathbf{Y}^\star,\mathbf{P})$ is still a vector variate, but now there is a one-to-one mapping between each scenario $\mathbf{p}$ and the objective vector $\boldsymbol{\gamma}(\mathbf{x},\mathbf{y}^\star,\mathbf{p})$.



(a) Adjustable design space



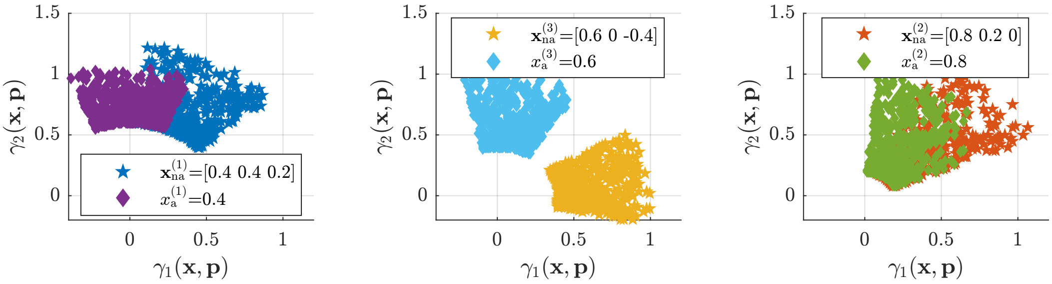
(b) Objective space

Figure 4.7: Single-objectivisation of an ARMOP for two scenarios of the uncertain parameters.

Figure 4.7 depicts a single-objectivisation of the example used in Figure 4.3, with γ_1 being the *master* objective. The original Pareto sets and Pareto frontiers are shown for the star and triangle scenarios. The configuration that minimizes γ_1 is marked with a black marker for each scenario. By using this approach, the ARMOP can be solved as a conventional multi-objective robust optimization problem, such as the one discussed in Section 4.3.2. In the following, the example ARMOP in Equation (4.3) is solved using single-objectivisation. First, the set of optimal configurations is found for each scenario, as described in Section 4.3.3, then the configuration that minimizes γ_1 is chosen.

To illustrate the difference in performance between adaptive and non-adaptive solutions, consider the non-adaptive solutions to the problem presented in Figure 4.2. Three solutions are depicted in this figure with x_1 values of 0.4, 0.6 and 0.8, and various values for x_2 and x_3. Now consider the ARMOP formulation of the problem, where x_1 is the only decision variable, x, and x_2 and x_3 are replaced with the adaptable decision variables, y_1 and y_2, respectively. Figure 4.8 depicts the distribution of objective vectors of the solutions in Figure 4.2 with their adaptive counterparts with the same value for x and x_1. While for the non-adaptive solutions every scenario $\mathbf{p}$ maps into a performance vector according to the fixed values of x_2 and x_3, for the adaptive solutions each scenario maps into the performance vector with minimum γ_1 from all possible configurations of y_1 and y_2. It is clearly visible that adaptation allows for a significant improvement of the dominant objective γ_1. It is worth mentioning however that the non-adaptive solutions were arbitrarily chosen, and not optimized to favour γ_1.

The ARMOP (4.3) consists of a single decision variable x. While it is quite challenging to display the distribution $\mathbf{\Gamma}(x)$ for every candidate solution in the feasible range $x \in (0, 1]$, a value of a robustness indicator for every distribution can be visualised. Figure 4.9 depicts two robustness criteria: the worst-case scenario, and the expected



(a) $x = 0.4$ (b) $x = 0.6$ (c) $x = 0.8$

Figure 4.8: Comparison of the non-adaptive solutions from Figure 4.2 with their adaptive counterparts.

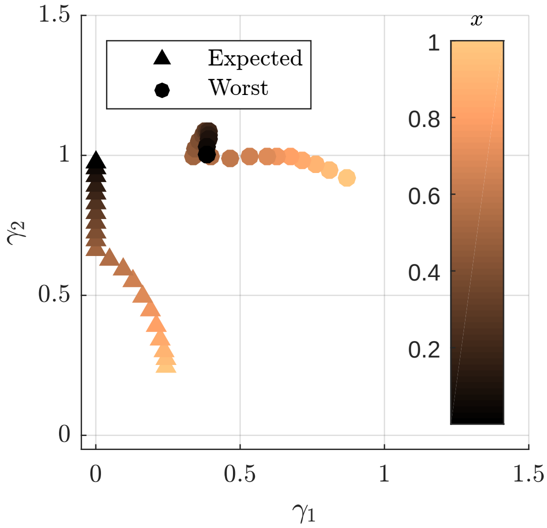


Figure 4.9: Expected and worst-case vectors for the entire search domain after single-objectivisation.

objective vector. Both indicators consider the marginal distributions of Γ_1 and Γ_2 after single-objectivisation. The worst-case objective vector is defined as follows:

$$\boldsymbol{I}_w[\boldsymbol{\Gamma}] = \left[\max(\Gamma_1), \dots, \max(\Gamma_i), \dots, \max(\Gamma_{n_\gamma})\right], \tag{4.21}$$

and the mean objective vector is defined as follows:

$$\boldsymbol{I}_E[\boldsymbol{\Gamma}] = \left[\mathrm{E}(\Gamma_1), \dots, \mathrm{E}(\Gamma_i), \dots, \mathrm{E}(\Gamma_{n_\gamma})\right]. \tag{4.22}$$

The colour intensity of the markers in Figure 4.9 correspond to the value of x. Lighter shades describe solutions with high value of x and low adaptability, and vice versa. Although γ_1 is prioritised over γ_2 when choosing the optimal configuration for each scenario, the trade-off between the objectives can be taken into account when choosing the final solution. For example, when considering the expected performance, an improvement of γ_1 can be noticed as x decreases from 1 to 0.5, and adaptability increases respectively. However, beyond $x = 0.5$, the improvement in γ_1 is marginal, while γ_2 keeps getting worse as x decreases. If γ_1 was the only consideration, the optimal solution with respect to expected performance would be $x = 1$. However, the solution $x = 0.5$ makes a better choice if both objectives are to be considered. When the worst-case is considered, the solution $x = 0.5$ is the most obvious choice, as it minimizes γ_1, and no significant improvement in the worst value of γ_2 is achieved by any alternative solution.

Single-objectivisation is recommended for solving ARMOPs where one of the objectives has a clear priority over the others when it comes to adaptation. Its main benefits are:

1. The complexity of the ARMOP is significantly reduced since a single-objective optimization problem needs to be solved for the optimal configuration, instead of an MOP.

2. The quality indicator for evaluating each set of optimal configurations is the objective vector with the best value of γ_{m}. Since it is defined in the original objective space, and no utility function is used, it is easier for a decision-maker to set a robustness criterion over the distribution of the quality indicator.

It is acknowledged however that single-objectivisation can only be used for a specific type of ARMOPs, where a leading objective for adaptation can be identified. When the ARMOP cannot be formulated in this fashion, a different approach should be taken. In the following section, a scalarising approach, inspired by decomposition-based EMOAs, is suggested.

4.4.3 Decomposition-Based Approach Using Scalarisation

Decomposition-based methods have gained increasing popularity for solving multi-objective optimization problems over the last decade, (Giagkiozis et al., 2013b). A decomposition-based algorithm decomposes an MOP into n single-objective problems, each targeting a different ratio between the objectives. This sort of decomposition can be seen in Figure 4.10: a set of six reference direction vectors (grey) is being used to guide the search towards different regions on the Pareto front. A different sub-problem is defined for every direction vector by using a scalarising function, $s(\boldsymbol{\gamma}, \mathbf{w})$, that maps an objective vector, $\boldsymbol{\gamma}$, into a scalar value according to a vector of weights, $\mathbf{w}$. The weights vector, $\mathbf{w}$, is composed of non-negative components that sum to one. The i^{th} component of $\mathbf{w}$ is the relative weight of objective, γ_i.

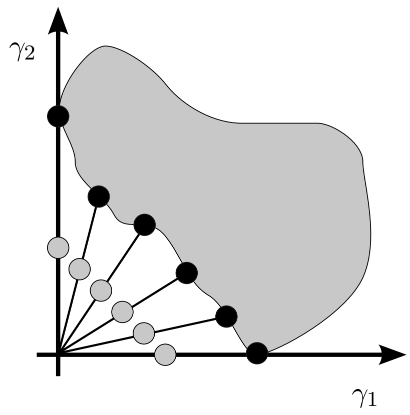


Figure 4.10: Decomposition of a bi-objective problem using a set of reference direction vectors.

A variety of scalarising functions exists in the literature, as well as methods for generating the set of weighting vectors (see Hughes, 2003; Tan et al., 2012; Giagkiozis et al., 2014). Without loss of generality, the approach is illustrated in the following using the *Weighted Chebyshev* scalarising function described in Equation (4.23) below:

$$s = \max_{i=1}^{n_\gamma} \big(w_i \cdot (\gamma_i - \gamma_i^\star) \big), \tag{4.23}$$

where $\boldsymbol{\gamma}^\star$ is a reference vector in objective space, typically the *ideal* vector, composed of the lower bound for each objective.

It is usually desired in multi-objective optimization to find a set with an even distribution across the Pareto front. The distribution of the obtained solutions is highly influenced by the choice of weighting vectors used to decompose the problem. A common practice is to use an evenly spaced set of weighting vectors (e.g., using a simplex-lattice design (Scheffe, 1958)). However, an even distribution of the weights may lead to a distorted distribution in objective space. For this reason, the *generalized decomposition* framework, suggested by Giagkiozis et al. (2014), is utilised to generate an evenly spaced set of reference direction vectors in objective space, and to find the optimal weighting vector for each direction. Wang et al. (2013) has shown that the optimal weighting vector $\mathbf{w}$ for the weighted Chebyshev scalarising function (4.23) is the normalised reciprocal of the direction vector, $\mathbf{d}$:

$$\begin{aligned} &\mathbf{w} = \left[w_1, \ldots, w_{n_\gamma} \right], \\ &w_i = a \cdot (d_i + \epsilon)^{-1} \quad , \quad i = 1, \ldots, n_\gamma, \\ &\sum_{i=1}^{n_\gamma} w_i = 1, \end{aligned} \tag{4.24}$$

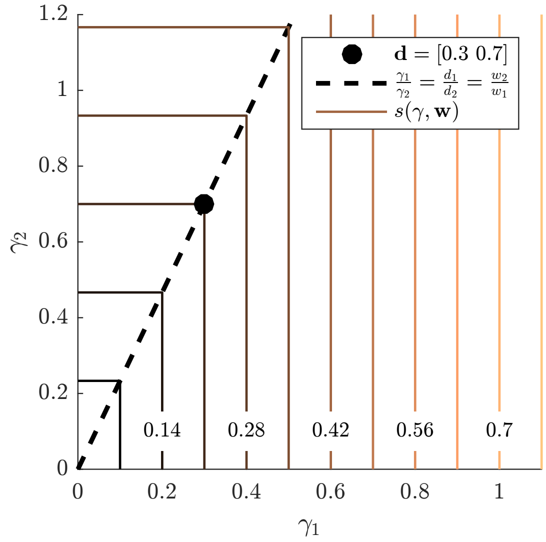


Figure 4.11: Contours of equal weighted Chebyshev values in a bi-objective space for a given direction vector.

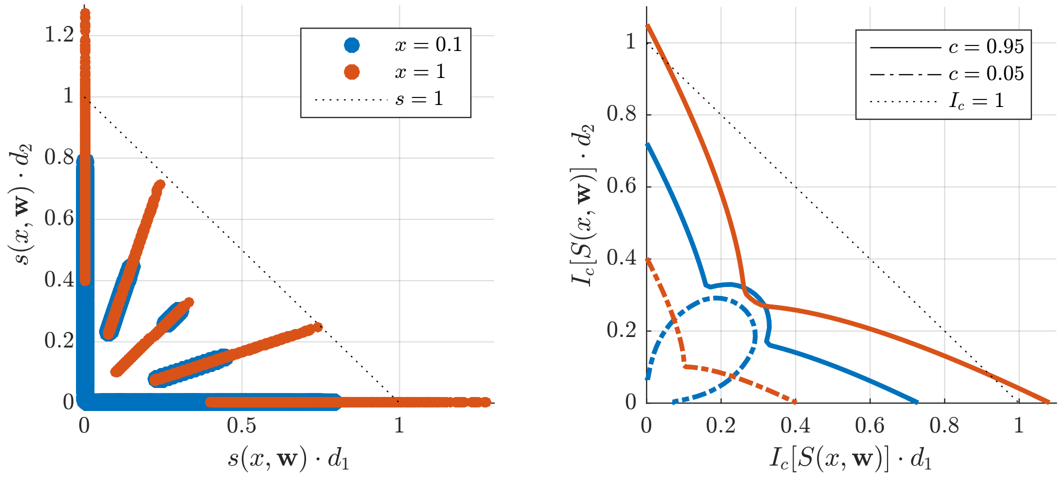
where ϵ is a small number to prevent division by zero, and a is a normalisation factor.

Figure 4.11 depicts the topography of the weighted Chebyshev function in a bi-objective space for a given direction vector $\mathbf{d} = [0.3, 0.7]$. The direction vector is marked with a black dot, and all objective vectors with the ratio $\gamma_1/\gamma_2 = 3/7$ (i.e., with the same direction as $\mathbf{d}$) are marked with a dashed line. The coloured lines represent contours of the scalarising function in Equation (4.23), with $\boldsymbol{\gamma}^\star$ located at the origin. Below the direction line, where $\gamma_1/\gamma_2 > d_1/d_2$, the value of the scalar function only depends of the value of γ_1 according to the relation $s = 0.7\gamma_1$. Similarly, above the direction line the scalar function corresponds to $s = 0.3\gamma_2$. This illustrates the inverse relation $d_1/d_2 = w_2/w_1$, described in Equation (4.24).

Using a set of n_w weighting vectors and a scalarising function $s(\boldsymbol{\gamma}, \mathbf{w})$, the ARMOP defined in Section 4.2 is decomposed into n_w single-objective AROPs, where the i^{th} AROP is defined as follows:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathcal{X}} S\big(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}, \mathbf{w}^i\big)\,, \\ &\text{where: } \mathbf{Y}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} s\big(\mathbf{x}, \mathbf{y}, \mathbf{P}, \mathbf{w}^i\big)\,. \end{aligned} \tag{4.25}$$

For every scenario of the uncertainties $\mathbf{p}$, the optimal configuration $\mathbf{y}^\star$ is the member of the Pareto-optimal set $\underline{\mathbf{y}^\star}$ with the best scalar function value. Let us assume that the Pareto front, $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}, \underline{\mathbf{y}^\star}, \mathbf{p}\big)$, is continuous and includes an objective vector with the same ratio as the reference direction vector. For the weighted Chebyshev function, the optimal configuration is the one with the same ratio between the objectives as the ratio



(a) Distribution of scalarised function values for a set of five direction vectors.

(b) I_c values for all direction vectors.

Figure 4.12: Performance of two candidate solutions according to weighted Chebyshev for different direction vectors.

of the direction vector.

To illustrate how decomposition can be used to solve and analyse candidate solutions to ARMOPs, consider the analytic example (4.3). Figure 4.12 depicts the performance of two candidate solutions according to the weighted Chebyshev scalarising function. The sampled distribution of the scalar value $\bar{S}$ is displayed in Figure 4.12(a), and the confidence robustness indicator for this distribution is depicted in Figure 4.12(b). Both figures describe a two-dimensional space where the units in both dimensions are those of the scalar function. Each sample $s(\boldsymbol{\gamma}, \mathbf{w}^i)$ is represented by the vector $s\mathbf{d}^i$, where $\mathbf{w}^i$ is the optimal weight for direction, $\mathbf{d}^i$. The dotted line corresponds to $s = 1$ in all directions.

A comparison between two extreme solutions is shown in the figure. The blue solution $x = 0.1$ is highly adaptive, which means that the direction of the objective vector can be controlled and adjusted to the required direction in each sub-problem. However, the distance term is generally large for most samples of $\mathbf{p}$. The red solution $x = 1$ has no adaptability at all, and its performance is determined solely by the realisation of the uncertainties. Thanks to the high value of x, the distance term is lower than that of the other solution for most of the scenarios. It can be concluded from Figure 4.12 that adaptation has an advantage for the above problem when one objective is preferred over the other. In this case, the adaptive solution achieves better values of the scalar function for all scenarios. However, when both objectives are equally important, the non-adaptive solution has an advantage.

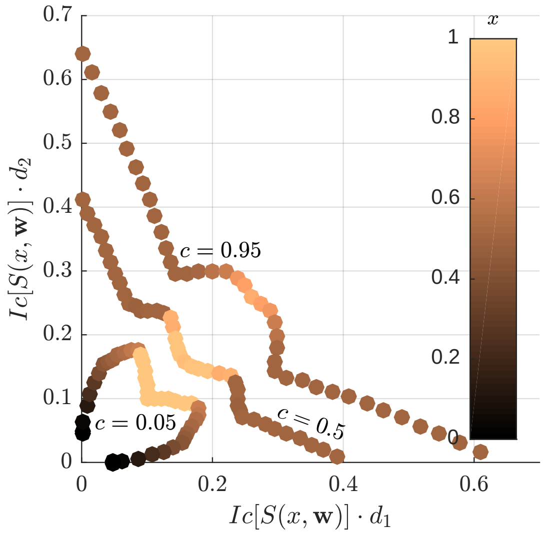


Figure 4.13: Optimal I_c value for each direction for three levels of confidence, c.

The above example only considered two candidate solutions for the problem (4.25), which are either extremely adaptive or non-adaptive. In the following, the ARMOP is decomposed to $n_w = 41$ sub-problems, and is solved for the I_c indicator with three levels of confidence: $c = 0.05$, $c = 0.5$ and $c = 0.95$. The solution is obtained by using brute force. For every sub-problem, the indicator value is calculated for an enumeration of the decision space $\mathcal{X}$ with a resolution of 0.01. The solution with the minimum value of the indicator I_c is then selected as the optimal solution for the sub-problem.

Figure 4.13 depicts the optimal indicator value at every direction in a similar fashion to Figure 4.12. For each sub-problem with direction vector, $\mathbf{d}^i$, and weighting vector, $\mathbf{w}^i$, the optimal solution is depicted by the vector, $I_c\big[S\big(\mathbf{\Gamma}, \mathbf{w}^i\big)\big]\,\mathbf{d}^i$. The colour of the marker represents the value of the optimal solution, $\mathbf{x}$.

By examining the results in Figure 4.13, we learn that the solution $x = 0.5$ is the optimal solution for a wide range of trade-offs between the objectives. As the required level of confidence increases, the solution $x = 0.5$ becomes the optimal solution for a wider range of direction vectors. This finding supports the optimization results in Section 4.4.2, where the solution $x = 0.5$ was identified as the preferred solution for the single-objectivisation variant of the ARMOP.

Both approaches for evaluating adaptive solutions presented so far consider a single optimal configuration from the set $\underline{\mathbf{y}^\star}$. In the following, the entire set of optimal configurations for each scenario of the uncertainties is evaluated as a whole, using quality indicators for set-based optimization.

4.4.4 Set-Based Unary Indicator

In the field of evolutionary multi-objective optimization it is common to benchmark newly proposed algorithms on a set of test problems, and compare the approximated Pareto frontiers achieved by each algorithm with the aid of quality indicators. The quality indicators used in EMO follow the guidelines described in Section 4.4.1, i.e., proximity, diversity and pertinence. This section describes how a unary indicator for quantifying a set of performance vectors can be used to solve the ARMOP. A unary quality indicator $q\left[\underline{\boldsymbol{\gamma}}(\mathbf{x}, \underline{\mathbf{y}^{\star}}, \mathbf{p})\right]$ is used for every scenario of the uncertainties to scalarise the Pareto frontier of every candidate solution. The indicator variate of different candidate solutions can then be compared and ranked through the use of robustness criteria. Keeping in mind that the loss of meaningful information is inevitable when using a scalarising function, it is important to use an indicator that preserves as much information as possible regarding the quality of the trade-off surface $\underline{\boldsymbol{\gamma}}(\mathbf{x}, \underline{\mathbf{y}^{\star}}, \mathbf{p})$.

One of the well-known quality indicators for approximation sets is the hypervolume, defined as the volume of objective space enclosed by the Pareto front and a reference point (Zitzler, 1999). The HV measure provides an integrated measure of proximity, diversity and pertinence, although it has been shown by Knowles and Corne (2002) that it is sensitive to the choice of a reference point. Despite this drawback, it is used in this section to demonstrate the unary indicator approach.

The HV of the Pareto front of solution $\mathbf{x}$ for scenario $\mathbf{p}$ can be denoted as $q_{hv}\left[\underline{\boldsymbol{\gamma}}(\mathbf{x}, \underline{\mathbf{y}^{\star}}, \mathbf{p})\right]$. For clarity, the shortened notation $q_{hv}[\mathbf{x}, \mathbf{P}]$ is used hereafter. It is calculated as follows according to the ideal vector $\boldsymbol{\gamma}^{*}$ and the worst objective vector $\boldsymbol{\gamma}^{w}$: First, the ideal and worst objective vectors are identified as the vectors with minimum and maximum objective values, respectively, amongst all known solutions and scenarios. Next, the objectives $\underline{\boldsymbol{\gamma}}(\mathbf{x}, \underline{\mathbf{y}^{\star}}, \mathbf{p})$ are normalised in a manner that supports DM's preferences (e.g., setting $\boldsymbol{\gamma}^{*}$ to zero and $\boldsymbol{\gamma}^{w}$ to a vector of weights between 0-1). Finally, the hypervolume $q_{hv}[\mathbf{x}, \mathbf{p}]$ is calculated, using the worst objective vector as a reference. The variate of $q_{hv}[\mathbf{x}, \mathbf{p}]$ that corresponds to the variate $\mathbf{P}$ is denoted as $Q_{hv}[\mathbf{x}, \mathbf{P}]$.

Fig 4.14 demonstrates the above procedure for a population of two solutions. Three scenarios of $\mathbf{p}$ are considered, where the Pareto frontiers of the two solutions $\mathbf{x}^{\star}$ and $\mathbf{x}^{\circ}$ are depicted in stars and circles, respectively. Dashed contours show the domains in objective space for scenario $\mathbf{p}^{3}$ that include the performances of all evaluated configurations. The worst objective vector is calculated according to the objective vectors of all configurations, including non-optimal ones. The ideal vector is marked with a black triangle and the worst objective vector with a white triangle. No preference

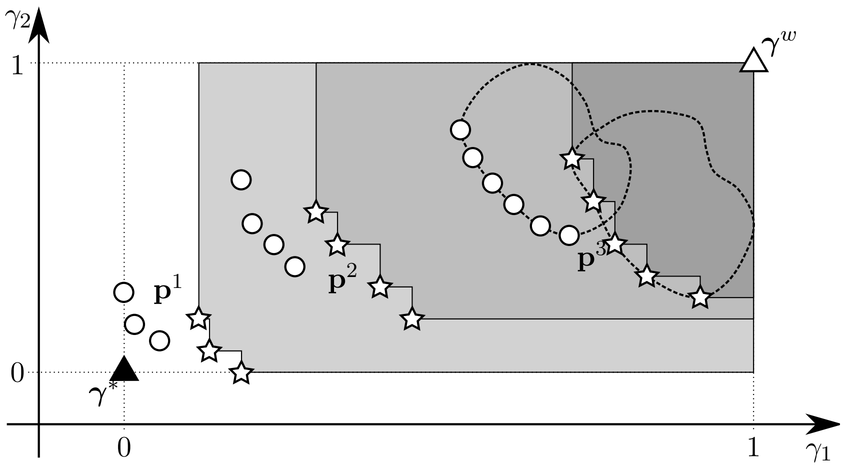


Figure 4.14: Pareto frontiers of two solutions for three scenarios of the uncertainties, and hypervolumes of one solution for the three scenarios.

is considered between the two objectives, and therefore the worst objective vector is set to $\boldsymbol{\gamma}^w = [1, 1]$. The variate $Q_{hv}\left[\mathbf{x}^{☆}, \mathbf{P}\right]$ is shown as the collection of three HVs for $\mathbf{x}^{☆}$.

Once the Pareto frontiers are scalarised, the ARMOP becomes a single-objective robust optimization problem. The distribution of the HV variates for the analytic example in Equation (4.3) are depicted in Figure 4.15. The variate of Pareto frontiers $\underline{\boldsymbol{\Gamma}}(x, \underline{\mathbf{Y}}^{\star}, \mathbf{P})$ was calculated for an enumeration of the decision space, with intervals of $x = 0.01$. The hypervolume of each Pareto front was calculated by normalising the objectives between $\gamma^* = [-0.4, -0.4]$ and $\boldsymbol{\gamma}^w = [1.25, 1.25]$. The best and worst hv value from the set of 500 samples are marked with blue and red lines, respectively. The distributions $Q_{hv}[x, \mathbf{P}]$ of nine solutions are depicted as box plots with the median marked as a black dot within a circle, the range between the 25$^{\text{th}}$ and the 75$^{\text{th}}$ percentiles with a thick line, and the rest of the samples that are not considered as outliers with thin whiskers. Outliers are marked with circles. The data provided in the box plots is in fact equivalent to $I_c[Q_{hv}[x, \mathbf{P}]]$.

The values of the hypervolume indicator for this example confirm the results from the previously presented approaches. The superiority of the solution $x = 0.5$ can be easily identified from Figure 4.15. $x = 0.5$ is a local optimum when optimizing the confidence indicator $I_c[Q_{hv}[0.5, \mathbf{P}]]$ for any confidence level c. Additionally, it is the global optimum for confidence levels of $0.35 \leq c \leq 0.95$. (For confidence levels of $c < 0.35$, the global optimum is $x = 1$, and for $c > 0.95$, the global optimum is $x = 0$).

The hypervolume indicator provides a straightforward approach to solve an ARMOP. It considers all objectives and supports preference incorporation through the normalisation of the objective vectors. However, as is common with utility objectives,

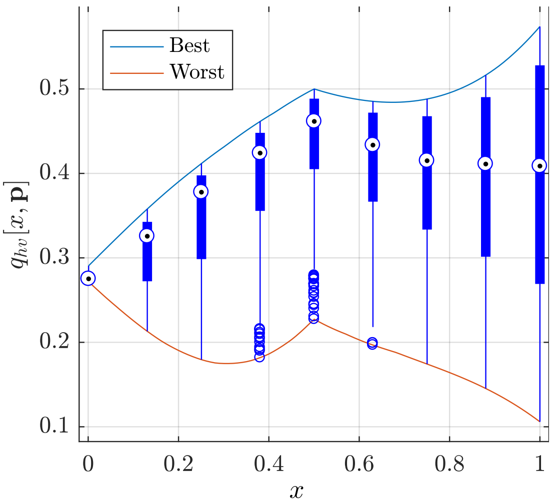


Figure 4.15: Distribution of hypervolume quality indicator for various candidate solutions over 500 samples of the uncertain parameters.

the interpretation of the indicator value is difficult, and it may be challenging to choose a robustness indicator for quantifying the variate of HV values. The computation of the hypervolume of a set is extensive, and the complexity grows exponentially in the worst-case with the number of objectives (Beume et al., 2009). Therefore, it is not advised to use this indicator for problems with many objectives. The next section demonstrates how a binary quality indicator that does not suffer from the "curse of dimensionality" can be used to compare between candidate adaptive solutions.

4.4.5 Set-Based Binary Indicator

Binary quality indicators provide a binary comparison between two candidate solutions. In contrast to unary indicators that assign a scalar value to each candidate solution, and provide a complete ordering of the population, a cyclic relation can occur with binary indicators such as $\underline{\mathbf{a}} \lhd \underline{\mathbf{b}}$, $\underline{\mathbf{b}} \lhd \underline{\mathbf{c}}$ and $\underline{\mathbf{c}} \lhd \underline{\mathbf{a}}$, where $\underline{\mathbf{a}} \lhd \underline{\mathbf{b}}$ denotes that the set of vectors $\underline{\mathbf{a}}$ is better than the set $\underline{\mathbf{b}}$. This property limits the scope of this approach to optimization schemes based on binary comparisons of candidate solutions, such as the (1+1) evolutionary algorithm (Droste et al., 2002) and differential evolution (Storn and Price, 1997).

Several binary indicators can be used to conduct the comparison, each with its strengths and weaknesses (e.g., Generational Distance (Van Veldhuizen, 1999), C metric (Zitzler and Thiele, 1998)). It is demonstrated here with the q_{ϵ_+} measure presented in Section 2.3.4. $Q_{\epsilon_+}[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ is the variate of comparisons between two candidate solutions considering the entire uncertainties domain $\mathbf{P}$.

A robustness criterion $I_{\epsilon+}\left[\mathbf{x}^{(a)}, \mathbf{x}^{(b)}\right]$ to select between two candidate adaptive

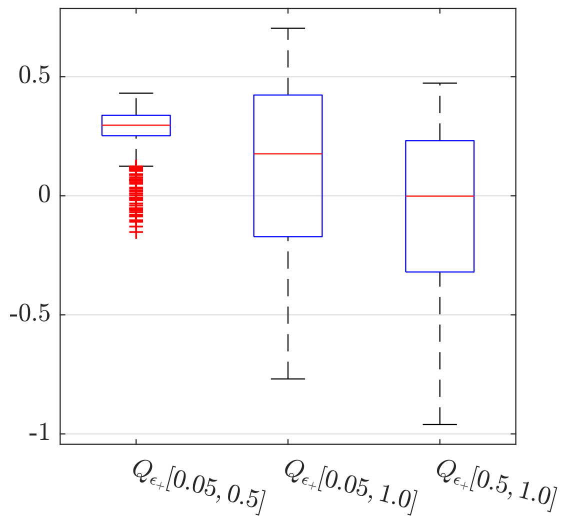


Figure 4.16: Distributions of binary comparisons between three candidate solutions over 500 samples of the uncertain parameters.

solutions $\mathbf{x}^{(a)}$ and $\mathbf{x}^{(b)}$ can be calculated by the following steps:

1. For every scenario of the uncertain parameters $\mathbf{p}$:

   (a) Calculate $\underline{\gamma}\big(\mathbf{x}^{(a)}, \underline{\mathbf{y}}^{\star}, \mathbf{p}\big)$ and $\underline{\gamma}\big(\mathbf{x}^{(b)}, \underline{\mathbf{y}}^{\star}, \mathbf{p}\big)$.

   (b) Normalise all $\boldsymbol{\gamma}$ values according to global best and worst objective vectors, $\boldsymbol{\gamma}^{*}$ and $\boldsymbol{\gamma}^{w}$.[1,2]

   (c) Calculate $q_{\epsilon+}\big[\underline{\gamma}\big(\mathbf{x}^{a}, \underline{\mathbf{y}}^{\star}, \mathbf{p}\big), \underline{\gamma}\big(\mathbf{x}^{b}, \underline{\mathbf{y}}^{\star}, \mathbf{p}\big)\big]$.

2. Construct the variate $Q_{\epsilon+}\big[\underline{\boldsymbol{\Gamma}}(\mathbf{x}^{a}, \underline{\mathbf{Y}}^{\star}, \mathbf{p}), \underline{\boldsymbol{\Gamma}}\big(\mathbf{x}^{b}, \underline{\mathbf{Y}}^{\star}, \mathbf{p}\big)\big]$ from the indicator values for all scenarios.

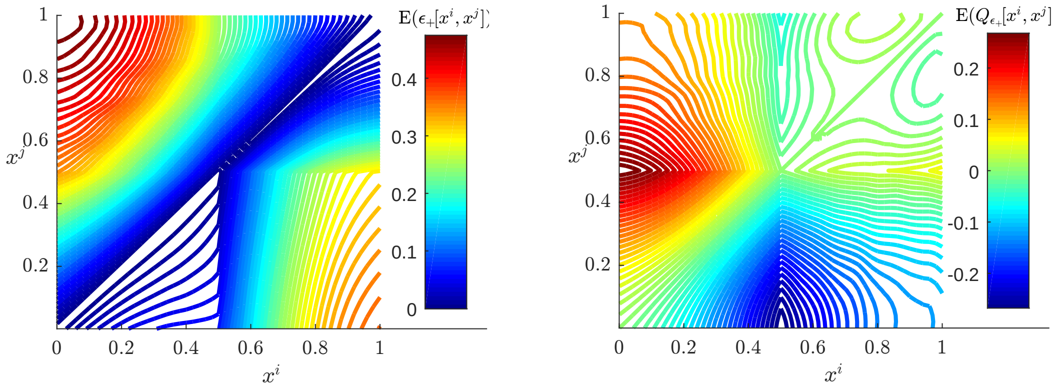
3. Choose a robustness indicator $I[\cdot]$ and compute:

$$I_{\epsilon+}\Big[\mathbf{x}^{a}, \mathbf{x}^{b}\Big] = I\Big[Q_{\epsilon+}\Big[\underline{\boldsymbol{\Gamma}}(\mathbf{x}^{a}, \underline{\mathbf{Y}}^{\star}, \mathbf{p}), \underline{\boldsymbol{\Gamma}}\Big(\mathbf{x}^{b}, \underline{\mathbf{Y}}^{\star}, \mathbf{p}\Big)\Big]\Big] \tag{4.26}$$

To illustrate how the $\epsilon+$ based binary indicator can be applied, consider the example problem in Equation (4.3). The uncertain parameters were sampled 500 times, as in previous sections, and a binary comparison between any pair of $x^{i} \in \mathcal{X}$ and $x^{j} \in \mathcal{X}$ was conducted. Figure 4.16 depicts the distribution of quality indicator $Q_{\epsilon+}$ for three candidate solutions: $x = 0.05$, $x = 0.5$ and $x = 1.0$. Three distributions of

[1]The extreme objective vectors are either based on previous knowledge of the problem at hand, or on current understanding of the objective space.

[2]The normalisation can express decision-maker's preferences by setting the worst objective vector elements to values different than one.



(a) Asymmetric ϵ_+ comparisons

(b) Symmetric quality indicator based on the comparisons in (a).

Figure 4.17: Expected values of binary comparisons between all pairs of candidate solutions.

binary comparisons can be seen in the figure in the form of box plots: $Q_{\epsilon_+}[0.05, 0.5]$, $Q_{\epsilon_+}[0.05, 1.0]$ and $Q_{\epsilon_+}[0.5, 1.0]$. Keeping in mind that a negative value of $q_{\epsilon_+}\left[x^a, x^b\right]$ implies that x^a is preferable to x^b for the simulated scenario, and vice versa, it can be concluded that the solution $x = 0.05$ is worse than the other two solutions in any possible criterion.

An interesting property of the ARMOP is revealed from the boxplot of $Q_{\epsilon_+}[0.5, 1.0]$; for half of the scenarios $q_{\epsilon_+}[0.5, 1.0] > 0$ and for the other half $q_{\epsilon_+}[0.5, 1.0] < 0$. This implies that adaptability can improve the overall quality of the trade-off performance for only half of the scenarios, while for the other half, the single configuration of the non-adaptive solution $x = 1.0$ outperforms the entire set of optimal configurations of the adaptive solution $x = 0.5$. An example for such a case can be seen in Figure 4.4 for the scenario $\mathbf{p} = [0.4, 0.4]$ (marked with diamonds) where the non-adaptive solution (yellow) almost entirely dominates the adaptive solution $x = 0.6$ (blue).

Despite the fact that the median of $Q_{\epsilon_+}[0.5, 1.0]$ is very close to zero, it can be concluded from the boxplot that the solution $x = 0.5$ is better than $x = 1.0$. For the scenarios where $x = 0.5$ is preferable to $x = 1.0$, the degree of ϵ_+ dominance is higher than for the opposite case. In other words, while $x = 1.0$ offers some improvement over $x = 0.5$ for half of the scenarios of $\mathbf{P}$, for the other half, $x = 0.5$ offers a more significant improvement.

The expected values of $\epsilon_+\left[x^i, x^j\right]$ and $Q_{\epsilon_+}\left[x^i, x^j\right]$ for all x^i and x^j within the domain are depicted in Figure 4.17. Cold shades, representing low values, suggest that x^i is preferable over x^j, and warm shades that x^j is preferable over x^i. Figure 4.17(a) presents the expected value of $\epsilon_+\left[x^i, x^j\right]$. It is evident from the figure that all of the values are non-negative, which implies that on average, all solutions have at least one non-dominated configuration when compared to another candidate solution. Note the asymmetry between the values of $\epsilon_+[x^i, x^j]$ and $\epsilon_+[x^j, x^i]$. As demonstrated in Figure A.1 in Appendix A, when two sets do not dominate each other, the ϵ_+ comparison yields positive values for both comparisons. The solution that has the lower ϵ_+ value when provided as the first argument offers a better set of trade-offs between the objectives.

The expected values of $Q_{\epsilon_+}[x^i, x^j]$ are depicted in Figure 4.17(b). The symmetry of the indicator can be observed in the figure. All indicator values on the main diagonal where $x^i = x^j$ equal zero, and the diagonal serves as the axis of negative symmetry. The advantage of the solution $x = 0.5$ over any other candidate solution stands out from the figure. All indicator values for $x^i = 0.5$ are negative for $x^j \neq 0.5$, and therefore $x = 0.5$ serves both as a local and global optimum when using this indicator. The problem has another local optimum at the non-adaptive solution $x = 0.97$. It shows that adaptability provides a substantial advantage only when the direction related variable, y_2, can be exploited within its maximum adaptability range of $-0.5 \leq y_2 \leq 0.5$.

Various approaches to evaluate and compare adaptive solutions to ARMOPs were presented in this section. The next section demonstrates how these methods can be incorporated in search heuristics in order to solve active robust multi-objective optimization problems.

4.5 Solution Approach to ARMOPs

This section demonstrates how the quality indicators for solving ARMOPs, suggested in the previous section, can be used in several variants of evolutionary algorithms.[1] The algorithms are described in a very generic manner without getting into the algorithm-specific operators. Only the high-level properties of EAs are described, namely a population of candidate solutions that evolve using selection and variation operators.

First, the basic structure of an EA that uses a quality indicator for solving ARMOPs is described and discussed. Then, four algorithms are presented, one for each indicator presented in the previous section. These algorithms share the same basic structure, but some modifications are made in order to account for the differences between the robustness indicators.

[1] EAs are a popular tool for solving difficult MOPs, but other classes of multi-objective optimizers that can exploit these indicators can also be used.

4.5.1 A Generic Algorithm for Solving ARMOPs

Algorithm 4.1 describes the template for using EAs to solve ARMOPs with the aid of a quality indicator for evaluating Pareto sets. It consists of a main EA for robust optimization and a nested optimizer.[1] The robust optimization algorithm is based on a sampled representation of the uncertain parameters $\mathbf{P}$. At the beginning of the algorithm, a set of n_p samples is derived, and the same samples are used to evaluate every candidate solution throughout the course of the optimization.

Algorithm 4.1 A generic EA for solving an ARMOP

1: sample the uncertain domain $\mathbf{P}$
2: generate an initial population
3: **repeat**
4:     **for** every new solution $\mathbf{x}$ **do**
5:         **for** every scenario $\mathbf{p} \in \mathbf{P}$ **do**
6:             optimise configurations $\underline{\mathbf{y}}^\star(\mathbf{x}, \mathbf{p})$
7:             calculate quality indicator $q\big[\underline{\boldsymbol{\gamma}}\big(\mathbf{x}, \underline{\mathbf{y}}^\star, \mathbf{p}\big)\big]$
8:         **end for**
9:         construct quality indicator variate $Q[\underline{\mathbf{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}}^\star, \mathbf{P})]$
10:         calculate robustness indicator $I[Q]$
11:     **end for**
12:     evolve new population (selection and variation)
13: **until** stopping criteria satisfied

The high complexity associated with solving ARMOPs can be observed from the depth of nested loops in Algorithm 4.1, where Line 6 in the most inner loop encapsulates an entire multi-objective optimization run. Lines 4–11 describe the procedure for calculating the fitness function of a single candidate solution $\mathbf{x}$ for the main problem. The fitness function is the robustness indicator value for the uncertain quality indicator $I[Q]$. This procedure requires the inner MOP in Line 6 to be solved n_p times, i.e., once for every sampled scenario. The total number of function evaluations required to solve an ARMOP is $n_\mathrm{o} \cdot n_\mathrm{i} \cdot n_p$, where n_o and n_i are the number of function evaluations required for the outer and inner optimization algorithms, respectively.[2] In addition to the function evaluations, the operators of the inner optimizer, as well as the calculation of $q\big[\underline{\boldsymbol{\gamma}}\big]$, need to be executed $n_\mathrm{o} \cdot n_p$ times, and the operators of the outer optimizer need to be executed as well.

[1] The nested optimizer can be any algorithm for multi-objective optimization.

[2] The number of function evaluations of the inner and outer algorithms can be either fixed or subject to variations according to the termination criteria.

4.5.2 Indicator-Specific Algorithms

In the following, the details of the robustness indicator are incorporated into the basic EA in Algorithm 4.1. Four different algorithms are described, one for each of the presented approaches for evaluating candidate solutions to ARMOPs.

Algorithm 4.2 demonstrates how the single-objectivisation approach can be implemented with an EA. Note that in this algorithm, the inner problem is a single-objective optimization problem, and the outer problem is a multi-objective robust optimization problem. There is no indicator for assessing the PF for each scenario, and instead, the objective vector of the optimal configuration for the leading objective is used. If the robustness indicator for assessing the multivariate $I[\boldsymbol{\Gamma}]$ is a scalar function, the outer algorithm is a single-objective optimizer. If the indicator is a vector function (e.g., a different indicator for each marginal distribution), the outer algorithm is a multi-objective optimizer.

Algorithm 4.2 Single-objectivisation based EA

define a leading objective γ_{m}
sample the uncertain domain $\mathbf{P}$
generate an initial population
repeat
    for every new solution $\mathbf{x}$ **do**
        for every scenario $\mathbf{p} \in \mathbf{P}$ **do**
            optimise configuration $\mathbf{y}^{\star}(\mathbf{x}, \mathbf{p}) = \underset{\mathbf{y}}{\operatorname{argmin}}\, \gamma_{\mathrm{m}}(\mathbf{x}, \mathbf{y}, \mathbf{p})$
        end for
        construct vector variate $\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{Y}^{\star}, \mathbf{P})$
        calculate MO robustness indicator $I[\boldsymbol{\Gamma}]$
    end for
    evolve new population (selection and variation)
until stopping criteria satisfied

Algorithm 4.3 describes the most simple decomposition-based EA for solving ARMOPs. The ARMOP is decomposed into n_w single-objective AROPs and each AROP is solved without any interaction with the other sub-problems. This simple example is used for clarity to demonstrate the decomposition approach without shifting the focus to sophisticated algorithmic features that are not unique to the framework. Obviously, state-of-the-art methods in decomposition-based algorithms should be used when this approach is taken for solving an ARMOP.

Algorithm 4.4 demonstrates how the unary I_{hv} indicator can be used within an evolutionary algorithm. The indicator is computed for normalised values of the objective vectors, as explained in Section 4.4.4. If the boundaries of the objective space are not known and cannot be approximated, it is suggested to update them according to

Algorithm 4.3 Decomposition based EA

generate a set of weight vectors $\underline{\mathbf{w}}$
define a scalarising function $s(\boldsymbol{\gamma}, \mathbf{w})$
sample the uncertain domain $\mathbf{P}$
for every weight vector $\mathbf{w}^i \in \underline{\mathbf{w}}$ **do**
    generate an initial population
    repeat
        for every new solution $\mathbf{x}$ **do**
            for every scenario $\mathbf{p} \in \mathbf{P}$ **do**
                optimise configuration $\mathbf{y}^\star(\mathbf{x}, \mathbf{p}) = \operatorname*{argmin}_{\mathbf{y}} s\big(\boldsymbol{\gamma}(\mathbf{x}, \mathbf{y}, \mathbf{p}), \mathbf{w}^i\big)$
            end for
            construct variate $S\big(\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}), \mathbf{w}^i\big)$
            calculate robustness indicator $I[S]$
        end for
        evolve new population (selection and variation)
    until stopping criteria satisfied
    add solution of sub-problem i to the solution set
end for

the knowledge acquired during the optimization process. After every evaluation of the objective functions, the limits are updated so that the ideal vector consists of the best values for every objective, and the anti-ideal the worst values. Since in EAs the recently generated solutions compete against existing solutions, the indicator values of the existing solutions need to be updated whenever the limits have changed. This allows for a fair comparison between all candidate solutions. To do so, the variate of Pareto frontiers $\underline{\boldsymbol{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}^\star}, \mathbf{P})$ is stored in the memory for every solution in the current population. Whenever the known ideal or anti-ideal vectors change, the indicator values can be recalculated without the need to re-evaluate the objective functions. Since the I_{hv} indicator is a scalar function, the main optimizer is a single-objective robust optimization algorithm.

Algorithm 4.5 describes an EA that uses the binary $I_{\epsilon+}$ indicator to solve an ARMOP. While Algorithms 4.2–4.4 can be classified as genetic algorithms, the EA in Algorithm 4.5 follows the structure of differential evolution (DE) presented by Storn and Price (1997). At every generation of DE, new solutions are generated, and each of them is compared with an existing solution. If the new solution outperforms the old solution, then the old solution is replaced with the new one. Otherwise, the old solution remains for the next generation.

Similarly to Algorithm 4.4, the indicator is calculated for the normalised objectives according to the known boundaries of the objective space. The ideal and anti-ideal vectors are continuously updated when searching for the optimal configurations in the inner problem. At each comparison between two candidate solutions, the objectives are

Algorithm 4.4 Unary indicator based EA with dynamic reference vectors

sample the uncertain domain $\mathbf{P}$
initialise ideal and anti-ideal points (limits)
generate an initial population
repeat
  for every new solution $\mathbf{x}$ **do**
    for every scenario $\mathbf{p} \in \mathbf{P}$ **do**
      optimise configurations $\underline{\mathbf{y}}^{\star}(\mathbf{x}, \mathbf{p})$ and store PF $\underline{\gamma}(\mathbf{x}, \underline{\mathbf{y}}^{\star}, \mathbf{p})$
    end for
    construct variate $\underline{\mathbf{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}}^{\star}, \mathbf{P})$
  end for
  if limits have changed **then**
    update limits
    calculate robustness indicator $I_{\text{hv}}[\underline{\mathbf{\Gamma}}]$ of entire population
  else
    calculate robustness indicator $I_{\text{hv}}[\underline{\mathbf{\Gamma}}]$ of new solutions
  end if
  evolve new population (selection and variation)
until stopping criteria satisfied

normalised based on the current known limits.

4.6 Review of Solution Methods for ARMOPs

Four different approaches to evaluate and compare candidate solutions were suggested in this chapter:

1. **Single-objectivisation.** This approach transforms the ARMOP into a single-objective AROP by choosing a dominant objective to guide the search for an optimal configuration. The fitness of the other objectives is not taken into account during adaptation, but the entire objective vector is considered when evaluating the candidate design. Having a single optimal configuration for every realisation of the uncertainties, the adaptive solution has a one-to-one mapping between the variate of uncertain parameters $\mathbf{P}$ and the variate objective vector $\mathbf{\Gamma}$. It was demonstrated how adaptive solutions can outperform their non-adaptive counterparts for the example ARMOP when using single-objectivisation.

   Since a single-objective problem, rather than a multi-objective one, needs to be solved for every realisation of the uncertainties, the complexity of the ARMOP is significantly reduced. Therefore, this approach is highly recommended for problems where a leading objective can be identified.

   Another advantage of the single-objectivisation approach is that it does not use

Algorithm 4.5 Binary indicator based EA with dynamic reference vectors

sample the uncertain domain $\mathbf{P}$
initialise ideal and anti-ideal points (limits)
generate an initial population
for every new solution $\mathbf{x}$ **do**
  for every scenario $\mathbf{p} \in \mathbf{P}$ **do**
    optimise configurations $\underline{\mathbf{y}^\star}(\mathbf{x}, \mathbf{p})$ and store PF $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}, \underline{\mathbf{y}^\star}, \mathbf{p}\big)$
  end for
  construct variate $\underline{\boldsymbol{\Gamma}}(\mathbf{x}, \underline{\mathbf{Y}^\star}, \mathbf{P})$
end for
repeat
  for every solution $\mathbf{x}$ **do**
    Create a trial solution $\mathbf{x}'$ using genetic operators
    for every scenario $\mathbf{p} \in \mathbf{P}$ **do**
      optimise configurations $\underline{\mathbf{y}^\star}(\mathbf{x}', \mathbf{p})$ and store PF $\underline{\boldsymbol{\gamma}}\big(\mathbf{x}', \underline{\mathbf{y}^\star}, \mathbf{p}\big)$
    end for
    construct variate $\underline{\boldsymbol{\Gamma}}(\mathbf{x}', \underline{\mathbf{Y}^\star}, \mathbf{P})$
    update limits
    if $I_{\epsilon+}[\mathbf{x}, \mathbf{x}'] > 0$ **then**
      replace $\mathbf{x}$ with $\mathbf{x}'$
    end if
  end for
until stopping criteria satisfied

any utility function to evaluate the Pareto front. This makes it easy for the decision-maker to comprehend the results and to choose a desired solution.

2. **Decomposition.** With this approach the multi-objective problem is decomposed into multiple single-objective problems using a scalarising function and different weighting vectors. For every sub-problem on a given scenario of the uncertainties, the optimal configuration is the member of the Pareto-optimal set that has the best scalar function value. Since every scenario is associated with a single optimal configuration, there is a one-to-one mapping between the variate of uncertain parameters and the scalar function variate. The robustness criterion is applied to the scalar function variate in order to evaluate and compare candidate solutions in every sub-problem.

   When decomposition is used to solve the ARMOP, the best solution for every trade-off between the objectives can be identified. However, there is no guarantee that this solution will perform in a satisfactory manner when different trade-offs are desired.

   For the example ARMOP, it was found that a high measure of adaptability is desirable when one objective is preferred over the other, while a non-adaptive solution has a better distribution of the scalar function when both objectives are equally preferred. When decomposition was used to solve the example ARMOP, a single solution has been identified as the optimal solution for the majority of sub-problems. This solution was also identified as the preferred solution after solving the ARMOP through single-objectivisation.

3. **Scalarisation.** With this approach, each Pareto front is quantified by a unary indicator for sets evaluation. It is important to use an indicator that can preserve as much information as possible regarding the quality of the original Pareto front in terms of proximity, diversity and pertinence. A variate of indicator values is constructed for each candidate solution, according to the variate of uncertain parameters. Robustness indicators are then applied to the scalar variate and a single-objective robust optimization procedure can be conducted.

   The main advantage of unary indicators, when it comes to solving ARMOPs, is the ability they provide to rank a population of candidate solutions and compare them against each other. However, it is not possible to learn from the indicator values of two candidate solutions about the differences in individual qualities such as proximity or diversity of their Pareto frontiers.

   The scalarisation approach was demonstrated with the hypervolume indicator on the example ARMOP. Based on the statistical properties of the indicator values for different values of x, the same solution that was identified by other approaches could be identified as the preferred solution. The preferred solution was not identified based on the indicator value *per se*, but based on a comparison with the rest of the solutions. This is due to the loss of physical meaning of the performance vectors after passing them through a scalarising function (i.e., the hypervolume indicator). The hypervolume indicator can be used to solve ARMOPs with a small number of objectives, such as the one presented in this chapter. It is not advised to use it for many-objective problems, as it suffers from the *"curse of dimensionality"* due to exponential growth of complexity for increasing number of objectives. The time required to compute the hypervolume can be substantially reduced if the true hypervolume value is replaced with an approximation. This approach is applied to indicator-based evolutionary algorithms such as HypE (Bader and Zitzler, 2011).

4. **Binary comparison.** This approach uses a binary indicator for sets evaluation to compare between Pareto frontiers of two candidate solutions. A variety of indicators for sets comparison can be used. The Pareto frontiers of two candidate

adaptive solutions are compared for every scenario of the uncertainty domain, and quantified by the binary indicator. Based on the distribution of indicator values and a robustness criterion, one solution is preferred over the other.

The main argument for using a binary indicator for solving an ARMOP rather than a unary indicator is the availability of binary indicators that do not suffer from the "curse of dimensionality". The ϵ_+ based indicator presented in this chapter is one of them. However, this approach cannot be exploited by optimization methods that require a complete ordering between the candidate solutions. Search heuristics such as differential evolution or 1+1 evolutionary algorithm, that only conduct a comparison between an existing solution and a newly generated candidate solution, are suitable for this approach.

Each of the above approaches were used to solve the example ARMOP using an enumeration of the decision space and the uncertainty domain. The single decision variable was evaluated to a resolution of 0.01, and the optimal configurations were found for a sample of 500 scenarios of the uncertain parameters. This could be easily done for such a small scale problem, but real-world optimization problems do not tend to be that simple. Search heuristics such as evolutionary algorithms are very common for solving difficult optimization problems. To demonstrate how each approach can be applied to solve more complicated ARMOPs, high-level descriptions of evolutionary algorithms that incorporate the different metrics are given at the end of this chapter. The algorithms share a similar structure, but each of them has its own variation to allow the use of one of the suggested approaches.

The pseudo-algorithms presented in this chapter highlight the high complexity involved in solving AROPs, in general, and ARMOPs, specifically. In order to evaluate each candidate solution, a multi-objective optimization problem needs to be solved for every sample of the uncertainties. The total number of function evaluations required to solve an ARMOP is $n_\mathrm{o} \cdot n_\mathrm{i} \cdot n_p$, where n_o and n_i are the number of function evaluations required for the outer and inner optimization algorithms, respectively, and n_p is the number of samples. In addition, the quality indicator to assess the Pareto frontier of each configuration needs to be computed $n_\mathrm{o} \cdot n_p$ times.

The different approaches for solving an ARMOP that were presented in this chapter have different strengths and weaknesses. When a solution approach needs to be chosen, the following points should be taken into account:

1. **Complexity.** As discussed previously, the complexity of the algorithms for solving ARMOPs is high. However, it differs from one solution approach to another. In *single-objectivisation*, the inner problem is a single-objective problem that is easier

to solve than an MOP, and generally requires fewer function evaluations. When *decomposition* is used, many SOPs are simultaneously solved, which increases the complexity. The *set-based* approaches require the solution of an MOP for every considered scenario and the calculations of indicators to assess the Pareto frontiers. Therefore, they require the greatest computing resources. The complexity of the algorithm can serve as an additional criterion for choosing between binary and unary indicators.

2. **Flexibility for decision-making.** The ability to change the product's configuration during its service can be used for real-time decision-making. The *single-objectivisation* approach is recommended when one objective can be identified as a single criterion to drive adaptation. With this approach, additional objectives can be considered for choosing the solution, but no decision-making should be made during the product's service. The *decomposition* approach offers more opportunity for decision-making as it searches for a set of solutions, each specialised for a different trade-off between the objectives. However, this approach is not recommended if the weights between the objectives are expected to change during the product's operation, since the optimality of the identified solution is not guaranteed for the weights for which it was not optimized. The *set-based* approach offers the most flexibility for real-time decision-making. The solutions are measured according to their ability to provide a good set of configurations at various scenarios, thus facilitating a choice between different objective trade-offs.

3. **Resemblance to the original objectives.** The use of indicators to solve the ARMOP results in a modification to the original optimization problem. The amount of information contained in the original objectives that can be inferred from the utility measures should be kept as high as possible. This can allow for a rational selection between alternative solutions. The ability to identify the original objectives in the quality indicator is different from one solution approach to another. The *single-objectivisation* approach is the only presented approach that does not use a utility function to quantify the candidate solutions' performance and preserves the physical meaning of the objective vector. The *decomposition* approach aggregates the objectives through a scalarising function, and therefore some of their original information is lost. However, it is possible to visualise the indicator values by incorporating them into representations of the original objective space, as demonstrated in Figures 4.12 and 4.13. The quality indicators used in the *set-based* approaches provide very little information on each individual objective. This reduces the role of the decision-maker during the

optimization phase. Therefore, their appeal is lessened when the involvement of the decision-maker in the optimization process is desired.

4.7 Summary

Real-world optimization problems often include more than one objective. In this chapter, a generalization of the active robust optimization problem was introduced to accommodate optimization of adaptive products for multiple performance criteria. The *active robust multi-objective optimization problem* was defined as an optimization problem with at least two objectives of Type γ, i.e., a change in the product's configuration affects more than one objective.

When solving an ARMOP, a multi-objective problem needs to be solved in order to find the optimal configuration for every realisation of the uncertainties under consideration. If some objectives are in conflict, the solution is a set of optimal configurations offering different trade-offs between the objectives. As a result, the performance of a candidate solution to an ARMOP is a variate of Pareto frontiers, corresponding to the variate of uncertain parameters. Having such a performance representation makes it very difficult to rank and compare candidate solutions to an ARMOP.

To demonstrate the structure of ARMOPs, the challenges they present and the possible approaches to solve them, a simple bi-objective analytic example was presented. The problem uses the single-objective function from the AROP in Chapter 3 as a distance property for minimization. The two objectives share the same distance term while the trade-off between them is controlled by an additional direction term. The problem consists of a single decision variable, two adjustable variables and two uncertain parameters. One of the uncertain parameters affects the distance term and the other the direction term. Similarly, each adjustable variable can react to changes in either the distance or direction term. The amount of adaptability is determined by the value of the decision variable. The problem has the smallest amount of variables and parameters to contain all the interesting features of the ARMOP, while its simplicity makes it possible to analyse and understand the properties of alternative approaches for solving ARMOPs.

One of the main goals for this study was to suggest evaluation metrics for adaptive products. In the previous chapter, several indicators for robust optimization were applied to AROPs, based on the optimal configuration for each scenario. As demonstrated in this chapter, the existing robustness indicators cannot be used in a similar fashion when evaluating a candidate solution to an ARMOP. The fact that each candidate solution has a set of optimal configurations for every realisation of the uncertainties, requires new methods for evaluation.

To address this challenge, the requirements for a robustness indicator for ARMOPs were listed, and guidance on how to construct such an indicator were provided. The idea is to apply higher-level knowledge to quantify each Pareto front with a scalar value, and then to apply existing methods for robust optimization to evaluate the solution based on the variate of this scalar value. The choice of how to scalarise the Pareto front may have a great impact on the solution to the problem. Therefore, it must be properly tailored to the optimization problem at hand.

Four different approaches were presented in this chapter to solve an ARMOP. Each approach uses a different strategy to quantify the variate of Pareto frontiers. As a result, the optimization algorithms that can be used for each approach differ from one another. High-level descriptions of evolutionary algorithms for solving ARMOPs by each of the four approaches were presented, and a comparison between them was conducted. Finally, some recommendations for choosing the most suitable approach for solving an ARMOP were given.

The next chapter presents some case studies of AROPs from the field of engineering design. Evolutionary algorithms that follow the structure described in this chapter are used to solve the multi-objective variants of the problems. The algorithms employ Monte-Carlo sampling to represent the uncertainties and evaluate the performance variate. Since the complexity of the algorithm is proportional to the number of samples, it is evident that more efficient uncertainty quantification methods should be used. The design of efficient algorithms for solving AROPs is not covered within the scope of this thesis. It is acknowledged, however, that in order to popularise the AR methodology as an attractive tool for the design of adaptive products, efficient algorithms need to be developed.

Chapter 5

Case Studies

5.1 Introduction

The active robust optimization methodology covers a wide range of problem formulations, and can support a variety of design optimization activities. In this chapter, two applications from different fields are used to demonstrate how AROPs are formulated and solved for real-world applications. In order to focus on the methodological aspects of the framework instead of the technical issues for each application, the examples are simplified and modelled from first principles.

Section 5.2 describes an ARMOP of designing an optical table for maximum performance and minimum cost. The single-objectivisation approach is used to solve the ARMOP and to find a design that can outperform a product designed using conventional methods.

Sections 5.3 and 5.4 demonstrate how a gearbox design problem can be formulated as an AROP. The gearing ratios of the gearbox need to be selected for optimal performance over a range of load requirements. In Section 5.3 the problem considers the steady-state performance of the gearbox, as well as its cost. Since the cost is not affected by adaptation, this problem is considered as a single-objective AROP. In Section 5.4 the transient performance of the gearbox is also considered. This leads to a multi-objective formulation of the problem, and the Unary indicator approach is used to solve the ARMOP.

5.2 Optical Table

The first case study demonstrates how the ARO methodology is applied to design an adjustable *optical table*. It was first published by Salomon et al. (2014) and is given here with permission from the authors. An optical table is a platform that supports systems for optics experiments. Optics equipment often requires vibrations to be sub-wavelength (Newport Corporation, 2012), therefore the optical table has to minimize the platform motion caused by floor vibrations. The legs of an optical table usually include an isolation system (e.g., passive rubber mounts, air springs and regulated pneumatic isolators). The stiffness, damping and location of the legs affect the competency of the isolation system to absorb the floor vibrations.

There are two dominant sources of uncertainty associated with the operating conditions of optical tables:

- **Source of external vibration.** Floor vibration can be caused by a variety of sources. Some examples are street traffic, door slams, nearby machinery such as fans and air-conditioners and acoustic noise. The diverse sources for vibration are associated with a wide range of frequencies and therefore the isolation system between the floor and the platform needs to reduce the vibration's amplitude over a wide spectrum.

- **Setting up of the experimental equipment.** The surface of an optical table includes an array of mounting points to support different configurations of the experimental equipment. A well designed optical table should isolate the experiment from external vibrations regardless of the manner in which the equipment is distributed.

When setting up a new experiment, the level of uncertainty regarding the operating conditions is much smaller than its level at the stage of product design. An adaptive optical table that can accommodate the changing conditions can therefore offer a better insulation for a variety of experiments than a non-adaptive one. This section describes how such an adaptive optical table is optimized using the Active Robust Optimization framework. Two objectives are considered: vibration reduction and cost. The cost objective takes into account the effort required to adapt the table to its optimal configuration for every new experiment. Although the cost is affected by adaptation, the choice of optimal configuration is made only according to the amount of damping that can be achieved. This kind of problem leads to an ARMOP that can be solved using the *single-objectivisation* approach presented in Section 4.4.2.

5.2.1 Formulation

The case study considers a simplified planar model of an optical table. It consists of a rigid platform with an evenly distributed mass, supported by three elements: two springs and a viscous damper. The table should be suitable for various experiments,

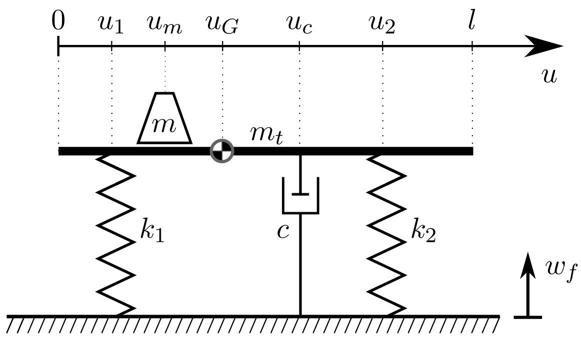


Figure 5.1: A model of an optical table. © 2014 IEEE.

and therefore, the mass of the experimental equipment and its centre position are uncertain (within known limits). The motivation is to search for optimal combinations of springs and damper and their positions, so as to: (**a**) minimize the amplitude ratio, denoted γ_a, between the displacement of the equipment's centre of mass and the floor's displacement, for a known band of vibration frequencies; and (**b**) minimize the cost, denoted γ_c. An adaptive design that can satisfy these goals is considered: an optical table with adjustable legs, that can be relocated before every new experiment. Two of the legs include springs and the third an adjustable damper (i.e. the damping coefficient can be altered with a valve).

A simplified planar model and its related parameters are depicted in Figure 5.1. The table's length is l and its mass is m_t. The experimental equipment has a total mass of m, its centre of gravity is located at u_m and its vertical displacement is denoted as w_m. The spring coefficients are k_1 and k_2, and the damping coefficient of the damper is c. The location of the i^{th} element is denoted as u_i. u_G represents the location of the system's centre of gravity, which is computed by:

$$u_G = \frac{m_t l + 2mu_m}{2(m_t + m)}. \tag{5.1}$$

The vertical displacement of the floor, denoted by w_f, is considered to be a simple harmonic motion with frequency ω. Horizontal displacement is not considered.

The aim of the AROP is to search for the best combination of spring coefficients, k_1 and k_2, for a range of experimental settings and vibration frequencies. The spring coefficient is a physical property which depends on the spring used to assemble the optical table. Therefore k_1 and k_2 are treated as decision variables of type x. The location of the legs and the damping coefficient, c, can be adjusted between experiments, and therefore they are treated as adjustable variables of type y. The parameters include the length and mass of the table, and three uncertain parameters: the equipment mass,

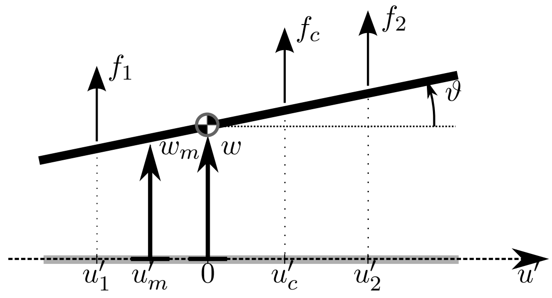


Figure 5.2: Free body diagram for the optical table. © 2014 IEEE.

m, its centre location, u_m, and the floor vibration frequency, ω. To distinguish the uncertain parameters, they are denoted as random variables M, U_m, and Ω. In the absence of information regarding the parameters' distribution functions, it is assumed that M, U_m and $\log(\Omega)$ have uniform probability distribution within their limits. The objectives vector variate that corresponds to the uncertain parameter is denoted as $\mathbf{\Gamma} = [\Gamma_\mathrm{a}, \Gamma_\mathrm{c}]$.

Figure 5.2 depicts the free body diagram of the table's surface. The horizontal coordinates are measured from u_G and are denoted as $u' = u - u_G$. The force applied to the table by component i is denoted as f_i. The grey line represents the steady-state location of the surface. As a reaction to floor vibrations, its centre of gravity is shifted by w and the whole surface is rotated by an angle ϑ.

Deriving the Amplitude Ratio

In the following, the model of the system dynamics is derived from first principles in order to compute the amplitude ratio between the floor vibration and the equipment, for a given combination of components, configuration and load setting. Assuming small angles, the model of the system can be described by the following set of equations:

$$f_1 + f_2 + f_c = (m_t + m)\ddot{w}, \tag{5.2}$$

$$f_1 u'_1 + f_2 u'_2 + f_c u'_c = j\ddot{\vartheta}, \tag{5.3}$$

$$\text{where :}\quad f_{1,2} = k_{1,2}(w_f - w_{1,2})\,, \tag{5.4}$$

$$f_c = c(\dot{w}_f - \dot{w}_c)\,, \tag{5.5}$$

$$w_{1,2} = w + \vartheta u'_{1,2}, \tag{5.6}$$

$$\dot{w}_c = \dot{w} + \dot{\vartheta} u'_c, \tag{5.7}$$

$$j = \frac{3m_t m(2u_m - l)^2 + m_t(m_t + m)\,l^2}{12(m_t + m)}. \tag{5.8}$$

$\dot{\square}$ and $\ddot{\square}$ denote the first and second time derivatives, respectively. Equations 5.2 and 5.3 are Newton's second law for vertical translation and rotation, respectively. Equations 5.4 and 5.5 describe the forces applied by the springs and damper, and Equations 5.6 and 5.7 describe geometric relations. Equation 5.8 describes how to calculate the moment of inertia, j[1], around the centre of gravity.

Equations (5.2)–(5.7) can be written in a matrix form as follows:

$$[M]\ddot{\mathbf{z}}+[C]\dot{\mathbf{z}}+[K]\mathbf{z}=[A]\mathbf{v}, \tag{5.9}$$

$$\begin{aligned}
\text{where}:\ \mathbf{z}\ &=\left[w(t),\vartheta(t)\right],\\
\mathbf{v}\ &=\left[w_f(t),\dot{w}_f(t)\right],\\
[M]&=\begin{pmatrix} m+m_t & 0\\ 0 & j\end{pmatrix},\\
[C]&=\begin{pmatrix} c & cu'_c\\ cu'_c & cu'^2_c\end{pmatrix},\\
[K]&=\begin{pmatrix} k_1+k_2 & k_1u'_1+k_2u'_2\\ k_1u'_1+k_2u'_2 & k_1u'^2_1+k_2u'^2_2\end{pmatrix},\\
[A]&=\begin{pmatrix} k_1+k_2 & c\\ k_1u'_1+k_2u'_2 & cu'_c\end{pmatrix}.
\end{aligned}$$

Assuming zero initial conditions, a matrix of transfer functions between $\mathbf{V}(s)=\mathcal{L}(\mathbf{v})$ and $\mathbf{Z}(s)=\mathcal{L}(\mathbf{z})$ may be obtained by performing a Laplace transform on both sides of (5.9):

$$\begin{aligned}
[G(s)]=\begin{pmatrix} G_{11} & G_{12}\\ G_{21} & G_{22}\end{pmatrix}&=\frac{\mathbf{Z}(s)}{\mathbf{V}(s)}\\
&=\left([M]s^2+[C]s+[K]\right)^{-1}[A].
\end{aligned} \tag{5.10}$$

A transfer function between the equipment's displacement, $W_m(s)$, and the floor's displacement, $W_f(s)$, can be obtained by recalling that $W_m(s)=Y(s)+u'_m\Theta(s)$, and

[1] In order to be consistent with the notation used throughout this thesis, lower-case j is used instead of the more common upper-case J to denote moment of inertia. As explained in Section 2.3.1, lower-case symbols denote deterministic values, while upper-case symbols denote random values. The same rules apply to other notations in this chapter.

$\mathcal{L}(\dot{w}_f(t)) = sW_f(s)$:

$$G(s) = \frac{W_m(s)}{W_f(s)} = G_{11} + sG_{12} + u'_m(G_{21} + sG_{22})\,. \tag{5.11}$$

Finally, the amplitude ratio between the equipment's displacement and the floor's displacement, when it vibrates at a frequency ω, is the norm of the transfer function in (5.11):

$$\gamma_{\mathrm{a}} = \|G(jw)\|_2\,. \tag{5.12}$$

The Cost Function

A cost function of an adaptive solution, $\gamma_{\mathrm{c}}(\mathbf{x}, \mathbf{y}, \mathbf{p})$, can consist of any of the following three components: (**a**) the initial implementation costs, denoted by $c_{\mathbf{x}}$, (**b**) the operational costs of using the design in a configuration $\mathbf{y}$, denoted by $c_{\mathbf{y}}$, and (**c**) the costs of the adaptations of a design as a reaction to changes in $\mathbf{p}$, denoted by $c_{\mathbf{p}}$. The cost function used in this case study is based on the following assumptions:

- The implementation cost, $c_{\mathbf{x}}$, does not consider costs that are identical for all different solutions. Therefore, it is a function of the solution's selected springs. For a given load, a small spring coefficient demands a larger spring (either more coils or a larger diameter), which is also more expensive. Considering the above, the implementation cost function of the product is:

$$c_{\mathbf{x}} := \frac{log(k^l)}{log(k_1)} + \frac{log(k^l)}{log(k_2)}, \tag{5.13}$$

  where k^l is the lower limit of the spring coefficient (most expensive).

- The configuration in which the design operates does not affect the cost. Therefore, $c_{\mathbf{y}} = 0$.

- The energy (and its associated cost) required to move the springs and damper is relative to the distances travelled. The damping coefficient's adjustment is a simple action of turning a knob and therefore it does not have a cost. The adaptation cost $c_{\mathbf{p}}$ between two optimal states, $\mathbf{y}_i^\star$ and $\mathbf{y}_j^\star$, is proportional to the difference between the two configurations:

$$c_{\mathbf{p}} := \mathbf{c_a} \left|\mathbf{y}_i^\star - \mathbf{y}_j^\star\right| \tau, \tag{5.14}$$

  where $\mathbf{c_a} = [0, 0.3, 0.3, 0.12]$ is a vector containing the costs of adjusting each variable per unit of change, and $\tau = 100$ is the expected number of adaptations during the lifetime of the product. Since the optimal configuration, $\mathbf{Y}^\star$, is a random vector that relies on the variate, $\mathbf{P}$, the adaptation cost can also be considered as a random variate:

$$C_{\mathbf{P}} := \mathbf{c_a} \left| \mathbf{Y}_i^\star - \mathbf{Y}_j^\star \right| \tau. \tag{5.15}$$

Thus, the cost function, Γ_{c}, is:

$$\Gamma_{\mathrm{c}}(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}) = c_{\mathbf{x}} + C_{\mathbf{P}}. \tag{5.16}$$

The cost function depends on the uncertain variables and the optimal configurations, which makes the optimization problem an ARMOP. However, it is reasonable to assume that the optimal configuration will always be chosen according to the ability of the table to reduce vibrations, regardless of the labour costs associated with adjusting the legs.

The Active Robust Optimization Problem

Using the above notations, the AROP for an adaptive optical table is formulated as the following single-objectivised ARMOP:

$$\min_{\mathbf{x} \in \mathcal{X}} \mathbf{\Gamma}(\mathbf{x}, \mathbf{P}) = \left[\Gamma_{\mathrm{a}}(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}), \Gamma_{\mathrm{c}}(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P})\right] \tag{5.17}$$

$$\text{where:} \quad \mathbf{Y}^\star = \underset{\mathbf{y} \in \mathcal{Y}(\mathbf{x})}{\operatorname{argmin}} \, \Gamma_{\mathrm{a}}(\mathbf{x}, \mathbf{y}, \mathbf{P}) \tag{5.18}$$

$$\mathbf{x} = [k_1, k_2] \tag{5.19}$$

$$\mathbf{y} = [c, u_1, u_2, u_c] \tag{5.20}$$

$$\mathbf{P} = [M, U_m, \Omega, m_t, l] \tag{5.21}$$

The parameters' values and the limits of search variables and uncertainties are given in Table 5.1.

Robustness Criteria for the ARMOP

Due to the high sensitivity of the optics equipment, the amplitude ratio is considered to be its worst-case over all sampled realizations of the uncertainties:

$$I_w[\Gamma_{\mathrm{a}}] := \max_{\mathbf{P}} \Gamma_{\mathrm{a}}(\mathbf{x}, \mathbf{Y}^\star, \mathbf{P}). \tag{5.22}$$

The uncertain cost function can be scalarised by adding the deterministic initial

Table 5.1: Variables and parameters of the optical table optimization problem. © 2014 IEEE.

Type	Symbol	Units	Lower limit	Upper limit
$\mathbf{x}$	k_1, k_2	N/mm	1	100
$\mathbf{y}$	c	N·s/mm	1	10
	u_c	m	0.1	1.9
	u_1	m	0.1	0.9
	u_2	m	1.1	1.9
$\mathbf{p}$	m	kg	20	50
	u_m	m	0.1	1.9
	ω	rad/s	1	10^4
	l	m	2	
	m_t	kg	200	

implementation cost, c_x, to the expected overall adaptation cost, $\mathrm{E}(C_{\mathbf{P}})$. Hence, the expected value of Γ_c can be used in the following manner: For a sampled set of the uncertain vector $\overline{\mathbf{P}}$ with k samples, the adaptation cost considers all possible adaptations between two optimal states $\mathbf{y}_i^\star$ and $\mathbf{y}_j^\star$ that belong to the sampled set $\overline{\mathbf{Y}^\star}$:

$$I_E[\Gamma_c] := c_x + \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} \mathbf{c_a} \left| \mathbf{y}_i^\star - \mathbf{y}_j^\star \right|}{k(k-1)} \tau \tag{5.23}$$

Given the above robustness criteria, Equation 5.17 of the ARMOP is solved as a multi-objective problem with the objective vector

$$\boldsymbol{I}[\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{P})] = \left[I_w[\Gamma_\mathrm{a}] , I_E[\Gamma_\mathrm{c}] \right] . \tag{5.24}$$

5.2.2 Simulations and Results

The ARMOP in Equations (5.17)–(5.21) is solved with an EA as suggested in Algorithm 4.2. The MOP in (5.17) is solved using NSGA-II-PSA (Salomon et al., 2013b) with a population size of $n = 100$ for 50 generations. All parameters are set according to the values suggested in Salomon et al. (2013b) (real-coded chromosome, SBX crossover and polynomial mutation with distribution indices $\eta_c = 15$ and $\eta_m = 20$, respectively, crossover probability $p_c = 1$ and mutation probability $p_m = 0.2$). The DOP in (5.18) is solved using a single-objective genetic algorithm with the same crossover, mutation, population size and the number of generations as mentioned above. Two features were added to this algorithm in order to reduce the number of function evaluations:

1. For every new sampled scenario in Algorithm 4.2, Steps 5–7, the existing sample of $\mathbf{Y}^\star$ is added to the random initial population of the EA. This enables faster conver-

gence towards the optimal configuration when it is close to another configuration already found for a similar scenario.

2. A stopping criterion based on improvement of the objective function is added to the generations' count criterion. In the event that no improvement is made over 20 generations, the inner algorithm ends.

The real-coded algorithm was directly implemented in MATLAB. In practice, the above features helped in speeding up the algorithm by approximately 100%, where a complete optimization run could be completed on a standard desktop in about five hours instead of ten.

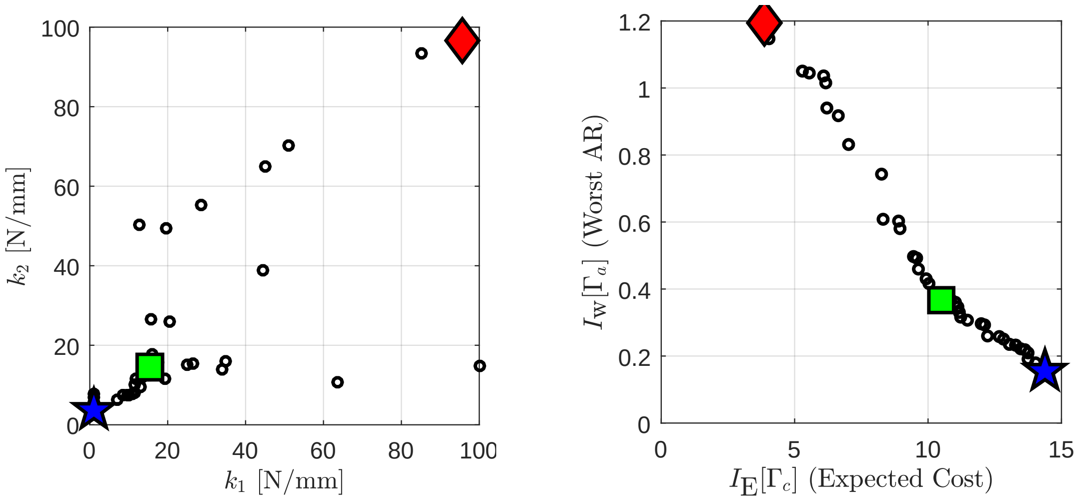
First results indicated a strong correlation between the highest amplitude ratio and the lower limit of the equipment's mass. As a result, since the worst-case scenario is considered, the value of the mass was taken as its lower limit $m = 20$kg. The sampled set $\bar{\mathbf{P}}$ consists of $k = 5{,}000$ samples, distributed according to the PDFs of u_m and ω with the Latin hypercube sampling method (McKay et al., 1979).

The final approximated set and approximated front are depicted in Figure 5.3. The results indicate that softer springs achieve better performance in reducing the reaction to the floor's vibrations, but they are more expensive. Interestingly, the solution with the best damping is not the one with the smallest value of spring coefficient for both springs, but a solution with $k_1 = 1\frac{\mathrm{N}}{\mathrm{mm}}$ and $k_2 = 3.5\frac{\mathrm{N}}{\mathrm{mm}}$. This difference is shown to decrease the equipment's displacement better than two springs with the same coefficient of $1\frac{\mathrm{N}}{\mathrm{mm}}$.

The amplitude ratio and adjustments of three solutions as a function of u_m are depicted in Figure 5.4. These three solutions are highlighted in Figure 5.3 as a square, a star and a diamond. Figure 5.4(a)–5.4(c) depict the amplitude ratio and optimal locations of springs and damper for each of the three solutions, and Figure 5.4(d) depicts the optimal damping coefficients. Note that solutions with stiffer springs, in addition to their lower cost, require smaller adjustments to changes in location of the experimental equipment. Another interesting observation is that the optimal adjustments of the damper and components' locations for the star related solution are not symmetric. This is a consequence of the differences between its springs.

In order to assess the reliability of the obtained approximated front, the problem was solved for twenty independent runs of the EA. The statistics for the solution with the best amplitude ratio (marked with a star in Figure 5.3) is depicted in Figure 5.5.

To check the added value of adaptation to the performance of the adaptive optical table, it is compared with a similar design that is not adaptive. This design possesses the same characteristics, but it cannot be changed once implemented (i.e., all its variables are



(a) The obtained approximated Pareto-optimal set. The spring coefficients of the three highlighted solutions are: $\mathbf{x}^{\bigstar} = [1, 3.7]\,\frac{\mathrm{N}}{\mathrm{mm}}$, $\mathbf{x}^{\blacksquare} = [15, 14]\,\frac{\mathrm{N}}{\mathrm{mm}}$, and $\mathbf{x}^{\blacklozenge} = [96, 96]\,\frac{\mathrm{N}}{\mathrm{mm}}$.

(b) The obtained approximated Pareto front. The solution with the smallest amplitude ratio is marked with a star, the solution with the lowest cost is marked with a diamond, and a trade-off solution is marked with a square.

Figure 5.3: Final approximated set and Pareto front after 50 generations of the evolutionary algorithm. © 2014 IEEE.

of Type $\mathbf{x}$). The costs are not considered for this comparison, and the only consideration is minimum amplitude ratio. The optimal non-adaptive product is found by solving the following worst-case RO problem:

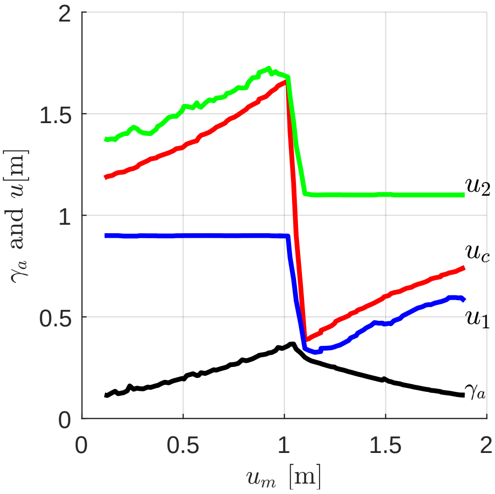
$$\mathbf{x}_{\mathrm{na}} = \underset{\mathbf{x}\in\mathcal{X}}{\operatorname{argmin}} \max_{\mathbf{p}\in\mathbf{P}} \Gamma_{\mathrm{a}}(\mathbf{x}, \mathbf{P}) \tag{5.25}$$

where $\mathbf{x} = [k_1, k_2, c, u_c, u_1, u_2]$, $\mathbf{P} = [M, U_m, \Omega, m_t, l]$. The search was conducted with the same genetic algorithm as for the AROP.

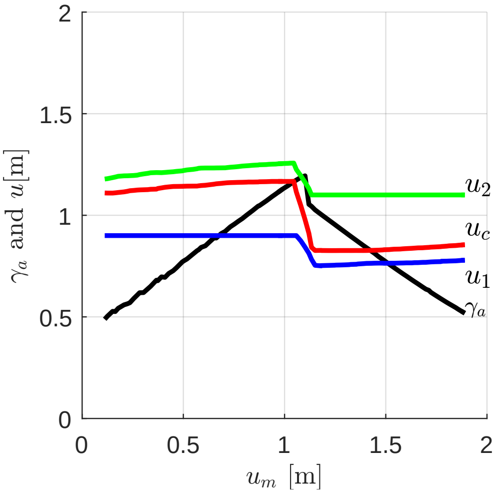
The solution to the problem in (5.25) is $\mathbf{x}_{\mathrm{na}} = [1, 1, 1, 1, 0.9, 1.1]$, i.e., both springs and the damper are the weakest possible, the damper is located at the centre of the table, and the springs are as close to the centre as possible. The worst amplitude ratio for this configuration occurs when the centre of mass aligns with the centre of the table, and its value is $I_w[\Gamma_{\mathrm{a}}(\mathbf{x}_{\mathrm{na}})] = 0.456$. This value is three times larger than the worst amplitude ratio of the adaptive solution $\mathbf{x}^{\bigstar}$, which is $I_w\left[\Gamma_{\mathrm{a}}\left(\mathbf{x}^{\bigstar}\right)\right] = 0.15$.

5.2.3 Discussion

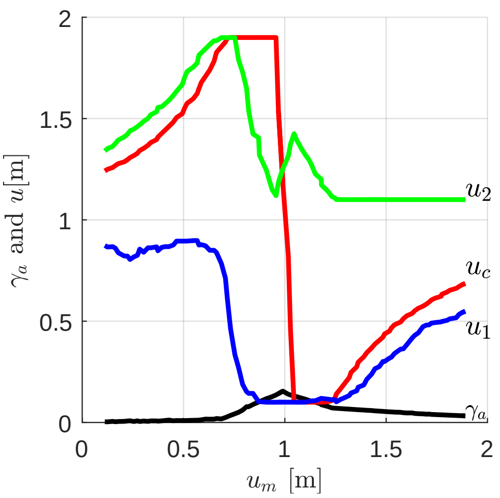
This case study demonstrated how the active robustness methodology can be applied to improve existing concepts in engineering design. An adaptive version of an optical table was suggested as an improvement of the existing design. While the suspension



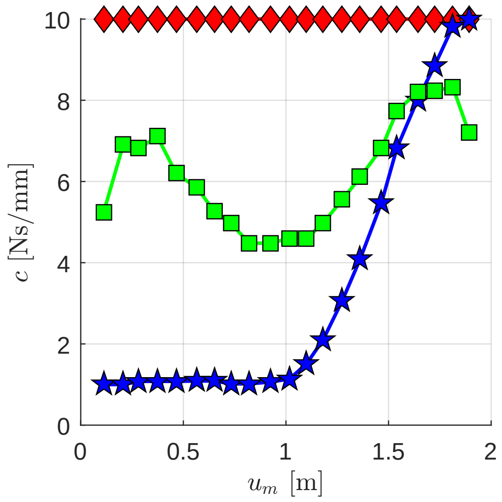
(a) Amplitude ratio and locations of the components of $\mathbf{x}^{\blacksquare}$ for the various location of the equipment.



(b) Amplitude ratio and locations of the components of $\mathbf{x}^{\blacklozenge}$ for the various location of the equipment.

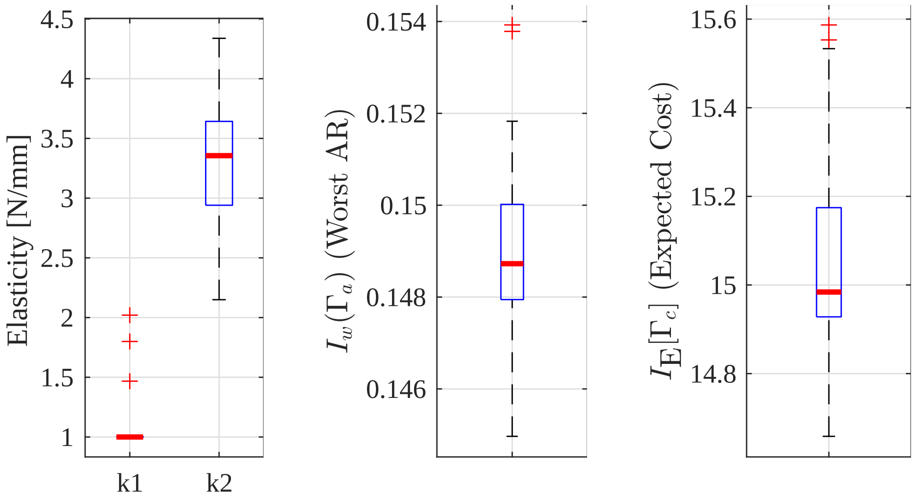


(c) Amplitude ratio and locations of the components of $\mathbf{x}^{\bigstar}$ for the various location of the equipment.



(d) Damping coefficients for every location of the equipment. The values for each solution are marked with the solution's associated shape.

Figure 5.4: The amplitude ratio and optimal configurations of the highlighted three solutions. © 2014 IEEE.



(a) Design space (b) Amplitude ratio (c) Cost

Figure 5.5: Box-plots for the results of the obtained adaptive solution with the best amplitude ratio, from 20 independent simulations of the EA. © 2014 IEEE.

system of the conventional optical table is made of fixed legs, the legs of an adaptive table can move to better absorb floor vibrations for a given experimental setting. The ARO methodology provides the missing tools for optimizing this kind of product to exploit the full capacity of its adaptability.

A generic cost function for adaptive products was introduced in this chapter. Cost may not be always the main criterion when designing a product, but it is always a concern. An adaptive product is associated with three types of cost: *manufacturing* costs, *operational* costs and *adaptation* costs. The manufacturing costs include all the costs associated with producing the product. When formulating an AROP cost function there is no need to consider costs that are identical for all solutions, and the emphasis needs to be given to the differences between solutions, reflected in the solution, $\mathbf{x}$, such as component prices, assembly costs, etc. Manufacturing costs do not depend on the operating conditions, and therefore they do not need to be considered when searching for the optimal configuration, $\mathbf{y}^{\star}$. Operational costs are the costs of operating the product in a given configuration, $\mathbf{y}$. They depend on the realisation of the uncertain parameters, and their value is part of the inner optimization problem of the AROP, to find the optimal configuration. Adaptation cost is the cost of changing the product from one configuration to another in response to a change in the uncertain parameters. When calculating adaptation cost, one needs to approximate which adaptations will take place (i.e., which changes will occur in the realisations of the uncertain parameters).

In many applications the adaptability of the product can be assumed to be fully exploited by the user in order to improve performance. In this case study, the adaptation to a configuration with better damping increases the cost function. A realistic assumption is that if a customer is willing to invest in an adaptive product, he/she will use its adaptive capabilities regardless of the adaptation and operational costs. If a single performance objective exists, in addition to cost, the single-objectivisation is a natural choice to solve the ARMOP. At any realisation of the uncertainties, the configuration is determined according to optimal performance. The cost can be considered as a secondary objective for the outer optimization problem.

A nested evolutionary algorithm was used to solve the ARMOP in this case study. The algorithm followed the guidelines to construct a single-objectivisation EA from Chapter 4. By their nature, active robust optimization problems require a very large number of function evaluations due to the nested nature of the problem. The case study highlights the need to tailor the optimization algorithm to the problem in order to reduce the number of function evaluations, while still converging close enough to the true ARMOP solution. The fact that the objective functions could be derived from first principles as relatively straightforward expressions enabled a Monte-Carlo based approach to quantify the uncertain objective functions and conduct a large number of function evaluations. In applications where the objective functions are more computationally expensive, some sacrifices have to be made, either regarding the fidelity of the evaluation by using surrogate methods, or the convergence of the algorithm by conducting fewer function evaluations.

The algorithm was able to find a solution to the ARMOP – a set of solutions that offer a trade-off between cost and performance. Although the main consideration when using the adaptive optical table is vibration damping, some costumers might not require the best performance for their application, and can settle for a less expensive product. The configurations of all solutions followed a similar pattern in adapting to changes in the equipment position. The components generally stay away from the equipment's centre of gravity, while the distance depends on stiffness of the springs.

When the problem is solved for a similar non-adaptive design without considering costs, the optimum design contains the softest springs and damper. Compared with this solution, the adaptive design with best vibration damping was found to be three times better at the worst-case. Interestingly, the optimum solution consists of springs with different elasticity. This asymmetry allows the system to operate efficiently at different frequencies by changing the position of the springs according to the experimental setting. The ability of the proposed algorithm to converge to the true solution was tested by solving the problem for twenty independent runs. The same solution where one spring is at the lower limit of the elasticity range and the other is about 3.5 times stiffer, was repeatedly found.

5.3 Gearbox Design - Single-Objective Formulation

This section and Section 5.4 apply the Active Robustness methodology to a very common problem in engineering design–the choice of gearing ratios for a geared system. This section addresses the performance of a gearbox in steady-state operating conditions, and the next section extends the problem to also consider dynamic performance.

In Section 5.3.1 the motivation and relevant background is presented. Section 5.3.2 includes a mathematical model of the geared system. The AROP is formulated in Section 5.3.3. The complexity of the AROP is small enough to explore the entire design space and find the optimal set of solutions without relying on optimization algorithms. The solution is presented in Section 5.3.4. A sensitivity analysis of the AROP and its solution to several aspects of the problem formulation is given in Section 5.3.5.

5.3.1 Background

One of today's engineers greatest challenges is the development of energy efficient products to cope with limited resources and reduce ecological footprint. In systems that include a gearbox, careful design of this component can enhance the efficiency of the system. A gearbox is an assembly of gears with different ratios that provides speed and torque conversions from a motor to another device. With the use of a gearbox, a single motor can meet a span of load demands, which are combinations of required speed and torque. There is a unique gearing ratio for every given motor that will result in the least energy consumption for a specific load demand. Usually a geared system operates under a range of possible loads. If optimality with respect to energy consumption is targeted, the gearbox should include an infinite number of gears in order to accommodate all loads within this range. Naturally it is not possible to produce such a gearbox, and anyway, a gearbox with too many gears has more drawbacks than advantages (e.g. dimensions, weight, costs). Therefore, gearboxes used in real applications are made of a finite number of gears (typically up to six in the auto industry), where each gear covers a different range of the load demands (e.g. high reduction for high torque and low speed, and vice versa). The gearbox's gearing ratios should allow for the satisfaction of each possible load by one of the gears in a reasonably efficient manner. Therefore, the choice of the gears determines the overall performance of the gearbox. This choice can be supported by an optimization procedure for minimum energy consumption.

Some previous studies on gearbox optimization can be found in the literature. Guzzella and Amstutz (1999) presented a computer aided engineering tool for modelling and optimization of a hybrid vehicle. They showed an example of optimizing the transmission ratios for minimum fuel consumption. The model is deterministic, and the ratios are optimized for a single, arbitrarily chosen, load cycle. Roos et al. (2006) suggested an optimization procedure for selecting a motor and gearhead for mechatronic applications to maximize one of the following objectives: peak power, output torque or energy efficiency. This approach is suitable for a single gear system and not for a gearbox with several gears. The choice of the gearhead was conducted according to the worst-case of the expected load scenarios. Swantner and Campbell (2012) developed a framework for gearbox optimization that searches among different types of gears (helical, conic, worm, etc.), topologies, materials and sizing parameters. The gearbox was optimized for minimum dimensions, considering a set of functional constraints. Other problem settings for single-objective gearbox optimization include minimum variation from a given set of transmission ratios (Mogalapalli et al., 1992), minimum volume or weight (Yokota et al., 1998; Savsani et al., 2010), minimum vibration (Inoue et al., 1992) and minimum centre distance between input and output shafts (Li et al., 1996).

Some multi-objective gearbox optimization studies can also be found in the literature. Osyczka (1978) formulated a problem to minimize simultaneously four objective functions: volume of elements, peripheral velocity between gears, width of gearbox, and centre distance. Wang (1994) considered centre distance, weight, tooth deflection, and gear life as objective functions. Thompson et al. (2000) optimized for minimum volume and surface fatigue life. Kurapati and Azarm (2000) optimized a gearbox for minimum volume and minimum stress in the output shaft. Deb et al. (2000) designed a compound gear train to achieve a specific gear ratio. The objectives of the gear train design were minimum error between the obtained gear ratio and the required gear ratio and maximum size of any of the gears. Deb and Jain (2003) have optimized an 18-speed, 5-shafts gearbox for two, three and four objectives. Among the objectives were power, volume, centre distance and variation from desired output speed. The same optimization problem was used by Deb (2003) to demonstrate how design principles can be extracted by investigating the relations between design variables of the Pareto-optimal solutions in the design space. Li et al. (2008a) optimized a two-stage gear reducer for minimum dimensions, minimum contact stress and minimum transmission precision errors.

The optimization involved within all studies above was conducted for given reduction ratios, or at least for a given speed-torque scenario or cycle. However, most applications that include a gearbox (such as vehicles) are subjected to a large span of uncertain load requirements, as a result of a variety of possible environmental conditions. The stochastic nature of the required torque and speed must be considered during the design phase. In order to optimize a gearbox for uncertain load requirements, a robust optimization procedure should be considered. In this case study, a gearbox is optimized for minimum energy consumption where the load demand is uncertain. A robust set of transmission ratios is sought to maximize the system's efficiency taking account of the uncertain load domain.

A gearbox cannot be optimized for robustness using conventional robust optimization, since its performance does not solely depend on its preliminary design. The performance is also influenced by the manner in which the gearbox is being operated. A gearbox with a good selection of gearing ratios for a span of load scenarios can be very inefficient if it is not being used properly. For best performance, the proper gear in the set has to be selected for each realization of the uncertain load demand. When cruising on the highway, the best efficiency is achieved with the highest gear (say, sixth). A driver that uses the fifth gear for this scenario does not operate the gearbox in an optimal manner. Hence, robustness to the uncertain load demand is actively attained by selecting the proper gear for each load scenario. The selection of the optimal gear for each scenario can be made either manually by a skilled user, or with the use of a controller in the case of an automatic transmission.

The active robustness methodology provides the precise tools required for optimizing a gearbox. The adaptability of a geared system is provided by the user's ability to change the gearing ratio by using a different gear. This adaptability is taken into account at the evaluation of a candidate solution; it is evaluated according to its best possible performance for each scenario of the uncertain parameters. For the example above, it is assumed that the driver uses the sixth gear while cruising on the highway and second gear when carrying a heavy load up the hill. Since enhanced adaptability usually comes with a price (e.g., a gearbox with more gears would be more expensive), the objectives of the AROP are the solution's best possible performance, evaluated at different scenarios of the uncertainties involved, and its cost.

The problem formulated in this section is the optimization of a gearbox for a random variate of torque and speed requirements. Both the number of gears and their characteristics are optimized in order to minimize the overall energy consumption and gearbox cost. The solution to the problem is a set of gearboxes with a trade-off between energy efficiency and low cost. The methodology is demonstrated with a power system of an electric motor and a simple two-stage reduction gearbox. This example can be adapted to design other geared systems such as vehicles, motorcycles, wind turbines, industrial and agricultural machinery.

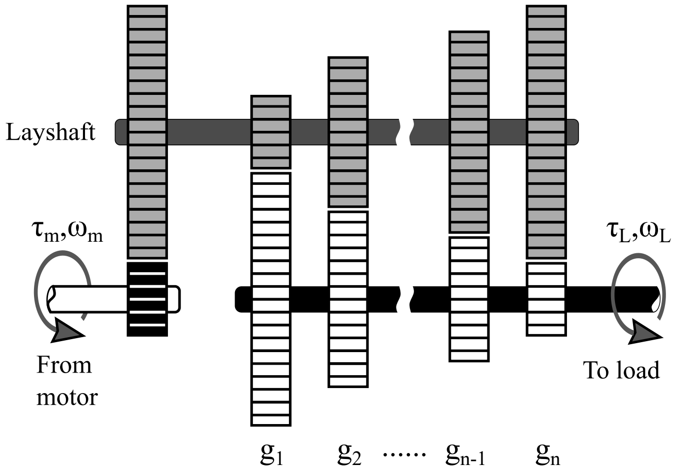


Figure 5.6: A gearbox with n gears. All gears are rotating while at any given moment the power is transmitted through one of them.

5.3.2 Motor and Gear System

The problem at hand is the optimization of a gearbox for a span of torque-speed scenarios. An electric motor of type Maxon A-max 32 is to convey a torque τ_L at speed ω_L. In order to do so, it is coupled with a gearbox, as shown in Figure 5.6. The motor's output shaft (white) rotates at speed ω_m and transmits a torque τ_m. It is firmly connected to a cogwheel (black) that is constantly coupled to the layshaft. The layshaft consists of a shaft and n_g gears (grey), rotating together as a single piece. n_g gears (white) are also attached to the load shaft (black) with bearings, so they are free to rotate around it. The gears are constantly coupled to the layshaft and rotate at different speeds, depending on the gearing ratio. A collar (not shown in the figure) is connected, through splines, to the load shaft and spins with it. It can slide along the shaft to engage any of the gears, by fitting teeth called "dog teeth" into holes on the sides of the gears. In this manner the power is transferred to the load through a certain gear, with the desired reduction ratio.

As noted above, the aim of this study is to optimize the gearbox to achieve good performance over a variety of possible load scenarios. Several objectives might be considered: monetary costs, energy efficiency for different loads and the transient behaviour of the gearbox (e.g. energy consumption during speed transitions and time required to change the system's speed). A problem formulation that considers all of the aforementioned objectives will be addressed in Section 5.4. In order to demonstrate the features and concerns of the active robustness approach, this section focuses on a more restricted formulation of the gearbox optimization problem. Therefore, only the steady-state behaviour of the gearbox is addressed at this stage.

The number of gears in the gearbox, n_g, and the number of teeth in each i^{th} gear, z_i, are to be optimized. The objectives considered are minimum energy consumption and minimum manufacturing cost of the gearbox. The system is evaluated at steady-state, i.e., operating at the torque-speed scenarios. The power required for each scenario is considered, while the objective is to find a set of gears that will require the minimum average invested power over all scenarios. For every scenario, the gearbox is evaluated by the smallest possible value of input power. This value is achieved by transmitting the power through the most suitable gear in the box.

Model Formulation

In this section, the model for the motor and gearbox system is presented according to Krishnan (2001), and the required performance measures are derived.

The motor armature current can be described by applying Kirchhoff's voltage law over the armature circuit:

$$V = L\dot{I} + rI + k_v\omega_m, \tag{5.26}$$

where V is the input voltage, L is the coil inductance, I is the armature current, r is the armature resistance and k_v is the velocity constant.[1] The ordinary differential equation describing the motor's angular velocity as related to the torques acting on the motor's output shaft is:

$$j_m\dot{\omega}_m = k_t I - b_m\omega_m - \tau_m, \tag{5.27}$$

where j_m is the rotor's inertia, k_t is the torque constant and b_m is the motor's damping coefficient associated with the mechanical rotation. Since this study only deals with the gearbox's performance at steady-state, the derivatives of I and ω_m are considered to be zero.

There are two speed reductions between the motor and the load. The first is from the motor shaft to the layshaft. This reduction ratio, denoted as n_1, is z_l/z_m, where z_m is the number of teeth in the motor shaft cogwheel and z_l is the number of teeth in the layshaft cogwheel. The second reduction, denoted as n_2, is from the layshaft to the load shaft. Each gear on the load shaft rotates at a different speed according to its gearing ratio $n_{2,i} = z_{g,i}/z_{l,i}$, where $z_{g,i}$ is the number of teeth of the i^{th} gear's load shaft cogwheel and $z_{l,i}$ is the number of teeth of its matching layshaft wheel. n_2

[1] V, I and L are the universal notations to describe voltage, current and inductance. For clarity, these are used here to describe deterministic values, in contrast to the usual convention of this thesis where capital letters are used to describe random variates.

depends on the selected gear, and it can be one of the values $\{n_{2,1}, \ldots, n_{2,n_g}\}$. The total reduction ratio from the motor to the load is $n = n_1 * n_2$, and the load speed $\omega = \omega_m/n$. The motor and load shafts are coaxial, and the modules for all cogwheels are identical. Therefore, the total number of teeth z_t for each gearing couple is identical:

$$z_t = z_l + z_m = z_{g,i} + z_{l,i} \quad , \quad \forall i \in 1, \ldots, n_g. \tag{5.28}$$

At steady-state, Equation (5.27) can be reflected to the load shaft as follows:

$$0 = nk_tI - \left(b_g + n^2 b_m\right)\omega - \tau, \tag{5.29}$$

where τ is the load's torque and b_g is the gear's damping coefficient with respect to the load's speed.

If ω from Equation (5.29) is known, the armature current can be derived:

$$I = \frac{\left(b_g + n^2 b_m\right)\omega + \tau}{nk_t}. \tag{5.30}$$

Once the current is known, and after neglecting $\dot{I}$, the required voltage can be derived from Equation (5.26):

$$V = rI + nk_v\omega. \tag{5.31}$$

The invested electrical power is:

$$s = VI. \tag{5.32}$$

Manufacturing costs are the only kind of cost applicable to this problem. It is conceivable that they depend on the number of wheels in the gearbox, their size, and overheads. A function of this type is suggested for this generic problem to demonstrate how the various costs can be quantified:

$$c = \alpha {n_g}^{\beta} + \lambda \sum_{i=1}^{n_g} \left({z_{l,i}}^2 + {z_{g,i}}^2\right) + \delta, \tag{5.33}$$

where α, β, λ and δ are constants. The first term considers the number of gears. It takes into account their influence on the costs of components such as the housing and shafts. The second term relates to the cogwheels' material costs, which are proportional to the square of the number of teeth in each wheel. The third represents the overheads. In practice, other cost functions could be used.

5.3.3 Problem Formulation

The gearbox optimization problem, formulated as an AROP, is the search for the number of gears n_g and the number of teeth in each gear $z_{g,i}$ that minimize the production cost, c, and the power input, s. The variables are sorted into three vectors:

- $\mathbf{x}$ is a vector with the variables that define the gearbox, namely the number of gears and their number of teeth. These variables can be selected before the gearbox is produced, but cannot be altered by the user during its life cycle. The variables in $\mathbf{x}$ are the problem's design variables.

- $\mathbf{y}$ is a vector with the adjustable variables. It includes the variables that can be adjusted by the gearbox's user: the selected gear, i, and the supplied voltage, V. The decisions how to adjust these variables are made according to the load's demand, and can be supported by an optimization procedure. For example, a high reduction ratio will be chosen for low speed, and a low ratio for high speeds, while the voltage is adjusted to maintain the desired velocity for the given torque.

- $\mathbf{p}$ is a vector with all the environmental parameters that affect performance and are independent of the design variables. Some of the parameters in this problem are considered as deterministic, but some possess uncertain values. The uncertainty for ω and τ is aleatory, since they inherently vary within a range of possible load scenarios. The random variates of ω and τ are denoted as Ω and $\mathcal{T}$, respectively. Some values of the motor parameters are given tolerances by the supplier. The terminal resistance, r, has a tolerance of 5% and the motor damping coefficient, b_m, has a tolerance of 10%. Additionally, the gearbox damping coefficient, b_g, can be only estimated, and therefore it is treated as an epistemic uncertainty. The random variates of r, b_m and b_g are denoted as R, B_m and B_g, respectively. The resulting variate of $\mathbf{p}$ is denoted as $\mathbf{P}$.

A certain load scenario might have more than one feasible $\mathbf{y}$ configuration. When the gearbox (with the gearing ratios represented by $\mathbf{x}$) is evaluated for each scenario, the optimal configuration (the one that requires the least input power) is considered. This configuration is denoted as $\mathbf{y}^\star$, and it consists of the optimal transmission, i, and input voltage, V, for the given scenario. The variate of optimal configurations that correspond to the variate $\mathbf{P}$ is termed as $\mathbf{Y}^\star$.

Two objectives are considered for optimization: the electric power invested at the motor, s, and the manufacturing costs, c. The electric power is affected by the uncertain parameters and the configuration of the gearbox. Therefore, the input power objective is denoted as γ_s. The manufacturing costs only depend on the design, and therefore the cost objective is denoted as ψ_c.[1]

Following the above, the AROP is formulated:

$$
\begin{aligned}
& \min_{\mathbf{x}\in\mathcal{X}} \boldsymbol{\Gamma}(\mathbf{x},\mathbf{P}) = \left\{\Gamma_s(\mathbf{x},\mathbf{Y}^\star,\mathbf{P}), \psi_c(\mathbf{x})\right\}, \\
& \mathbf{Y}^\star = \operatorname*{argmin}_{\mathbf{y}\in\mathcal{Y}(\mathbf{x})} \Gamma_s(\mathbf{y},\mathbf{P}), \\
\text{subject to}: \quad & I \leq I_{nom}, \\
& z_{g,i} + z_{l,i} = z_t \quad , \quad \forall i = 1,\dots,n_g, \\
\text{where}: \quad & \mathbf{x} = \left[n_g, z_{g,1},\dots,z_{g,i},\dots,z_{g,n_g}\right], \\
& \mathbf{y} = [i, V], \\
& \mathbf{P} = [\Omega, \mathcal{T}, R, B_m, B_g, k_v, k_t, I_{nom}, n_1, z_t, \\
& \qquad \alpha, \beta, \lambda, \delta].
\end{aligned}
\tag{5.34}
$$

The constraints are evaluated according to Equations (5.30) and (5.31), and the objectives according to Equations (5.32) and (5.33). I_{nom}, the nominal current, is the highest continuous current that does not damage the motor. It is significantly smaller than the motor's stall current.

The problem (5.34) is a bi-objective problem, but only γ_s is affected by adaptation. As a result, there is a single optimal configuration for every realisation of the uncertain parameters, which classifies the problem as a single-objective AROP. The mean value criterion, I_{E}, is a reasonable choice to assess the distribution of Γ_s, as it captures the efficiency of the gearbox when it operates over the entire range of expected load scenarios. In the following sections, the performance of each candidate design is evaluated according to $I_E[\Gamma_s]$.

By operating with maximum input power (i.e. with maximum voltage and current), for each velocity, ω, there is a single transmission ratio, n, that would allow the maximum torque, denoted as $\tau_{max}(\omega)$. This torque can be derived from Equations (5.29) and (5.31) by replacing I with I_{nom} and V with V_{max}:

$$
\begin{aligned}
\tau_{\max}(\omega) = \max_{n\in\mathcal{Y}} \; & n k_t I_{nom} - \left(b_g + n^2 b_m\right)\omega, \\
\text{subject to}: \quad & r I_{nom} + n k_v \omega = V_{max},
\end{aligned}
\tag{5.35}
$$

where $\mathcal{Y} \subset \mathbb{R}$ is the range of possible reduction ratios for this problem. Since a gearbox in the above AROP consists of a finite number of gears, it cannot operate at $\tau_{\max}$

[1]Recall that ψ denotes a deterministic objective value.

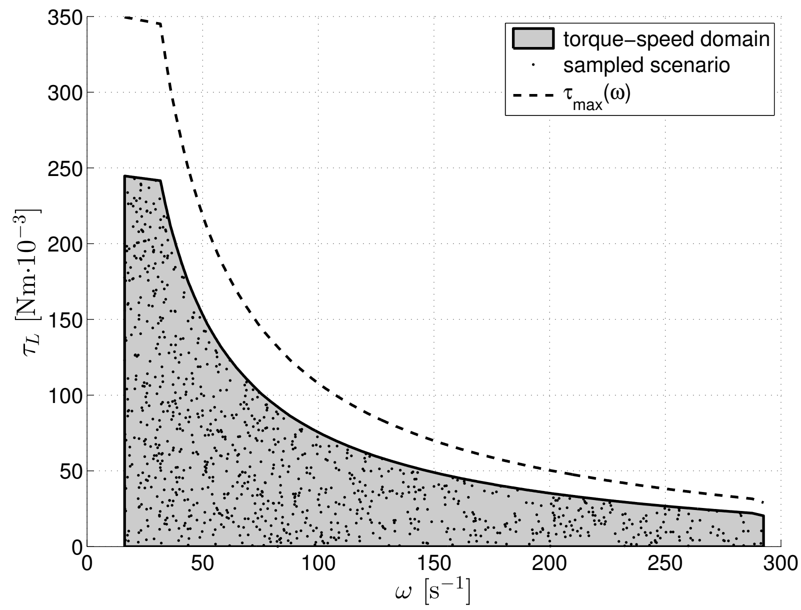


Figure 5.7: The possible domain of torque-speed scenarios, and a representative set randomly sampled with an even probability.

for most of the velocities. In order to obtain feasible solutions with five gears or less, the domain of possible scenarios in this example is assumed to be in the range of $0 \leq \tau(\omega) \leq 0.55\tau_{\max}(\omega)$. The effects of this assumption on the obtained solutions' robustness are further discussed in Section 5.3.5.

Some information on the probability of load scenarios is usually known in a typical gearbox design (e.g. drive cycle information in vehicle design). In this generic example this kind of information is not available, and therefore a uniform distribution is assumed. The other uncertainties are treated in a similar manner: A uniform distribution is assumed for R and B_m, since the tolerance information provided by the manufacturer only specifies the boundaries for the actual property values, but does not specify their distribution. The epistemic uncertainty regarding b_g also results in a uniform distribution of B_g within an estimated interval.

Monte-Carlo sampling is used to represent the uncertain parameter domain $\mathbf{P}$. A set, $\overline{\mathbf{P}}$, of size, k, is constructed by a random sampling of $\mathbf{P}$ with an even probability. In this example, $\overline{\mathbf{P}}$ consists of $k = 1,000$ scenarios. The choice of sample size is further investigated in Section 5.3.5. Figure 5.7 depicts the domain of load scenarios Ω and $\mathcal{T}$, together with their samples in $\overline{\mathbf{P}}$ and the curve $\tau_{max}(\omega)$.

The parameter values and the limits of search variables and uncertainties are presented in Table 5.2. The values and tolerances for the motor parameters were taken

Table 5.2: Variables and parameters for the gearbox AROP.

Type	Variable/ Parameter	Symbol	Units	Lower limit	Upper limit
$\mathbf{x}$	no. of gears	n_g		2	5
	no. of teeth	z_g		19	61
$\mathbf{y}$	gear no.	i		1	n_g
	input voltage	V	V	0	12
$\mathbf{p}$	load speed	ω	s^{-1}	16	295
	load torque	τ	$\mathrm{Nm}\cdot 10^{-3}$	0	$0.55 \cdot \tau_{\max}(\omega)$
	armature resistance	r	Ω	2.1	2.4
	motor damping coefficient	b_m	$\mathrm{Nm}\cdot\mathrm{s}\cdot 10^{-6}$	2.8	3.5
	gear damping coefficient	b_g	$\mathrm{Nm}\cdot\mathrm{s}\cdot 10^{-6}$	25	35
	velocity constant	k_v	$\mathrm{V}\cdot\mathrm{s}\cdot 10^{-3}$	24.3	
	torque constant	k_t	$\mathrm{Nm}\cdot\mathrm{A}^{-1} \cdot 10^{-3}$	24.3	
	max nominal current	I_{nom}	A	1.8	
	first reduction ratio	n_1		61/19	
	transmission no. of teeth	z_t		80	
	cost coefficient	α	\$	5	
	cost coefficient	β		0.8	
	cost coefficient	λ	\$	0.01	
	cost coefficient	δ	\$	50	

from the online catalog of Maxon (2014). Note that the upper limit of the selected gear, i, is n_g, meaning that different gearboxes possess different domains of adjustable variables. This notion is manifested in the problem definition as $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$.

5.3.4 Simulation Results

The discrete search space consists of 1,099,252 different combinations of gears (2–5 gears, 43 possibilities for the number of teeth in each gear: $C_2^{43} + C_3^{43} + C_4^{43} + C_5^{43}$). The constraints and objective functions depend on the number of teeth, z; as a result, they only have to be evaluated 43 times for each of the 1000 sampled scenarios. As a result, it is feasible to find the true Pareto-optimal solutions to the above problem by evaluating all of the solutions. The entire simulation takes less than one minute, using standard desktop computing equipment.

A feasible solution is a gearbox that has at least one gear that does not violate the constraints for each of the scenarios (i.e., $I \leq I_{nom}$ and $V \leq V_{max}$). Figure 5.8 depicts the objective space of the AROP. There are 194,861 feasible solutions (marked with green dots), and the 103 non-dominated solutions are marked with blue dots. It is noticed that the solutions are grouped into three clusters with a different price range

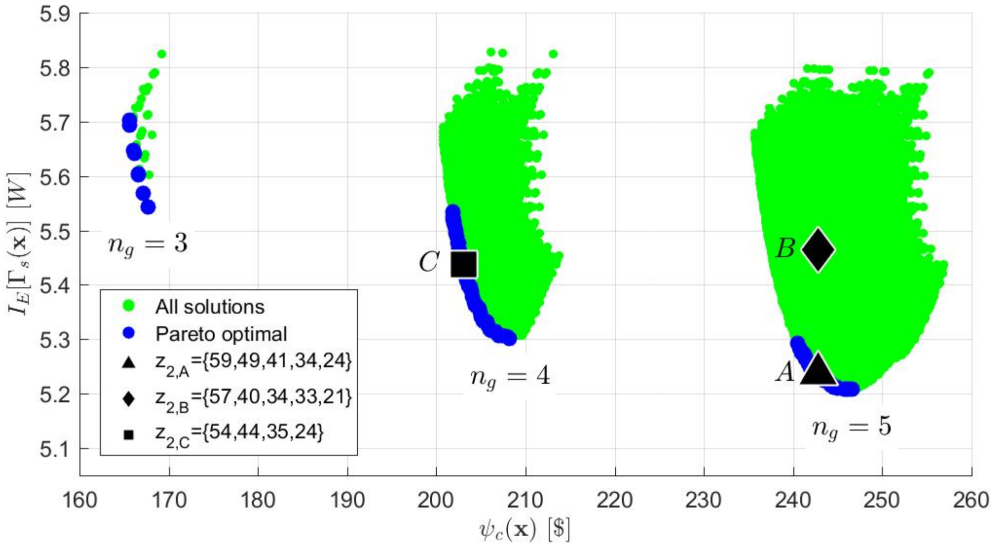


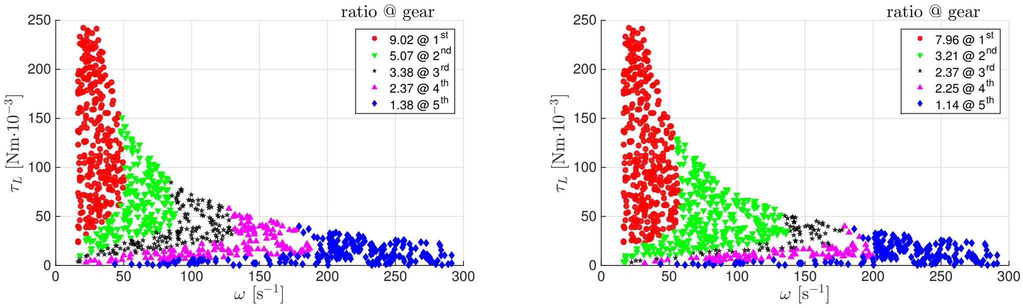
Figure 5.8: The objective values of all feasible solutions to the problem in Equation (5.34) and the Pareto front.

for each number of gears. The three clusters correspond to $n_g \in \{3, 4, 5\}$, where fewer gears are related with a lower cost. None of the solutions with $n_g = 2$ is feasible.

A Comparison Between an Optimal Solution and a Non-Optimal Solution

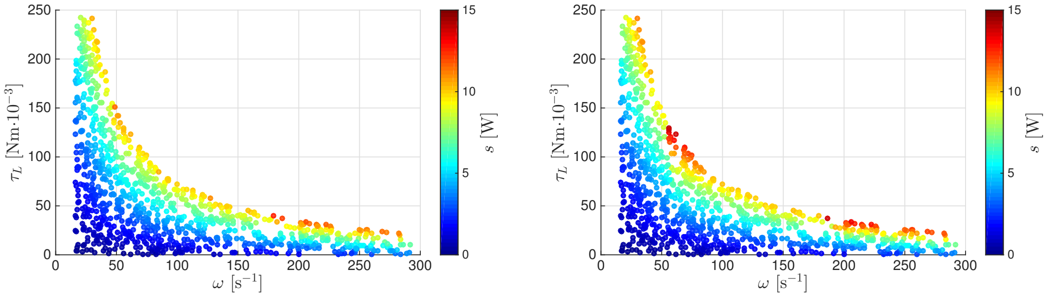
For a better understanding of the results obtained by the AR approach, two candidate solutions are examined: one that belongs to the Pareto-optimal front, and another that does not. Consider a scenario where lowest energy consumption is desired for a given budget limitation. For the sake of this example, a budget limit of \$243 per unit is arbitrarily chosen. The gearbox with the best performance for that cost is marked in Figure 5.8 as Solution A. This solution consists of five gears with $\mathbf{z_2}_{,A} = \{59, 49, 41, 34, 24\}$ and corresponding transmission ratios are $\mathbf{n}_A = \{9.02, 5.07, 3.38, 2.37, 1.38\}$. Another solution with the same cost is marked in Figure 5.8 as Solution B. The gears of this solution are $\mathbf{z_2}_{,B} = \{57, 40, 34, 33, 21\}$, and its corresponding transmission ratios are $\mathbf{n}_B = \{7.96, 3.21, 2.37, 2.25, 1.14\}$.

Figure 5.9 depicts the set of optimal transmission ratios at every sampled scenario for both solutions. Each transmission is marked in the figure with a different marker. This set is, in fact, the set, $\overline{\mathbf{Y}^\star}$, from Equation (5.34), that corresponds to the sampled set of load scenarios $\bar{\mathbf{P}}$, in Figure 5.7. It is observed that the reduction ratios of Solution A almost form a geometrical series, where each consecutive ratio is divided by 1.6 approximately. The resulting $\overline{\mathbf{Y}^\star}(\mathbf{x}_A)$ is such that all gears are optimal for a similar number of load scenarios. Solution B on the other hand has two gears with very similar ratios. It can be seen in Figure 5.9(b) that the third and the fourth gears are barely



(a) Solution A (b) Solution B

Figure 5.9: Optimal transmission ratio for every sampled scenario.



(a) Solution A (b) Solution B

Figure 5.10: Lowest power consumption for every sampled scenario.

used. These gears do not contribute much to the gearbox's efficiency, but significantly increase its cost. As can be seen in Figure 5.8, there are gearboxes with four gears that achieve the same or better efficiency as Solution B.

Figure 5.10 depicts the lowest power consumption for every sampled scenario, $\bar{\Gamma}_s\big(\mathbf{x}, \overline{\mathbf{Y}^\star}, \bar{\mathbf{P}}\big)$. This consumption is achieved by using the optimal gear for each load scenario (those in Figure 5.9). It can be seen that Solution A uses less energy at many load scenarios compared with Solution B. This is depicted by the warmer shades of many of the scenarios in Figure 5.10(b). The expected power consumption of both solutions are $I_E\big[\bar{\Gamma}_s\big(\mathbf{x}_A, \overline{\mathbf{Y}^\star}, \bar{\mathbf{P}}\big)\big] = 5.23\text{W}$ and $I_E\big[\bar{\Gamma}_s\big(\mathbf{x}_B, \overline{\mathbf{Y}^\star}, \bar{\mathbf{P}}\big)\big] = 5.47\text{W}$. These are calculated by averaging the values of all points in Figure 5.10. Considering that both solutions cost the same, this confirms Solution A's superiority over Solution B. Given a budget limitation of \$243, Solution A should be preferred by the decision-maker.

5.3.5 Robustness of the Obtained Solutions

In this section the sensitivity of the AROP's solution to several factors of the problem formulation is examined. Two aspects are considered with respect to different robustness metrics and parameter settings: i) the optimality of a specific solution, and ii) the difference between two alternative solutions. For this purpose, three tests are performed. The first relates to the robustness of the solutions to epistemic uncertainty, namely the unknown range of load scenarios. The second test relates to the robustness of the solutions to a different robustness metric. The third test examines the sensitivity to the sampling size.

Sensitivity to Epistemic Uncertainty

The domain of load scenarios is bounded between $0 \leq \tau \leq 0.55 \cdot \tau_{\max}(\omega)$. The choice of 55% is arbitrary, and it reflects an assumption made to quantify an epistemic uncertainty about the load. Similarly, the upper bound for $\mathcal{T}$ could be a function $a \cdot \tau_{\max}(\omega)$ with a different value of a. The Pareto frontiers for several values of a can be seen in Figure 5.11. For $a = 40\%$, the Pareto set consists of solutions with two, three, four and five gears, whereas for $a = 70\%$ the only feasible solutions are those with five gears. For percentiles larger than 70% there are no feasible solutions within the search domain.

To examine the effect of the choice of maximum torque percentile on the problem's solution, the three solutions from Figure 5.8 are plotted for every percentile in Figure 5.11. Solutions A and C, which belong to the Pareto set for $a = 55\%$, are also Pareto-optimal for all other values of a smaller than 65%. Solution B remains dominated by both Solutions A and C. When very high performance is required (i.e. maximum torque percentiles of 65% or higher), both Solution A and Solution C become infeasible.

It can be concluded that the mean value, as a robustness metric, is not sensitive to the maximum torque percentile. On the other hand, the reliability of the solutions, i.e. their probability to remain feasible, is sensitive to the presence of extreme loading scenarios.

Sensitivity to Preferences

The target-based robustness indicator, Iq, (see Section 2.4) is used to examine the sensitivity of the obtained solutions to preferences. The aim is to check whether the distinction between an optimal and a non-optimal design is maintained when different targets are required for the power consumption.

Figure 5.12 depicts the results of the AROP described in Section 5.3.3, when the consumption target is set to $q = 11\text{W}$. The same three solutions from Figure 5.8 are also shown here. Solution A, whose mean power consumption is the best for its price, is not optimal when the probability for especially poor performance is considered. Solution A manages to satisfy the goal for 98.6% of the sampled scenarios, while another solution

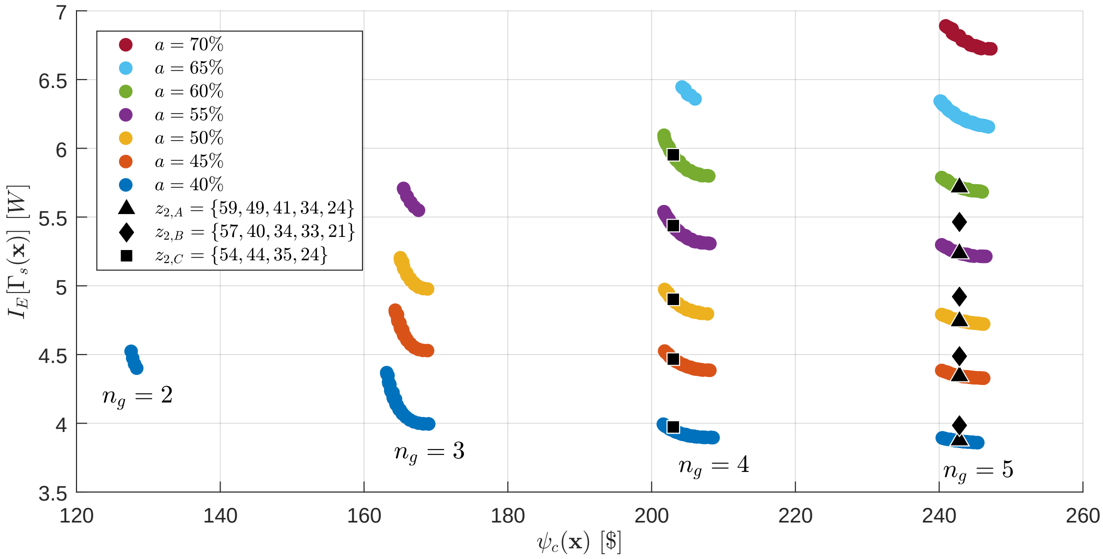


Figure 5.11: Pareto frontiers for different upper bounds of the uncertain load domain $a \cdot \tau_{\max}(\omega)$.

with the same price satisfies 99% of the scenarios. It is up to the decision-maker to determine whether the difference between 98.6% and 99% is significant or not.

Solutions B and C are consistent with the other robustness metric. Solution B is far from optimal, and Solution C is still Pareto-optimal. This consistency is maintained for different values of the threshold, q, as can be seen in Figure 5.13. Figure 5.13 also demonstrates that setting an over-ambitious target results in a smaller probability of fulfilment by any solution.

Sensitivity to the Sampled Representation of Uncertainties

The random variates are represented in this study with a sampled set, using Monte-Carlo methods. The following experiment was conducted in order to verify that 1,000 samples are enough to provide a reliable evaluation of the solutions' statistics: Solutions A and C were evaluated for their mean power consumption over $5,000$ different sampled sets with sizes varying from $k = 100$ to $k = 100,000$. Figure 5.14(a) depicts the metric values of the solutions for every sample size. It is evident from the results that a large number of samples is required for the sampling error to converge. For both solutions, the standard deviation is 15%, 6%, 2% and 0.5% of the mean value, for sample sizes of $k = 100$, $k = 1,000$, $k = 10,000$, and $k = 100,000$, respectively. If an accurate estimate is required for the actual expected power consumption, a large sample size must be used (i.e. larger than $k = 1,000$, which was used in this study).

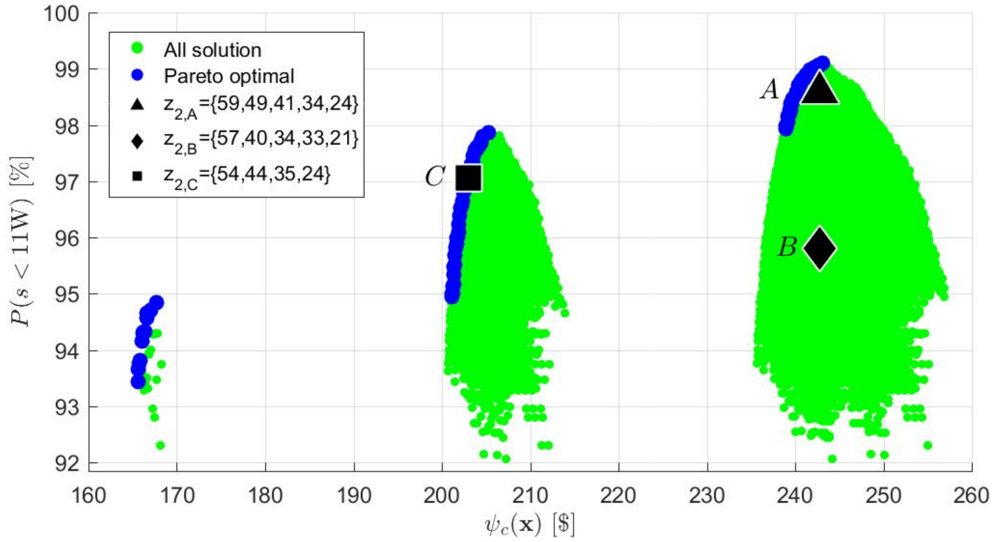


Figure 5.12: The objectives values of all feasible solutions and Pareto front, for maximizing the threshold probability.

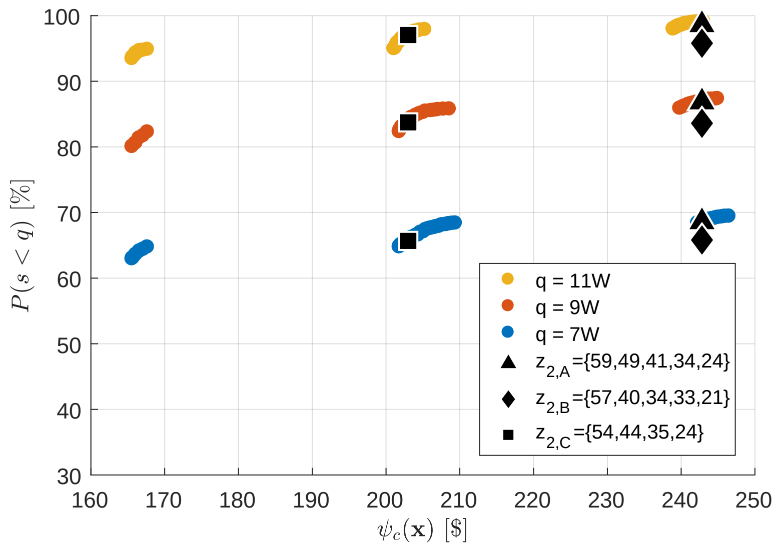
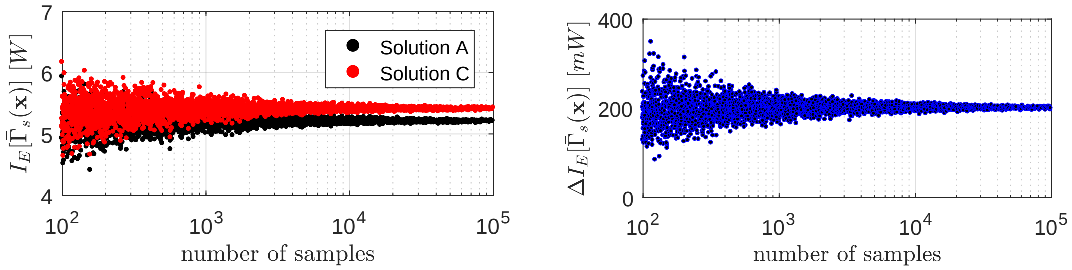


Figure 5.13: Pareto frontiers for different thresholds, q.



(a) Mean power consumption of Solution A and Solution C.

(b) Difference between the mean power consumption of the two solutions.

Figure 5.14: Convergence of the mean power consumption of two solutions for different numbers of samples.

On the other hand, a comparison between two candidate solutions can be based on a much smaller sampled set. Although the values of $I_E\left[\bar{\Gamma}_s\left(\mathbf{x}, \overline{\mathbf{Y}}^{\star}, \overline{\mathbf{P}}\right)\right]$ may change considerably between two consequent realisations of $\overline{\mathbf{P}}$, a similar change will occur for all candidate solutions. This can be seen in Figure 5.14(a) where the "funnels" of the two solutions seem like exact replicas with a constant bias. The difference in performance between the two solutions $\Delta I_E\left(\overline{\mathbf{P}}\right)$ is defined:

$$\Delta I_E\left(\overline{\mathbf{P}}\right) = I_E\left[\bar{\Gamma}_s\left(\mathbf{x}_C, \overline{\mathbf{Y}}^{\star}, \overline{\mathbf{P}}\right)\right] - I_E\left[\bar{\Gamma}_s\left(\mathbf{x}_A, \overline{\mathbf{Y}}^{\star}, \overline{\mathbf{P}}\right)\right] \tag{5.36}$$

Figure 5.14(b) depicts the value of $\Delta I_E\left(\overline{\mathbf{P}}\right)$ for every evaluated sampled set. It can be seen that $\Delta I_E\left(\overline{\mathbf{P}}\right)$ converges to 200mW. For a sample size of $k = 100$, the standard deviation of $\Delta I_E\left(\overline{\mathbf{P}}\right)$ is 25mW, which is only 12% of the actual difference. This means that it can be argued with confidence that Solution A has better performance than Solution C, based on a sample size of $k = 100$.

Based on the results from this experiment, it can be concluded that the solution to the AROP (i.e. the set of Pareto-optimal solutions) is not sensitive to the sample size. The Pareto front, shown in Figure 5.8, might be shifted along the $I_E\left[\bar{\Gamma}_s\right]$ axes for different sampled representations of the uncertainties, but the same (or very similar) solutions would always be identified.

5.3.6 Discussion

This study is the first of its kind to extend gearbox design optimization to consider the realities of uncertain load demand. It demonstrates how the stochastic nature of the uncertain load demand can be fully catered for during the optimization process using an Active Robustness approach. A set of optimal solutions with a trade-off between cost and efficiency was identified, and the advantages of a gearbox from this set over a non-optimal one were shown.

The robustness of the obtained Pareto-optimal solutions to several aspects of the problem formulation was verified. It was found that the solution is not sensitive to the assumptions regarding the uncertain load domain. The solutions' results were also obtained when a different robustness criterion was used. The influence of the sample size for representing the uncertain parameters was also examined. Although the indicator's value requires a very large sample to converge (more than 10,000), a sample as small as 100 has been found to be sufficient for a comparison between two candidate solutions.

Computational complexity is a concern for the ARO approach demonstrated in this study. This case study used very simple analytic functions to evaluate each candidate solution. Therefore, the real solution to the AROP could be found almost instantly. When applying this method to real-world applications, every function evaluation might require extensive computational effort. In this case, efficient optimization algorithms would be required, and the uncertainties may need to be described by methods other than Monte-Carlo sampling. However, the large amount of function evaluations required to solve a typical AROP is a feasible prospect for real industrial problems. Since the problem is solved off-line, before the product goes to manufacturing, supercomputing facilities are likely to be available, and a reasonable time-scale for solving the problem might be days or even a few weeks.

In this study the gearbox's adaptability was evaluated by only considering its performance at each of the sampled load scenarios, i.e., at steady-state. However, the Active Robustness methodology considers adaptability in a wider sense. In addition to its performance at steady-state, the solution's transient behaviour during adaptation to environmental changes should also be considered. For the problem presented in this section, an environmental change is a change in demand from one load scenario to another. Although the optimal configurations can be found for both scenarios, the gearing ratios and input voltages applied while changing between these configurations may have a substantial impact on the solution's performance. This notion was deliberately not considered in the current study in order to focus on basic aspects of the approach.

The next section extends the AROP to also consider the transitions between optimal configurations. Two additional objectives for the transient state are discussed: adaptation time and energy consumption.

5.4 Gearbox Design - Multi-Objective Formulation

This section extends the AROP presented in the previous section to evaluate the candidate designs for their transient performance. This extension transforms the problem from a single-objective AROP to an active robust *multi-objective* optimization problem. In addition to finding the optimal configuration for every load scenario at steady-state, the trajectory of adjustable variables when accelerating the load from rest is also considered. The acceleration trajectory is evaluated by two additional objectives: the energy required to accelerate to steady-state speed, denoted γ_e, and the time it takes to accelerate from rest, denoted γ_t. The power consumption of the gearbox at steady-state, γ_s, still needs to be minimized. The cost objective is discarded from the problem formulation in order to focus on the new problem features that distinguish it from the AROP in Section 5.3.

Since cost is removed from the problem formulation, there is no need to consider gearboxes with fewer than five gears. Therefore, the search is for the optimal set of five gears, while the adjustable variables include the motor input voltage and the selected gear, similarly to the AROP in Section 5.3. The uncertain parameters include the required speed and torque, as well as the inertia of the load, which is required for transient analysis.

All objectives depend on the uncertain parameters, $\mathbf{p}$, and the configuration, $\mathbf{y}$, which makes the problem an ARMOP. The transient objectives are calculated by numerically solving an ordinary differential equation. The time required to evaluate the transient objectives does not enable the optimal solution to be found by using enumeration, as in Section 5.3, and requires an efficient optimization algorithm.

An evolutionary algorithm is used to solve the problem. It is constructed according to the unary indicator approach for solving ARMOPs that was presented in Section 4.5. It demonstrates the applicability of the ARO methodology to handle optimization problems with high complexity and more expensive evaluation functions.

5.4.1 Mathematical Model

The motor and gear system includes the same Maxon A-max 32 electric motor that was used in Section 5.3. All variables and parameters of the motor and gearbox, including inertias which are denoted with the letter j, are described in Table 5.3.

The transient version of the steady-state Equations (5.30)–(5.32) include inertia and acceleration:

Table 5.3: Variables and parameters for the gearbox ARMOP.

Type	**Variable/ Parameter**	**Symbol**	**Units**	**Lower limit**	**Upper limit**
$\mathbf{x}$	no. of teeth	z_g		19	61
$\mathbf{y}$	gear no.	i		1	5
	input voltage	V	V	0	12
$\mathbf{p}$	load speed	ω	s^{-1}	16.5	295
	load torque	τ	$\mathrm{Nm}\cdot 10^{-3}$	10	260
	load inertia	j_L	$\mathrm{Kg}\cdot\mathrm{m}^2 \cdot 10^{-3}$	5	10
	velocity constant	k_v	$\mathrm{V}\cdot\mathrm{s}\cdot 10^{-3}$	24.3	
	torque constant	k_t	$\mathrm{Nm}\cdot\mathrm{A}^{-1} \cdot 10^{-3}$	24.3	
	armature resistance	r	Ω	2.23	
	motor damping coefficient	b_m	$\mathrm{Nm}\cdot\mathrm{s}\cdot 10^{-6}$	3.16	
	motor inertia	j_m	$\mathrm{Kg}\cdot\mathrm{m}^2 \cdot 10^{-6}$	4.17	
	max nominal current	I_{nom}	A	1.8	
	gear damping coefficient	b_g	$\mathrm{Nm}\cdot\mathrm{s}\cdot 10^{-6}$	30	
	first reduction ratio	n_1		3.21	
	transmission no. of teeth	z_t		80	
	maximum acceleration time	$t_{\max}$	s	20	
derived	armature current	I	A	0	5.39
	second reduction ratio	n_2		0.311	3.21
	total reduction ratio	n		1	10.3
	layshaft inertia	j_l	$\mathrm{Kg}\cdot\mathrm{m}^2 \cdot 10^{-6}$	15.9	64.5
	load shaft inertia	j_g	$\mathrm{Kg}\cdot\mathrm{m}^2 \cdot 10^{-6}$	5.21	53.7

$$s(t) = V(t) * I(t)\,, \tag{5.37}$$

where:

$$I(t) = \frac{\left(j_L + j_g + n_2(t)^2\, j_l + n(t)^2\, j_m\right)\dot{\omega}(t) + \left(b_g + n(t)^2\, b_m\right)\omega(t) + \tau}{n(t)\, k_t}, \tag{5.38}$$

$$V(t) = rI(t) + n(t)\, k_v \omega(t)\,. \tag{5.39}$$

When the load is accelerated from rest, it is possible to calculate the speed trajectory, for given trajectories of input voltage and speed reduction, by solving the following differential equation:

$$\dot{\omega}(t) = \frac{n(t)\, k_t V(t) - n(t)^2\, k_v k_t \omega(t)}{\left(j_L + j_g + n_2(t)^2\, j_l + n(t)^2\, j_m\right) r} - \frac{\left(b_g + n(t)^2\, b_m\right)\omega(t) + \tau}{j_L + j_g + n_2(t)^2\, j_l + n(t)^2\, j_m}, \tag{5.40}$$

where $\omega(0) = 0$ is used as a starting condition.

Once the speed trajectory, $\omega(t)$, is known, the total energy required for acceleration, γ_e, can be derived from Equations (5.37) and (5.39):

$$u = \int_0^{t_f} \frac{V(t)\left(V(t) - n(t)\, k_v \omega(t)\right)}{r} \mathrm{d}t, \tag{5.41}$$

where t_f is the time at which ω reaches the required speed.

5.4.2 Problem Formulation

According to the ARO methodology, the problem variables are sorted in Table 5.3 into three types: $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{p}$. The only source of uncertainty considered in this problem is the uncertain load demand. Its three characteristics, ω, τ and j_L, are treated as random variates, denoted Ω, $\mathcal{T}$ and J_L, respectively.

A gearbox is required to perform well both in steady-state and during acceleration. These two requirements can be considered as different operation modes, with different configuration spaces. The configuration space in steady-state includes the choice of the gear, i, and the input voltage, V. During acceleration, it consists of trajectories in time of $i(t)$ and $V(t)$. Therefore, the search for the optimal configuration can be separated to $\mathbf{y}^\star_{ss}$ that minimises steady-state power consumption, γ_s, and to $\underline{\mathbf{y}^\star_{tr}}$ that minimises acceleration energy, γ_u, and time, γ_t. Since the latter is a solution to a MOP, it is expected to be a set. The variates of $\mathbf{y}^\star_{ss}$ and $\underline{\mathbf{y}^\star_{tr}}$ that correspond to the variate, $\mathbf{P}$, are denoted as $\mathbf{Y}^{\star}_{ss}$ and $\underline{\mathbf{Y}^{\star}_{tr}}$, respectively.

Following the above, the AROP is formulated:

$$\min_{\mathbf{x}\in\mathcal{X}} \left[\Gamma_s(\mathbf{x},\mathbf{Y}^{\star}_{ss},\mathbf{P})\,,\underline{\Gamma_u}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)\,,\underline{\Gamma_t}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)\right], \tag{5.42}$$

where :

$$\mathbf{Y}^{\star}_{ss} = \underset{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}{\operatorname{argmin}}\,\Gamma_s(\mathbf{x},\mathbf{y},\mathbf{P})\,, \tag{5.43}$$

$$\underline{\mathbf{Y}^{\star}_{tr}} = \underset{\mathbf{y}\in\mathcal{Y}(\mathbf{x})}{\operatorname{argmin}}\,[\Gamma_u(\mathbf{x},\mathbf{y},\mathbf{P})\,,\Gamma_t(\mathbf{x},\mathbf{y},\mathbf{P})]\,, \tag{5.44}$$

$$\mathbf{x} = [z_i]\,,\quad i = 1,\dots,5, \tag{5.45}$$

$$\mathbf{y} = [i, V]\,, \tag{5.46}$$

$$\mathbf{P} = [\Omega,\mathcal{T},J_L,k_v,k_t,r,b_m,I_{\text{nom}},b_g,n_1,z_t,j_m,j_l,j_G,t_{\max}]\,, \tag{5.47}$$

s.t. :

$$z_{g,i} + z_{l,i} = z_t,\quad i = 1,\dots,5, \tag{5.48}$$

$$I_{ss} \le I_{\text{nom}}, \tag{5.49}$$

$$t_f \le t_{\max}. \tag{5.50}$$

The steady-state current constraint is evaluated according to Eq. (5.38), and the objectives according to Equations (5.37), (5.40) and (5.41).

Since the ARMOP consists of separable configuration spaces, it can be decoupled into two subproblems, one that searches for $\mathbf{Y}^{\star}_{ss}$ and $\Gamma_s(\mathbf{x},\mathbf{Y}^{\star}_{ss},\mathbf{P})$, and another that searches for $\underline{\mathbf{Y}^{\star}_{tr}}$ and $\underline{\mathbf{\Gamma}_{tr}}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)$, where

$$\underline{\mathbf{\Gamma}_{tr}}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big) = \left[\underline{\Gamma_u}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)\,,\underline{\Gamma_t}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)\right]. \tag{5.51}$$

The former problem is a single-objective AROP, and the latter is an ARMOP. Using robustness indicators, Eq. (5.42) can be converted to the following bi-objective problem that simultaneously minimises the steady-state AROP and the transient ARMOP:

$$\min_{\mathbf{x}\in\mathcal{X}} \left[I\Big[\Gamma_s(\mathbf{x},\mathbf{Y}^{\star}_{ss},\mathbf{P})\Big], I_{\text{hv}}\Big[\underline{\mathbf{\Gamma}_{tr}}\Big(\mathbf{x},\underline{\mathbf{Y}^{\star}_{tr}},\mathbf{P}\Big)\Big]\right]. \tag{5.52}$$

The HV measure, q_{hv}, needs to be maximized. In order to pose the multi-objective problem (5.52) as a minimization problem, the value of $1 - Q_{hv}$ is used for calculating the robustness indicator I_{hv}.

5.4.3 Optimiser Design

The problem was solved by a bi-level EMOA whose structure is described in Algorithm 5.6.

First, the uncertain domain is sampled n_p times. These samples serve as the same representation of uncertainties to evaluate all solutions.

Next, Eq. (5.43) is solved for the entire design space, and $\mathbf{Y}^\star_{ss}$ and $\Gamma_s(\mathbf{x}, \mathbf{Y}^\star_{ss}, \mathbf{P})$ are stored in an archive for every feasible solution. It is possible to find the optimal steady-state configuration of every solution for all sampled load scenarios because the design space is discrete and the objective and constraints are simple expressions. The search space consists of 962,598 different combinations of gears (choice of 5 gears from 43 possibilities). The constraints and objective functions depend on the number of teeth, z, and therefore they only have to be evaluated 43 times for each of the sampled scenarios. A feasible solution is a gearbox that has at least one gear that does not violate the constraints for each of the scenarios (i.e., $I \leq I_{\text{nom}}$ and $V \leq V_{\max}$).

Next, a multi-objective search is conducted amongst the feasible solutions to solve Eq. (5.52). The solutions to Eq. (5.44) for every sampled scenario are obtained by the evolutionary algorithm described in Section 5.4.3. The solutions to Eq. (5.43) are already stored in an archive.

Algorithm 5.6 Pseudo-algorithm for solving the ARMOP

```
sample the uncertain domain
evaluate all possible solutions for steady-state (s.s)
initialise nadir and ideal points for transient objectives (limits)
generate an initial population
while stopping criterion not satisfied do
   for every scenario do
      for every new solution do
         optimise for time–energy and store PF
      end for
   end for
   if limits have changed then
      update limits
      calculate Q_hv of entire population
   else
      calculate Q_hv of new feasible solutions
   end if
   assign scalar indicator values for s.s and transient
   evolve new population (selection, cross-over and mutation)
   re-mutate solutions that were already evaluated / infeasible for s.s
end while
```

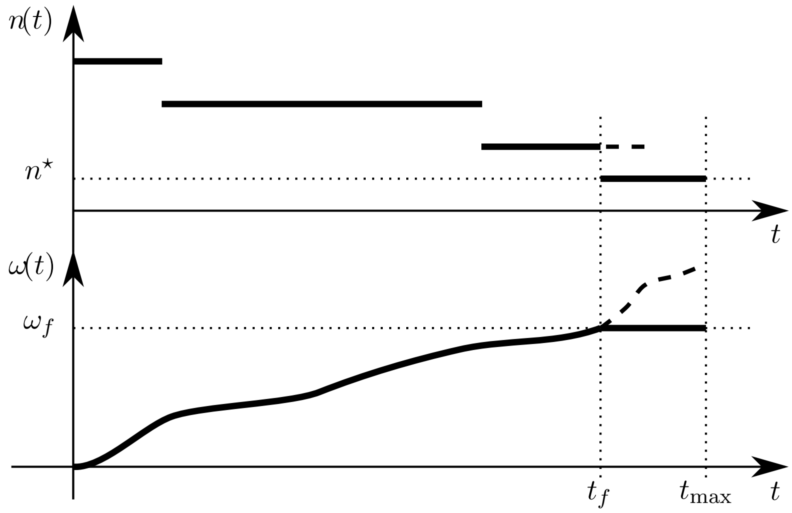


Figure 5.15: Gearing trajectory of a candidate solution and the resulting speed trajectory.

EMOA for Identifying Optimal Gearing Sequences

For every load scenario, a multi-objective optimization is conducted for each candidate solution to identify the optimal shift sequence that minimises energy and acceleration time. Early experiments revealed that maximum voltage results in better values for both objectives, regardless of the candidate solution or the load scenario. Therefore, the input voltage was considered as constant $V_{\max}$, and the only search variable is $i(t)$, the selected gear at time, t. A certain trajectory, $i(t)$, results in a gearing ratio trajectory, $n(t)$, that depends on the gearbox, $\mathbf{x}$, that is being evaluated.

The trajectory, $i(t)$, is coded as a vector of time intervals $\mathbf{dt} = [\mathrm{d}t_1, \ldots, \mathrm{d}t_{i^\star}]$ defining the duration of each gear in the sequence from first gear to the $i^{\star\text{th}}$, with $i^\star$ being the optimal gear at steady-state for the load scenario under consideration. The sum of all time intervals is equal to $t_{\max}$, and this relation is enforced whenever a new solution is created by setting:

$$\mathbf{dt} \leftarrow \frac{\mathbf{dt}}{\|\mathbf{dt}\|_1} t_{\max}. \tag{5.53}$$

Inserting $n(t)$ into Eq. (5.40) results in a trajectory, $\omega(t)$, which can be used to calculate Γ_u, Γ_t, or whether the gearbox has failed to reach the desired speed before $t_{\max}$. Figure 5.15 illustrates how the gearing sequence is coded for an example candidate solution, and the resulting speed trajectory. The final gear, which is optimal for the steady-state, is the fourth gear. Therefore the gearing sequence includes four time intervals, one for each gear from first to fourth. It can be seen that the required velocity has been reached before the maximum allowed time for acceleration. This time is marked as t_f. When the required velocity is reached, the system is still operated in third gear. The rest of the gearing trajectory and the resulting speed trajectory are marked with dashed lines. These lines mark the solution's genotype and the result of the differential equation. In practice, once the desired speed is reached, the optimal gear (fourth) is engaged and the input voltage is lowered from $V_{\max}$ to the optimal value.

A multi-objective evolutionary algorithm was used to estimate the set of optimal trajectories

$$\underline{\mathbf{y}_{tr}^{\star}} = \underset{n(t)}{\operatorname{argmin}} \left[\gamma_u(\mathbf{x}, n(t), \mathbf{p}), \gamma_t(\mathbf{x}, n(t), \mathbf{p})\right], \tag{5.54}$$

where both $\mathbf{x}$ and $\mathbf{p}$ are fixed during the entire optimization run. Solving the differential equation (5.40) repeatedly throughout the optimization run in order to obtain $\underline{\mathbf{y}_{tr}^{\star}}$ is the most expensive part of the algorithm in terms of computational resources. Therefore, all of the solutions to (5.54) are stored in an archive to avoid repeated computations.

Calculating the Set-based Robustness Indicator

The ARMOP's indicator, I_{hv}, uses a dynamic reference point. At every generation, after the approximated Pareto frontiers, $\underline{\boldsymbol{\Gamma}_{tr}}\left(\mathbf{x}, \underline{\mathbf{Y}_{tr}^{\star}}, \mathbf{P}\right)$, are identified for all evaluated solutions, the ideal and worst objective vectors are re-evaluated to include the objective vectors of the new solutions. If neither the ideal nor the worst objective vectors have changed, I_{hv} is calculated only for the recently evaluated solutions according to the procedure described in Section 4.4.4. Otherwise, the indicator values of the entire current population are recalculated as well, in order to allow for fair comparisons between new and old candidate solutions. No preferences were considered in this case study, hence, the objectives were normalised by setting $\boldsymbol{\gamma}^w$ to 1.

5.4.4 Simulation Results

Parameter Setting

The ARMOP described in Section 5.4.2 was solved with the proposed evolutionary algorithm. Two robustness criteria were considered: I_w considers the worst-case scenario, meaning the upper limits of the uncertain load parameters, as given in Table 5.3. I_E considers the expected value over a set of sampled load scenarios. For both cases the same criterion was used for the steady-state and transient indicators of Eq. 5.52, i.e., either $I_w[\Gamma_s]$ and $I_{hv,w}\left[\underline{\boldsymbol{\Gamma}_{tr}}\right]$ or $I_E[\gamma_s]$ and $I_{hv,E}\left[\underline{\boldsymbol{\Gamma}_{tr}}\right]$.

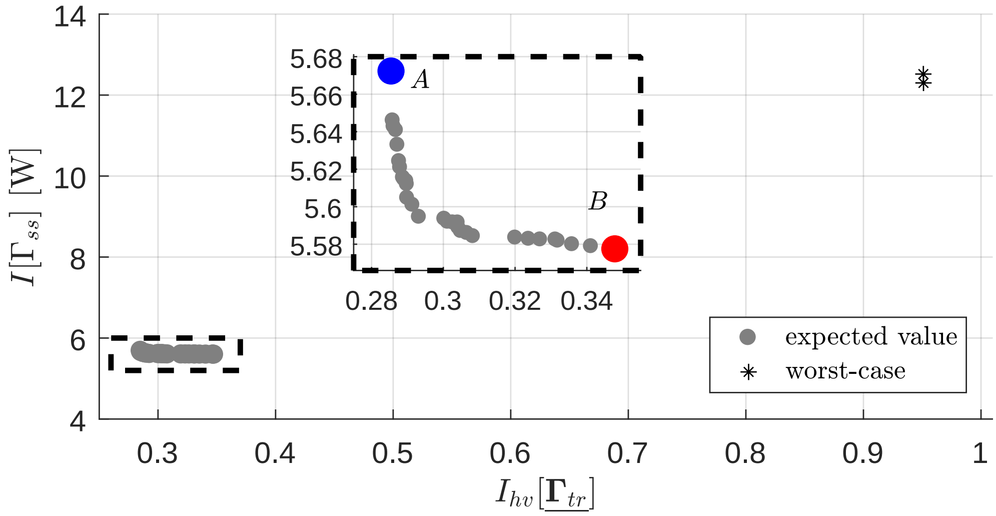


Figure 5.16: Approximated Pareto frontiers for the worst-case and mean-value criteria. A close-up of the robust mean Pareto front is shown with the extreme solutions marked as A and B.

NSGAII-PSA (Salomon et al., 2013b) with a fixed number of generations was used for both stages of the problem (referred to as outer and inner).
Parameter setting of the outer algorithm: population size $N = 100$, 50 generations, integer-coded, One-point crossover with crossover rate $p_c = 1$, polynomial mutation with mutation rate, $p_m = 1/n_x = 0.2$, and distribution index, $\eta_m = 20$.
Parameter setting of the inner algorithm: population size $N = 50$, 30 generations, real-coded, SBX crossover with crossover rate, $p_c = 1$, and distribution index, $\eta_c = 15$, polynomial mutation with mutation rate, $p_m = 1/n_y = 0.2$, and distribution index, $\eta_m = 20$.
Both stages used sequential tournament selection, considering constraint violation, non-dominance rank and niche count, and had an elite population size of $N_E = 0.4N$. The uncertain load domain was sampled 25 times using Latin hypercube sampling.

Results

The approximated Pareto frontiers for both worst-case and mean-value criteria are depicted in Figure 5.16. For the worst-case criterion, the PS consists of only two, almost identical, solutions. In a close-up view on the approximated PF for expected performance, the extreme solutions are marked as A and B.

Details on the solutions for both robustness criteria are summarised in Table 5.4. Note the similarity in both design and objective spaces between the two solutions of the worst-case problem, and the difference between Solutions A and B. Also note that

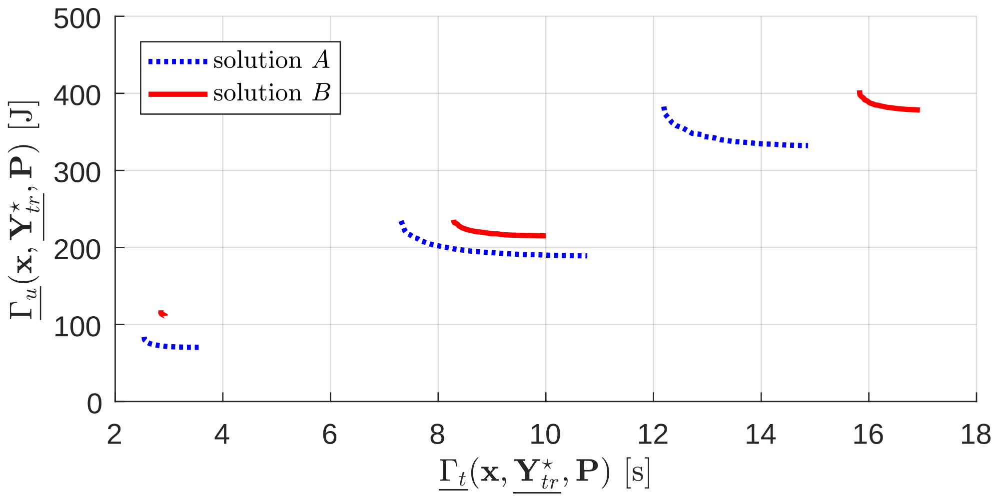


Figure 5.17: Approximated Pareto frontiers of two solutions for three (of the 25) scenarios.

the best solutions found for a certain robustness criterion, are dominated for another. Solution B performs well in most steady-state scenarios, since it has a large variety of high gears (small reduction ratio), but its ability to efficiently accelerate the load is limited for the same reason. Solution B becomes infeasible when the worst-case is considered. This was not detected while optimising for the mean value since the worst-case scenario was not sampled. This result highlights the impact of the choice of robustness criterion, and the challenge in optimising for the worst-case (see Branke and Rosenbusch (2008)).

The dynamic performances of Solutions A and B for three load scenarios are depicted in Fig 5.17. Solution A's superiority for both dynamic objectives is well captured by the I_{hv} indicator values.

Table 5.4: Optimization Results of the Gearbox ARMOP.

Goal	**Solu-tion**	**Reduction Ratios** 1st	2nd	3rd	4th	5th	$I_E[\Gamma_s]$	$I_{hv,E}[\underline{\Gamma_{tr}}]$	$I_w[\Gamma_s]$	$I_{hv,w}[\underline{\Gamma_{tr}}]$
I_E	A	9.02	4.34	2.62	1.93	1.30	5.672	0.2857	13.10	0.9631
	B	2.76	2.25	1.92	1.73	1.64	5.577	0.3481	infeasible	
I_w		7.06	3.38	2.14	1.55	1.14	5.649	0.2896	12.30	0.9511
		7.49	3.38	2.03	1.46	1.14	5.660	0.2899	12.52	0.9510

5.4.5 Discussion

This case study demonstrates how a real-world design optimization problem can be formulated as an active robust multi-objective optimization problem. The approach taken to solve the ARMOP is to use a scalarising function to represent the variate of Pareto frontiers of every candidate solution. This approach was found useful for the gearbox case study – solutions with better Pareto frontiers were assigned with a better indicator value. However, whenever a set is represented by a scalar value, some of its information must be lost. As a result, setting a robustness criterion for the utility indicator value does not automatically imply that the individual objectives will also be robust.

Being a bi-level optimization problem, an AROP requires many function evaluations. An ARMOP is even harder to solve, because the inner problem is a MOP. The strategy for obtaining robust solutions taken in this study was based on Monte Carlo simulations to represent the uncertain variables. This representation requires a large set of samples in order to adequately capture the nature of the uncertainties, and to gain confidence in the robustness of the obtained solutions. Due to limited computational resources, the approach was demonstrated with a small set of sampled scenarios, only to provide a proof of concept. Even for these minimal optimiser settings, almost 70 million function evaluations were conducted. It took approximately three days to compute on a 3.40GHz Intel® Core™ i7-4930K CPU, running Matlab® on 12 cores.

The approach takes account of – and exploits – user influence on system performance, but presently assumes that the user is able to operate the gearbox in an optimal manner to achieve best performance. Of course, this assumption can only be fully validated if a skilled user or a well-tuned controller activates the gearbox. This raises an important issue of how to train this user or controller to achieve best performance, which is identified as a priority for further research.

5.5 Summary

The framework for Active Robust Optimization enables designers to conduct optimization as part of the design process of changeable products. In order to demonstrate how this can be done in practice, two case study applications from the field of engineering design were presented in this chapter. The first case study considered a changeable optical table for protection of sensitive experimental equipment from floor vibrations. The second case study considered the optimization of a gearbox for an uncertain load demand. Simplified mathematical models for the performance of both products were derived from first principles. The models were constructed according to the ARO convention, which makes a distinction between uncontrolled parameters, fixed decision variables and adjustable decision variables.

Both single-objective and multi-objective AROPs were presented and solved. The optical table problem in Section 5.2 consists of two objectives, vibration damping and cost, both affected by the selected configuration. Since the damping could be identified as a leading objective for selecting the optimal configuration, the ARMOP was solved by the single-objectivisation approach.

The gearbox optimization case study consisted of two parts. The first, which was presented in Section 5.3, only considered steady-state conditions. The cost objective for this formulation is not affected by the selected configuration, and therefore, the problem was formulated as a single-objective AROP. The second part of the gearbox case study, presented in Section 5.4, considered transient performance as well. Two additional objectives were introduced: energy consumption during acceleration, and acceleration time. The configuration space of the steady-state and transient objectives is not the same, since the transient case considers the trajectory of the configuration in time. Therefore, the ARMOP was separated into two sub-problems, one for steady-state performance and another for transient performance. To evaluate the transient performance, the scalarisation approach, described in Section 4.4, was used.

The algorithms presented in Section 4.5 were used to solve the problems of the case studies. A set of robust solutions could be found for every problem. The robustness of the obtained solutions were analysed for a variety of aspects of the problem formulation. Additionally, the changeable solutions to the AROPs were compared with non-changeable solutions to conventional RO problems. The superiority of the changeable solutions could always be identified, and the added benefit of optimizing changeable solutions could be clearly shown. With the new ARO framework, many other real-world changeable products can now be optimized as well.

Chapter 6

Conclusions

The Active Robust Optimization framework has been established in this thesis in order to allow for the optimization of changeable products, i.e., products that can be adjusted by their users for improved performance in changing or uncertain environments. Changeable products are commonly designed and manufactured in many fields of industry. However, until now, there were no available tools to properly optimize such products during their design phase. Active Robust Optimization creates, for the first time, the opportunity to incorporate meaningful optimization into the design process of changeable products. In contrast to the few existing attempts to optimize changeable products, this framework considers the true effects of uncertainty factors and the ability of the product to respond.

The study combines concepts and methods from the fields of operational research, multi-criteria decision-making, engineering design and probability theory. The methodology includes an analysis of the uncertainties involved and a probabilistic view of the product's performance. Robust optimization is used to define the kind of robustness that is desired from the product, given the uncertainty over its performance. The ability of the product to be adjusted by its user is emphasised through a dynamic optimization scheme, where the optimal configuration is sought over different scenarios of the uncertain environment.

The Active Robust Optimization Problem, formulated in Chapter 3, forms the core of the framework. It is a nested optimization problem: the outer problem searches for the basic features of the design that distinguish it from alternative changeable products, and the inner problem searches for the optimal configuration for different scenarios of the uncertainties. The base assumption is that some aspects of the problem that are uncertain during the design phase, become certain when the product is put into service. Therefore, the search for the optimal configuration is undertaken for deterministic values of the previously uncertain parameters. For a single-objective AROP, a single optimal configuration exists for every scenario of the uncertainties.[1] This results in a one-to-one mapping between every uncertain scenario and the optimal performance. Therefore, the outer problem can be viewed as a conventional robust optimization problem. The effects of different robustness criteria on the solution of an AROP were analysed and discussed. Other aspects, such as different sources of uncertainties and sampling of the uncertainty factors, have also been addressed.

When the evaluation of a changeable product is based on more than a single objective, decision-making plays an important role in the optimization process. In Chapter 4, the Active Robust Multi-Objective Optimization Problem was introduced. It seems very similar to the single-objective AROP, where instead of a single objective, several objectives need to be simultaneously optimized. However, the ARMOP introduces considerable complexity that does not exist in any other optimization problem in the literature. Instead of a single optimal configuration for any realisation of the uncertainties, a candidate solution can have a set of Pareto-optimal configurations, where the user can choose any of them during product operation. As a result, the performance of a candidate solution is described by a variate of sets of objective vectors. In Chapter 4, strategies for evaluating, comparing and searching for candidate solutions of ARMOPs were presented and demonstrated. High-level descriptions of several evolutionary algorithms that use different evaluation approaches were presented as well.

The methodology was demonstrated with simple analytical functions. These functions were used to construct both single-objective and multi-objective AROPs. Additionally, two case studies involving the optimization of real-world changeable products were presented. The case studies provided proof-of-concept results for the applicability of the framework. The advantage of changeability for achieving robustness in a cost-effective manner was demonstrated for the case studies by comparing changeable products with their non-changeable counterparts. The changeable products were optimized using Active Robust Optimization, while the non-changeable products were optimized using conventional Robust Optimization.

This dissertation includes the following major contributions:

1. Framework for Active Robust Optimization.

2. Framework for Active Robust Multi-Objective Optimization.

3. Metrics for evaluation and comparison of ARMOP solutions.

4. Two case study applications.

[1] This is the case, unless the function is multi-modal, in which case there might be multiple configurations with the same optimal performance.

To conclude the thesis, this chapter provides the following remarks. First, the contributions and the key results are summarised in Section 6.1. Then, the conditions required for using the framework are discussed in Section 6.2 together with the framework's limitations. Finally, suggestions for additional research and development are made in Section 6.3.

6.1 Key Results

The findings of this study can be summarised according to the following contributions.

6.1.1 Framework for Active Robust Optimization

1. The success of the ARO methodology in optimizing changeable products is rooted in the way in which the performance of such products is modelled. The mathematical model of the objective function makes a clear distinction between uncontrolled parameters, fixed decision variables and adjustable decision variables. This distinction enables the designer to better understand the effect of adaptability on a product's performance. After optimizing the fixed decision variables, the designer can decide which adjustable features to include, and to what extent.

2. The AROP is a nested optimization problem. The main (outer) problem searches for the robust set of fixed decision variables that defines the changeable product. The inner problem searches for the optimal configuration of the adjustable variables for given realisations of the uncertainties. After optimizing the configuration for every uncertain scenario under consideration, a one-to-one mapping exists between the uncertain scenarios and the optimal configurations. As a result, the outer problem becomes identical to conventional RO problems, i.e., the objective is a random variate, and robustness criteria are used to evaluate it. Similarly to other RO problems, the definition for robustness plays an important role in identifying the most robust solution to an AROP.

3. An AROP must consist of at least one objective, one fixed decision variable, one adjustable decision variable and one source of uncertainty. The source of uncertainty can be a random environmental parameter, uncertainty over the actual value of the selected decision variables or over the prediction of the mathematical model of the objective function. An analytic example, with the dimensionality mentioned above, was crafted to highlight the special features of AROPs. A constraint between the fixed and adjustable decision variables was used for making the adaptability dependent on the design. The relative simplicity of the problem

made it possible to analytically propagate the uncertainties from the parameters to the objective. It also enabled the effects of different features in the problem to be isolated, as described in the points below.

4. A comparison between an adaptive solution and its non-adaptive counterpart could confirm that adaptability improves the robustness for a variety of robustness metrics and sources of uncertainty. Adaptability was found to be most effective in improving the performance for extreme cases of the uncertainty factors. Thus, the robustness of the solution is enhanced without sacrificing peak performance at nominal conditions.

5. Monte-Carlo sampling is often used in robust optimization to approximate the distribution of an uncertain objective function according to a sampled set. The *law of large numbers* states that an infinitely large sample of independent experiments will have the same distribution as the analytical one. The number of samples that is required to converge towards the analytical value for various robustness indicators was examined. It was found for the example analytical function that the convergence rate greatly varies with the metric used to evaluate robustness. Therefore, it is recommended to conduct some experiments before beginning the optimization in order to find the smallest sample size that provides a reliable approximation of the true indicator value.

6. The AROP was formulated in this study with uncertain environmental parameters (Type A uncertainty). An examination of similar AROPs with other sources of uncertainty was also conducted in order to identify the similarities and differences from the basic AROP. It was found that AROPs with uncertainty over the realised value of decision variables (Type B) and over the mapping between decision and objective space (Type C) can be treated in a similar fashion to AROPs with Type A uncertainties.

6.1.2 Framework for Active Robust Multi-Objective Optimization

1. The generalisation of the AROP to consider multiple conflicting objectives introduces a great challenge. Instead of a one-to-one mapping between every uncertain scenario and the optimal configuration, there might be a one-to-many mapping, as the solution to the inner problem may be a set of Pareto-optimal configurations. As a result, the performance of any candidate solution is a variate of sets of objective vectors. Quantifying and comparing candidate solutions according to this kind of representation has never been done before.

2. To demonstrate and analyse the unique structure of ARMOPs, the analytic example of a single-objective AROP was extended to a bi-objective problem. The ARMOP was constructed from a distance term and a direction term. The distance term includes a fixed decision variable, an uncertain parameter and an additional adjustable decision variable. The direction term includes an additional uncertain parameter and adjustable variable. While the distance term is shared by the two objectives, the direction term determines the trade-off between them. By introducing merely a single additional parameter and a single adjustable variable that affect the ratio between the objectives, all the unique features of ARMOPs could be revealed and discussed. This was achieved while keeping the dimensionality of the problem to a minimum.

3. The use of robustness indicators to solve an ARMOP is more complicated than in single-objective AROPs. The indicators suggested in this study followed these three steps: i) represent each Pareto front with a single scalar/vector, ii) consider the variate of this scalar/vector given the entire uncertainty range, and iii) use a robustness criterion to quantify the above variate.

4. Four different approaches for comparing ARMOP solutions were suggested in the thesis. Each approach is associated with a metric for the first step of the evaluation, i.e., with the way in which a set of Pareto-optimal configurations is represented by a single scalar/vector. The different approaches can be useful for a decision-maker to incorporate high-level knowledge into the ARMOP, in order to steer the search towards a solution with the most desirable trade-off between the objectives. The following approaches were suggested: *single-objectivisation, decomposition by scalarisation, set-based unary quantification* and *set-based binary comparison.* It was found that each approach has its advantages and disadvantages. Suggestions for choosing the most suitable approach for solving a given problem were provided according to the strengths and weaknesses of each approach in terms of complexity, ease of decision-making and resemblance to the original ARMOP objectives.

5. High-level descriptions of evolutionary algorithms were provided to demonstrate how each approach can be used to solve an ARMOP. The algorithms share a basic structure, with some customisations made for every approach. All algorithms consist of a main EA for robust optimization, based on a sampled representation of the uncertainties, and a nested algorithm for finding the optimal set of configurations for every sample. The high complexity, that is inherent in algorithms for

solving ARMOPs, was evident in the presented algorithms. In order to evaluate a single candidate solution, a MOP needs to be solved for every sampled scenario of the uncertainties. Then a quality indicator needs to be computed for every set of Pareto-optimal configurations. Many candidate solutions need to be evaluated according to the above procedure during the execution of the main EA for robust optimization. Some approaches, such as single-objectivisation and decomposition, transform the inner problem into a SOP, thereby substantially reducing the algorithm complexity.

6.1.3 Case Study Applications

Optical table

1. This case study examined whether adaptability can improve the robustness of an isolation system for optics experiments. In order to do so, a changeable optical table to isolate the sensitive equipment from floor vibrations was suggested. The isolation system is installed in the table's legs, and includes springs and dampers. In contrast to a conventional optical table, the position of the legs can be adjusted according to the experimental settings.

   The changeable optical table is designed to be robust to uncertainty with regard to the frequency of the floor vibration and the setting of the experimental equipment. While the uncertainty about the disturbance frequency does not decrease much when the table is put into service, all the required information on the weight distribution of the experimental equipment is known at the beginning of each experiment. This information enables the user to adjust the locations of the legs and the stiffness of the damper, to minimise the vibration transmitted from the floor to the equipment.

2. Cost needs to be considered in almost any design activity. A cost function for changeable products generally consists of three elements: manufacturing costs, operational costs and adaptation costs. These three elements were demonstrated for the adaptive optical table through a generic cost function. While the changing environment does not affect manufacturing costs, operational and adaptation costs are functions of the configuration and changes between configurations when the product is being used.

3. Since both the damping and the cost can be affected by adaptation, the problem is formulated as an ARMOP. However, the cost of adaptation and the operational costs could be considered to be secondary to the main objective of the product,

which is preventing floor vibrations from disturbing the experiment. Therefore, single-objectivisation was chosen as the most appropriate approach to solve the ARMOP. The solutions to the problem present different trade-offs between cost and vibration damping, but once a product is selected for a given cost, it would be operated at the most suitable configuration for every experimental setting.

4. Using methods from dynamics and vibration theory, a model of the adaptive optical table was derived from first principles. The amplitude ratio between the vibration of the floor and the equipment could be calculated with a closed form expression. This enabled many function evaluations to be conducted and made it possible to use Monte-Carlo sampling to approximate the distribution of the uncertain objectives. An evolutionary algorithm for single-objectivisation was designed and used to solve the ARMOP.

5. The suggested EA for single-objectivisation was found to be successful in finding a set of solutions with trade-offs between cost and vibration damping. Three Pareto-optimal solutions were investigated in order to examine the manner in which adaptability is applied to different experimental settings. It was found that both the position of the legs and the stiffness of the damper should be adjusted according to the weight distribution of the experimental equipment, to maximize absorption of vibration energy in the isolation system.

   The results show that isolation systems with softer springs can better absorb the energy than stiff springs. However, soft springs are more expensive. The changeable design that could provide the best isolation had springs with a stiffness ratio of $1:3$. This enables the product to react in an optimal manner to different scenarios.

6. A non-changeable design was also optimized for similar conditions. The difference from the changeable design is that the position of the legs and the stiffness of the damper must be determined during the design stage, and they cannot be changed before every experiment. The solution was found at the boundary of the design space, with the springs located as close to the centre, the damper located at the centre and both springs and damper have the lowest possible stiffness.

   The superiority of the changeable design over the conventional design was demonstrated. The changeable design with best vibration damping performed three times better than the non-changeable design.

Gearbox

1. The second case study considers a gearbox as a changeable product, and optimizes it with the tools of the ARO methodology. The gearbox is required to convert the power provided from an electric motor to a load shaft. The required speed and torque of the load may vary within known limits, and therefore they are treated as uncertain entities during the optimization. The changeability of the gearbox lies in the ability to decide which cogwheel to use for transmitting the power from the motor to the load. The optimization task is to decide how many gears to include in the gearbox, as well as their gearing ratios. The adaptability of each candidate design includes the available gears to select from, and therefore, each candidate solution has a different domain of changeable variables.

2. The case study begins with a single-objective AROP formulation. It consists of two objectives, but only one of them is affected by the uncertainties and adaptation. As a result, the problem is not an ARMOP. The two objectives were power consumption at steady-state and the cost of the gearbox.

3. A model for the system was derived from first principles. It was based on the physical phenomena in the electric motor, and the kinematic constraints in the gearbox. The objective functions could be calculated for every combination of parameters and decision variables by solving a simple equation. Furthermore, every load scenario had to be evaluated for only 43 possible gearing ratios. The combinatoric nature of the design space resulted in just over one million possible designs. The reasons mentioned above enabled a large number of samples to be obtained over the uncertain load domain and the problem was solved by evaluating all candidate designs. As a result, an analysis of the ARO framework for the application could be made without being biased by algorithmic issues.

4. Using the expected value as a robustness criterion, the solution to the AROP was found to be a set of designs, offering different trade-offs between expected power consumption and cost. The differences between an optimal design and another design with the same cost, but higher power consumption, could be identified.

5. Sensitivity of the obtained solutions was examined for the following aspects: epistemic uncertainty, preferences and the sampled representation of the uncertain parameters.

   Epistemic uncertainty is a result of missing information on the problem at the time of its formulation. To assess the effects of epistemic uncertainty on the obtained solution, the problem was solved over different domains for the uncertain load requirement. It was found that the chosen robustness criterion (the expected value) is not sensitive to this type of uncertainty. However, the reliability of solutions was found to be sensitive to the presence of extreme loading scenarios.

The choice of robustness criterion and its parameters can be considered as part of the problem formulation, and they may affect the solution of the AROP. To examine this effect, the AROP was solved with the target-based indicator using different targets for the random power consumption. It was found that similar solutions were obtained for the new robustness metric, with different targets. As expected, confidence in achieving the target decreases as the goal becomes more ambitious.

The size of the sample required to converge towards the true indicator value was also examined. The results have shown that a very large sample is required for the sampling error to decrease (approximately $10,000$ samples for a standard deviation of 2% of the mean value). However, it was also found that there is a very strong correlation between differences in performance of alternative solutions according to the sampled set. Therefore, a set as small as 100 samples was found to be enough to compare between two candidate solutions.

The results obtained for the above three experiments are very encouraging. They show that a reliable solution to an AROP can be obtained with a relatively small number of samples, and it can remain robust to the uncertainties when different robustness criteria are considered.

6. In the second part of the case study, two objectives were added to the AROP in order to optimize the transient behaviour of the gearbox. The transient objectives search for the optimal gearing sequence when accelerating the load from rest to each desired speed. They include the power consumed during acceleration and the time it takes to reach the steady-state velocity. A conflict between the two objectives may occur, resulting in a set of Pareto-optimal gearing sequences. Having a set of configurations for every scenario of the uncertain load requirements, transformed the gearbox optimization problem into an ARMOP.

7. The acceleration trajectory for every gearing sequence was simulated by numerically solving an ordinary differential equation. The duration of each simulation ruled out the possibility of solving the inner problem (i.e., finding the optimal gearing sequences) using enumeration, as conducted for the steady-state objective. Therefore, an optimization algorithm, based on the unary indicator approach, was

used to solve the ARMOP. The ability to find a solution to the ARMOP using the suggested EA has demonstrated how the ARO methodology can be applied to problems with more expensive evaluation functions.

8. The unary hypervolume indicator was used to scalarise each Pareto frontier of transient configurations. The outer problem of the AROP was then composed of transient and steady-state performance. The solution to this problem was again a set of Pareto-optimal solutions, providing a choice between steady-state and transient performance. The difference between the two extreme solutions could be analysed in order to examine the effectiveness of the suggested hypervolume indicator.

6.2 Limitations

The usefulness of the Active Robust Optimization methodology to optimize adaptive products was demonstrated throughout this thesis. The generic nature of the approach makes it suitable for optimizing many real-world applications. However, some limitations of the framework need to be considered in order to decide whether or not it can be used for a given application:

1. Similarly to any other computational method for optimization, the AROP requires models of the objective function(s) that can predict the product's performance. These models must be able to simulate the effects of changes in fixed decision variables, adjustable variables and parameters.

2. In order to conduct meaningful robust optimization, the uncertainty factors need to be described in a reliable fashion. Similar to other robust optimization/design activities, the less accurate the understanding of the uncertainties, the less robust the solution will be to these uncertainties.

3. The uncertainty quantification method that was demonstrated in this thesis was based on Monte-Carlo sampling. This method requires a large number of samples in order to reliably capture the true nature of the uncertain parameters. In the case of expensive evaluation functions,[1] Monte Carlo sampling cannot be used and needs to be replaced with a more efficient UQ method.

4. The AROP is a nested robust optimization problem, and therefore it requires a large amount of function evaluations in order to be solved. Applications that

[1]An expensive evaluation function may require a long time to compute and/or large amount of other resources.

include expensive evaluation functions may take too long to optimize with this approach. Please refer to the next section for some suggestions for overcoming this issue.

6.3 Future Work

The concepts and methods presented in this thesis are the foundations of the Active Robust Optimization framework. The framework can be extended in many directions in order to be established as a leading tool for optimizing changeable products. The suggested future studies that are listed below will help to enrich the framework and popularise it within the research community and among industry.

1. The first action would be to apply the methods already developed on a real industrial problem. In order to do this, a collaboration with an industrial partner needs to be established. This collaboration will be beneficial for the industrial partner that will be able to produce products with improved robustness. It will also promote the framework by benchmarking it on a real-world application. Furthermore, a collaboration between industry and academia is likely to identify the issues that require further attention in order to achieve a higher technical readiness level.

2. The ARO methodology assumes that the changeable product can be adjusted to the optimal configuration for every scenario of the uncertain parameters. The study presented in this thesis did not consider the control that needs to be applied in order to achieve the right adaptation for every scenario. An investigation of changeable products from a Control Theory point of view can raise issues such as controllability that may influence the feasibility or optimality of candidate solutions. Eventually, a solution to an AROP should also include the controller that will ensure correct adaptation by the product, as was assumed during the optimization process.

3. Optimization of the adaptation process itself was addressed in Section 5.4 as part of the gearbox optimization case study. This issue has been previously addressed by Salomon et al. (2013a) in the form of an *Optimization of Adaptation Problem*. The problem was proposed as an alternative to Optimal Control. It concerns designing an adaptation trajectory when a solution to a dynamic optimization problem needs to be adapted in order to track a changing optimum. The solution to an OAP is a set of trajectories that offer a trade-off between the function value during adaptation and the adaptation cost. An interesting future work will be

to include the objectives of the OAP as part of the AROP. This will provide a better understanding of the changeable product's adaptability and lead to a better design.

4. The AROP is formulated in a manner that assumes the changeable product can be adapted to the optimal configuration for every realisation of the uncertain parameters. In practice, when the product is put into service, it might be operated at sub-optimal configurations. The reasons may vary from a wrong controller design, inexperienced operator, wrong sensing of the changing environmental conditions, or a decision not to adapt due to considerations that were not taken into account during the formulation of the AROP. In any of these cases, the actual performance of the changeable product would be inferior to its predicted performance, which led to identifying it as the most robust solution.

   Sub-optimality can be taken into account by introducing uncertainty over the actual value of the identified optimal configuration. This means that the optimal configuration is assumed to be a random variate. The consequences of this assumption on the AROP structure, complexity and algorithmic design should be investigated.

5. Decision-making plays an important role in formulating and solving active robust multi-objective optimization problems. In this thesis, aspects of decision-making were used to choose the suitable algorithmic approach for solving a given ARMOP. Some of the suggested approaches can exploit *a priori* knowledge, while others allow for *a posteriori* decision-making.

   Another aspect of decision-making that exists for ARMOPs is also worth exploring. During product operation, once the environmental conditions have changed and a new set of trade-off configurations is available, a single configuration needs to be selected. It would be useful to define a set of rules that can guide decision-makers in choosing a new configuration. This selection, for example, might be according to similarity in performance to the previous configuration, as conducted by Avigad and Eisenstadt (2010), or by keeping the same ratio between the objectives. In case of automatic adaptation, these rules can serve as a basis for designing a controller.

6. The greatest challenge in developing the ARO methodology will be to overcome the inherently high complexity of the AROP. Focus should be applied on several fronts:

(a) Use state-of-the-art optimization algorithms, and develop efficient algorithms specifically designed to solve AROPs. This includes the leverage of existing methods for dynamic optimization to find the optimal configurations in the inner problem of the AROP.

(b) Explore efficient uncertainty quantification methods to approximate the random performance distribution. Reducing the amount of samples drawn from the uncertain parameters will greatly reduce the overall effort required to solve an AROP.

(c) In the case of expensive evaluation functions, a reliable use of surrogate models should be investigated. Using a surrogate model that is fast to compute, but introduces additional uncertainty, can greatly speed up the optimization process. It is important to investigate how surrogate models can be used without decreasing the reliability of the obtained solution due to the added uncertainty. A combination of high- and low-fidelity models can be applied, where the expensive high-fidelity model is mainly used for validation.

7. In order to encourage designers to use the methods presented in this study, they should be incorporated in an optimization environment (software). The best platform to start with would be an open-source framework such as jMetal (Durillo and Nebro, 2011; Nebro et al., 2015) or Liger (Giagkiozis et al., 2013a). Both frameworks can be extended with user-defined algorithms. While the former is more widespread, the latter offers more flexibility for crafting algorithms with unique architectures. After achieving a satisfactory level of maturity, the methods could be also incorporated into a commercial optimization software platform.

Bibliography

Amram, M. and Kulatilaka, N. *Real Options: Managing Strategic Investment in an Uncertain World*. Harvard Business School Press, Boston, Massachusetts, 1999.

Avigad, G. and Branke, J. Embedded Evolutionary Multi-objective Optimization for Worst Case Robustness. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 617–624, New York, NY, USA, 2008. ACM.

Avigad, G. and Eisenstadt, E. Robustness of Multi-objective Optimal Solutions to Physical Deterioration through Active Control. In Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S., Jain, A., Aggarwal, V., Branke, J., Louis, S., and Tan, K., editors, *Simulated Evolution and Learning SE - 43*, volume 6457 of *Lecture Notes in Computer Science*, pages 394–403. Springer Berlin Heidelberg, 2010.

Avigad, G. and Moshaiov, A. Interactive Evolutionary Multiobjective Search and Optimization of Set-Based Concepts. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(4):1013–1027, aug 2009a.

Avigad, G. and Moshaiov, A. Set-Based Concept Selection in Multi-Objective Problems: Optimality Versus Variability Approach. *Journal of Engineering Design*, 20(3): 217–242, may 2009b.

Avigad, G., Eisenstadt, E., and Goldvard, A. Pareto layer: Its Formulation and Search by Way of Evolutionary Multi-Objective Optimization. *Engineering Optimization*, 42 (5):453–470, may 2010.

Avigad, G., Eisenstadt, E., and Schuetze, O. Handling Changes of Performance Requirements in Multi-Objective Problems. *Journal of Engineering Design*, 23(8):597–617, nov 2011.

Avriel, M. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

Bader, J. and Zitzler, E. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.

Barrico, C. and Antunes, C. H. Robustness analysis in multi-obejective optimization Using a degree of robustness concept. *IEEE Congress on Evolutionary Computation*, pages 1887–1892, 2006.

Basseur, M. and Zitzler, E. Handling Uncertainty in Indicator-Based Multiobjective Optimization. *International Journal of Computational Intelligence Research*, 2(3): 255–272, 2006.

Bellman, R. *Dynamic Programming*. Princeton University Press, NJ, USA, 1st edition, 1957.

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. Adjustable Robust Solutions of Uncertain Linear Programs. *Mathematical Programming*, 99(2):351–376, mar 2004.

Ben-Tal, A., Ghaoui, L. E., and Nemirovski, A. *Robust Optimization.* Princeton University Press, 2009. ISBN 9781400831050.

Bertsimas, D. and Caramanis, C. Finite Adaptability in Multistage Linear Optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, dec 2010.

Bertsimas, D., Brown, D. B., and Caramanis, C. Theory and Applications of Robust Optimization. *SIAM Review*, 53(3):464–501, 2011a.

Bertsimas, D., Goyal, V., and Sun, X. A. A Geometric Characterization of the Power of Finite Adaptability in Multistage Stochastic and Adaptive Optimization. *Mathematics of Operations Research*, 36(1):24–54, feb 2011b.

Beume, N., Fonseca, C. M., Lopez-Ibanez, M., Paquete, L., and Vahrenhold, J. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, 2009.

Beyer, H. G. and Sendhoff, B. Robust Optimization - A Comprehensive Survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, jul 2007.

Birge, J. R. and Louveaux, F. *Introduction to Stochastic Programming.* Springer-Verlag, Berlin, 1997.

Blackwell, T. and Branke, J. Multi-swarm Optimization in Dynamic Environments. In Raidl, G., Cagnoni, S., Branke, J., Corne, D., Drechsler, R., Jin, Y., Johnson, C., Machado, P., Marchiori, E., Rothlauf, F., Smith, G., and Squillero, G., editors, *Applications of Evolutionary Computing SE - 50*, volume 3005 of *Lecture Notes in Computer Science*, pages 489–500. Springer Berlin Heidelberg, 2004.

Bradstreet, L. *The Hypervolume Indicator for Multi-objective Optimisation: Calculation and Use.* Phd dissertation, The University of Western Australia, 2011.

Branke, J., Asafuddoula, M., Bhattacharjee, K. S., and Ray, T. Efficient Use of Partially Converged Simulations in Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 21(1):52–64, 2017.

Branke, J. *Evolutionary Optimization in Dynamic Environments.* Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0792376315.

Branke, J. and Rosenbusch, J. New Approaches to Coevolutionary Worst-Case Optimization. In Rudolph, G., Jansen, T., Lucas, S., Poloni, C., and Beume, N., editors, *Parallel Problem Solving from Nature PPSN X SE - 15*, volume 5199 of *Lecture Notes in Computer Science*, pages 144–153. Springer Berlin Heidelberg, 2008.

Buche, D., Stoll, P., Dornberger, R., Koumoutsakos, P., and Büche, D. Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32 (4):460–473, 2002.

Buurman, J., Zhang, S., and Babovic, V. Reducing Risk Through Real Options in Systems Design: The Case of Architecting a Maritime Domain Protection System. *Risk Analysis*, 29(3):366–379, mar 2009.

Chalupnik, M. J., Wynn, D. C., and Clarkson, P. J. Comparison of Ilities for Protection Against Uncertainty in System Design. *Journal of Engineering Design*, 24(12):814–829, dec 2013.

Coello, C. A. C. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, 1(3):269–308, 1999.

Coello, C. A. C. and Cortes, N. C. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2): 163–190, jun 2005.

Cruz, C., González, J. R., and Pelta, D. a. Optimization in Dynamic Environments: A Survey on Problems, Methods and Measures. *Soft Computing*, 15(7):1427–1448, dec 2011.

Czyzzak, P. and Jaszkiewicz, A. Pareto Simulated Annealing–A Metaheuristic Technique for Multiple-Objective Combinatorial Optimization. *Journal of Multi-Criteria Decision Analysis*, 7(1):34–47, jan 1998.

Dantzig, G. B. and Thapa, M. N. *Linear Programming: 1: Introduction*, volume 1. Springer, 1997. ISBN 0387948333.

Das, S. and Suganthan, P. N. Differential Evolution: A Survey of the State-of-the-Art. *Evolutionary Computation, IEEE Transactions on*, 15(1):4–31, 2011.

de Neufville, R., Scholtes, S., and Wang, T. Real Options by Spreadsheet: Parking Garage Case Example. *Journal of Infrastructure Systems*, 12(2):107–111, jun 2006.

de Neufville, R. Real Options: Dealing With Uncertainty in Systems Planning and Design. *Integrated Assessment*, 4(1):26–34, mar 2003.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6 (2):182–197, 2002.

Deb, K. Unveiling Innovative Design Principles by Means of Multiple Conflicting Objectives. *Engineering Optimization*, 35(5):445–470, oct 2003.

Deb, K. and Gupta, H. Introducing Robustness in Multi-Objective Optimization. *Evolutionary Computation*, 14(4):463–494, nov 2006.

Deb, K. and Jain, S. Multi-Speed Gearbox Design Using Multi-Objective Evolutionary Algorithms. *Journal of Mechanical Design*, 125(3):609–619, sep 2003.

Deb, K., Pratap, A., and Moitra, S. Mechanical Component Design for Multiple Ojectives Using Elitist Non-dominated Sorting GA. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN VI SE - 84*, volume 1917 of *Lecture Notes in Computer Science*, pages 859–868. Springer Berlin Heidelberg, 2000.

Denhart, J. D. *Tradespace Exploration of Reconfigurable Systems with a Mars Rover Case Study*. M.sc thesis, North Carolina State University, 2013.

Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., and Stummer, C. Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. *Annals of Operations Research*, 131(1-4):79–99, 2004.

Dorigo, M., Maniezzo, V., and Colorni, A. Ant System: Optimization by a Colony of Cooperating Agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41, 1996.

Dorigo, M. and Blum, C. Ant Colony Optimization Theory: A Survey. *Theoretical Computer Science*, 344(23):243–278, nov 2005.

Droste, S., Jansen, T., and Wegener, I. On the Analysis of the (1+1) Evolutionary Algorithm. *Theoretical Computer Science*, 276(1-2):51–81, apr 2002.

Du, W. and Li, B. Multi-Strategy Ensemble Particle Swarm Optimization for Dynamic Optimization. *Information Sciences*, 178(15):3096–3109, aug 2008.

Durillo, J. J. and Nebro, A. J. jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760–771, 2011.

Eldred, M. S., Swiler, L. P., and Tang, G. Mixed Aleatory-Epistemic Uncertainty Quantification with Stochastic Expansions and Optimization-Based Interval Estimation. *Reliability Engineering and System Safety*, 96(9):1092–1113, 2011.

Emmerich, M., Beume, N., and Naujoks, B. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In Coello Coello, C., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization SE - 5*, volume 3410 of *Lecture Notes in Computer Science*, pages 62–76. Springer Berlin Heidelberg, 2005.

Evans, J. S. Strategic Flexibility for High Technology Manoeuvres: a Conceptual Framework. *Journal of Management Studies*, 28(1):69–89, jan 1991.

Ferguson, S. and Lewis, K. Effective Development of Reconfigurable Systems Using Linear State-Feedback Control. *AIAA Journal*, 44(4):868–878, apr 2006.

Fieldsend, J. E. and Everson, R. M. Multi-Objective Optimisation in the Presence of Uncertainty. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 243–250, 2005.

Fieldsend, J. E. and Everson, R. M. The Rolling Tide Evolutionary Algorithm: A Multi-Objective Optimiser for Noisy Optimisation Problems. *Evolutionary Computation, IEEE Transactions on*, PP(99):1, 2014.

Fisher, R. R. A. *Statistical Methods for Research Workers*. Genesis Publishing Pvt Ltd, 1925. ISBN 8130701332.

Fletcher, D., Brennan, R. W., and Gu, P. A Method for Quantifying Adaptability in Engineering Design. *Concurrent Engineering*, 17(4):279–289, jan 2010.

Fliege, J. and Werner, R. Robust Multiobjective Optimization & Applications in Portfolio Optimization. *European Journal of Operational Research*, 234(2):422–433, 2014.

Fonseca, C. M. and Fleming, P. J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *International Conference on Genetic Algorithms*, number July, pages 416–423, 1993.

Fonseca, C. M. and Fleming, P. J. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, mar 1995.

Fonseca, C. and Fleming, P. On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In Voigt, H.-M., Ebeling, W., Rechenberg, I., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN IV SE - 59*, volume 1141 of *Lecture Notes in Computer Science*, pages 584–593. Springer Berlin Heidelberg, 1996.

Ford, D. and Sobek, D. Adapting Real Options to New Product Development by Modeling the Second Toyota Paradox. *IEEE Transactions on Engineering Management*, 52(2):175–185, may 2005.

Gadalla, M. and Xue, D. Recent Advances in Research on Reconfigurable Machine Tools: a Literature Review. *International Journal of Production Research*, 55(5): 1440–1454, mar 2017.

Garcia, M. N. Space Debris and Human Spacecraft, 2013. URL `https://www.nasa.gov/mission{_}pages/station/news/orbital{_}debris.html`.

Gaspar-Cunha, A., Ferreira, J., and Recio, G. Evolutionary Robustness Analysis for Multi-Objective Optimization: Benchmark Problems. *Structural and Multidisciplinary Optimization*, 49(5):771–793, 2013.

Gasper, A., Collard, P., and Gaspar, A. From GAs to Artificial Immune Systems: Improving Adaptation in Time Dependent Optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 3, pages 1859–1866, 1999.

Giagkiozis, I., Purshouse, R. C., and Fleming, P. J. Generalized Decomposition and Cross Entropy Methods for Many-Objective Optimization. *Information Sciences*, 282: 363–387, 2014.

Giagkiozis, I., Lygoe, R. J., and Fleming, P. J. Liger: An Open Source Integrated Optimization Environment. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1089–1096, New York, NY, USA, 2013a. ACM.

Giagkiozis, I., Purshouse, R. C., and Fleming, P. J. An Overview of Population-Based Algorithms for Multi-Objective Optimisation. *International Journal of Systems Science*, pp:1–28, aug 2013b.

Gill, P. E., Murray, W., and Wright, M. H. *Practical Optimization.* Academic Press, London, 1981.

Goh, C. K. and Tan, K. C. An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization. *Evolutionary Computation, IEEE Transactions on*, 11 (3):354–381, jun 2007.

Goh, C., Tan, K., Cheong, C., and Ong, Y. An Investigation on Noise-Induced Features in Robust Evolutionary Multi-Objective Optimization. *Expert Systems with Applications*, 37(8):5960–5980, aug 2010.

Goldberg, D. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, jan 1989. ISBN 0201157675.

Gomory, R. E. Outline of an Algorithm for Integer Solutions to Linear Programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.

Gu, P., Hashemian, M., and Nee, A. Adaptable Design. *CIRP Annals-Manufacturing Technology*, 53(2):539–557, 2004.

Gu, P., Xue, D., and Nee, A. Y. C. Adaptable Design: Concepts, Methods, and Applications. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 223(11):1367–1387, nov 2009.

Gunawan, S. and Azarm, S. Multi-Objective Robust Optimization Using a Sensitivity Region Concept. *Structural and Multidisciplinary Optimization*, 29(1):50–60, 2005.

Guntsch, M., Middendorf, M., and Schmeck, H. An Ant Colony Optimization Approach to Dynamic TSP. In Spector, L., editor, *The Genetic and Evolutionary Computation Conference*, pages 860–867, Massachusetts, 2001.

Guzzella, L. and Amstutz, A. CAE Tools for Quasi-Static Modeling and Optimization of Hybrid Powertrains. *Vehicular Technology, IEEE Transactions on*, 48(6):1762–1769, 1999.

Haldaman, J. and Parkinson, M. B. Reconfigurable Products and Their Means of Reconfiguration. In *ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 219–228. ASME, 2010.

Han, Y., Chen, Y., Gu, P., Zhang, J., Bao, N., and Xue, D. A New Method for Evaluating Product Adaptability Based on Carbon Footprint Calculation. In Dorantes-Gonzalez, D. J., Li, Y., Wu, S., Hu, C., Xu, L., Chen, K., Lu, Z., Yang, F., Xu, Z., Song, Y., Li, Y., Qi, H., Geng, X., Wu, Y., and Lv, H., editors, *16th International Conference on Mechatronics Technology*, pages 290–296. Tianjin Foreign Language Electronic & Audio-Video Publishing House, 2012.

Hashemian, M. *Design for Adaptability.* PhD thesis, University of Saskatchewan, 2005.

Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* The University of Michigan Press, Ann Arbor, MI, 1975. ISBN 0472084607.

Horn, J., Nafpliotis, N., and Goldberg, D. E. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 82–87, 1994.

Hu, W., Azarm, S., and Almansoori, A. New Approximation Assisted Multi-objective collaborative Robust Optimization (new AA-McRO) under interval uncertainty. *Structural and Multidisciplinary Optimization*, 47(1):19–35, 2013.

Hughes, E. J. Evolutionary Multi-objective Ranking with Uncertainty and Noise. In Zitzler, E., Thiele, L., Deb, K., Coello Coello, C., and Corne, D., editors, *Evolutionary Multi-Criterion Optimization SE - 23*, volume 1993 of *Lecture Notes in Computer Science*, pages 329–343. Springer Berlin Heidelberg, 2001.

Hughes, E. J. Multiple Single Objective Pareto Sampling. *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, 4:2678–2684 Vol.4, 2003.

Inoue, K., Townsend, D. P., and Coy, J. J. Optimum Design of a Gearbox for Low Vibration. *International Power Transmission and Gearing Conference*, 2:497–504, 1992.

Jahn, J. *Mathematical Vector Optimization in Partially Ordered Linear Spaces.* Peter Lang, Frankfurt am Main, Germany, 1986.

Jiang, S. and Yang, S. Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons. *IEEE Transactions on Cybernetics*, 47(1):198–211, 2017.

Jin, Y. and Branke, J. Evolutionary Optimization in Uncertain Environments A Survey. *Evolutionary Computation, IEEE Transactions on*, 9(3):303–317, 2005.

Kall, P. and Wallace, S. W. *Stochastic Programming.* John Wiley and Sons Ltd, Chichester, England, 1994.

Kennedy, J. and Eberhart, R. Particle Swarm Optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Perth, WA, Australia, nov 1995. IEEE.

Knowles, J. and Corne, D. On Metrics for Comparing Nondominated Sets. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, pages 711–716. IEEE, 2002.

Knowles, J., Corne, D., and Reynolds, A. Noisy Multiobjective Optimization on a Budget of 250 Evaluations. In Ehrgott, M., Fonseca, C., Gandibleux, X., Hao, J.-K., and Sevaux, M., editors, *Evolutionary Multi-Criterion Optimization SE - 8*, volume 5467 of *Lecture Notes in Computer Science*, pages 36–50. Springer Berlin Heidelberg, 2009.

Knowles, J. D. and Corne, D. W. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, jun 2000.

Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H. Reconfigurable Manufacturing Systems. *CIRP Annals - Manufacturing Technology*, 48(2):527–540, 1999.

Koren, Y., Shpitalni, M., Gu, P., and Hu, S. Product Design for Mass-Individualization. *Procedia CIRP*, 36:64–71, 2015.

Koren, Y. and Shpitalni, M. Design of Reconfigurable Manufacturing Systems. *Journal of Manufacturing Systems*, 29(4):130–141, oct 2010.

Krishnan, R. *Electric Motor Drives - Modeling, Analysis, And Control.* Prentice Hall, 2001. ISBN 0-13-091014-7.

Kurapati, A. and Azarm, S. Immune Network Simulation With Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization. *Engineering Optimization*, 33 (2):245–260, dec 2000.

Kuroiwa, D. and Lee, G. M. On Robust Multiobjective Optimization. *Vietnam J. Math*, 40(2&3):305–317, 2012.

Landers, R. G., Min, B.-K., and Koren, Y. Reconfigurable Machine Tools. *CIRP Annals - Manufacturing Technology*, 50(1):269–274, jan 2001.

Lara, A., Alvarado, S., Salomon, S., Avigad, G., Coello, C. A. C., and Schütze, O. The Gradient Free Directed Search Method as Local Search within Multi-Objective Evolutionary Algorithms. In Schütze, O., Coello Coello, C. A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., and Legrand, P., editors, *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 153–168. Springer Berlin Heidelberg, 2013.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary computation*, 10(3):263–82, jan 2002.

Lawler, E. L. and Wood, D. E. Branch-and-Bound Methods: A Survey. *Operations Research*, 14(4):699–719, aug 1966.

Lee, K.-H. and Park, G.-J. Robust Optimization Considering Tolerances of Design Variables. *Computers & Structures*, 79(1):77–86, jan 2001.

Lewis, P. K., Mattson, C. A., and Murray, V. R. An Engineering Design Strategy for Reconfigurable Products. In *36th Design Automation Conference*, pages 249–263. ASME, 2010.

Li, R., Chang, T., Wang, J., and Wei, X. Multi-Objective Optimization Design of Gear Reducer Based on Adaptive Genetic Algorithm. *Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, pages 229–233, apr 2008a.

Li, X., Symmons, G. R., and Cockerham, G. Optimal Design of Involute Profile Helical Gears. *Mechanism and Machine Theory*, 31(6):717–728, aug 1996.

Li, Y., Xue, D., and Gu, P. Design for Product Adaptability. *Concurrent Engineering*, 16(3):221–232, sep 2008b.

Lu, K., Branke, J., and Ray, T. Improving Efficiency of Bi-level Worst Case Optimization. In Handl, J., Hart, E., Lewis, P. R., López-Ibáñez, M., Ochoa, G., and Paechter, B., editors, *Parallel Problem Solving from Nature – PPSN XIV: 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings*, volume 9921, pages 410–420. Springer International Publishing, 2016.

Mani, M., Singh, A. K., and Orshansky, M. Joint Design-Time and Post-Silicon Minimization of Parametric Yield Loss using Adjustable Robust Optimization. In *2006 IEEE/ACM International Conference on Computer Aided Design*, pages 19–26, nov 2006.

Marler, R. and Arora, J. Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, apr 2004.

Martinez, M. and Xue, D. Development of Adaptable Products Based on Modular Design and Optimization Methods. *Procedia CIRP*, 50:70–75, 2016.

Mattson, C. A. and Messac, A. Concept Selection Using s-Pareto Frontiers. *AIAA Journal*, 41(6):1190–1198, jun 2003.

Mattson, C. a. and Messac, A. Pareto Frontier Based Concept Selection Under Uncertainty, with Visualization. *Optimization and Engineering*, 6(1):85–115, mar 2005.

Maxon. Maxon Motor online catalog, 2014. URL `http://www.maxonmotor.com/maxon/view/catalog/`.

McKay, M. D., Beckman, R. J., and Conover, W. J. Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21(2):239–245, may 1979.

Meneghini, I. R., Guimarães, F. G., and Gaspar-Cunha, A. Competitive Coevolutionary Algorithm for Robust Multi-Objective Optimization: The Worst Case Minimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 586–593, 2016.

Miettinen, K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

Mirjalili, S. and Lewis, A. Hindrances for Robust Multi-Oobjective Test Problems. *Applied Soft Computing*, 35:333–348, oct 2015.

Mogalapalli, S. N., Magrab, E. B., and Tsai, L. W. A CAD System for the Optimization of Gear Ratios for Automotive Automatic Transmissions. Technical report, University of Maryland, 1992. URL `http://hdl.handle.net/1903/5299`.

Nair, V. N., Abraham, B., Mackay, J., Nelder, J. A., Box, G., Phadke, M. S., Kacker, R. N., Sacks, J., Welch, W. J., Lorenzen, T. J., Shoemaker, A. C., Tsui, K. L., Lucas, J. M., Taguchi, S., Myers, R. H., Vining, G., and Wu, C. F. J. Taguchi's Parameter Design: A Panel Discussion. *Technometrics*, 34(2):127–161, 1992.

Nebro, A. J., Durillo, J. J., and Vergne, M. Redesigning the jMetal Multi-Objective Optimization Framework. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO Companion '15, pages 1093–1100, New York, NY, USA, 2015. ACM.

Newport Corporation. Vibration Control - Identifying and Controlling Vibrations in the Workplace, 2012. URL `http://photonics.com/edu/Handbook.aspx`.

Nguyen, T. T., Yang, S., and Branke, J. Evolutionary Dynamic Optimization: A Survey of the State of the Art. *Swarm and Evolutionary Computation*, 6:1–24, oct 2012.

Okudan, G. E. and Tauhid, S. Concept Selection Methods A Literature Review from 1980 to 2008. *International Journal of Design Engineering*, 1(3):243, 2008.

Olewnik, A. and Lewis, K. A Decision Support Framework for Flexible System Design. *Journal of Engineering Design*, 17(1):75–97, jan 2006.

Olewnik, A., Brauen, T., Ferguson, S., and Lewis, K. A Framework for Flexible Systems and Its Implementation in Multiattribute Decision Making. *Journal of Mechanical Design*, 126(3):412, 2004.

Osyczka, A. An Approach to Multicriterion Optimization Problems for Engineering Design. *Computer Methods in Applied Mechanics and Engineering*, 15(3):309–333, sep 1978.

Paenke, I., Branke, J., and Jin, Y. Efficient Search for Robust Solutions by Means of Evolutionary Algorithms and Fitness Approximation. *Evolutionary Computation, IEEE Transactions on*, 10(4):405–420, 2006.

Pflug, G. C. and Pichler, A. *Multistage Stochastic Optimization*. Springer, Cham, 2014. ISBN 978-3-319-08843-3.

Phadke, M. S. *Quality Engineering Using Robust Design*. Prentice Hall PTR, Englewood Cliffs, NJ, USA, 1st edition, 1989. ISBN 0137451679.

Poles, S. and Lovison, A. A Polynomial Chaos Approach to Robust Multiobjective Optimization. In Deb, K., Greco, S., Miettinen, K., and Zitzler, E., editors, *Hybrid and Robust Approaches to Multiobjective Optimization*, volume 1 of *Dagstuhl Seminar Proceedings*, pages 1–15, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

Purshouse, R. C. *On the Evolutionary Optimisation of Many Objectives*. PhD thesis, The University of Sheffied, 2003.

Rao, C. R. Factorial Experiments Derivable from Combinatorial Arrangements of Arrays. *Journal of Royal Statistical Society*, 9(Series B):128–139, 1947.

Reyes-Sierra, M. and Coello, C. A. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

Rezvanian, A. and Meybodi, M. Tracking Extrema in Dynamic Environments Using a Learning Automata-Based Immune Algorithm. In Kim, T.-h., Yau, S., Gervasi, O., Kang, B.-H., Stoica, A., and Ślęzak, D., editors, *Grid and Distributed Computing, Control and Automation SE - 22*, volume 121 of *Communications in Computer and Information Science*, pages 216–225. Springer Berlin Heidelberg, 2010.

Roos, F., Johansson, H., and Wikander, J. Optimal Selection of Motor and Gearhead in Mechatronic Applications. *Mechatronics*, 16(1):63–72, feb 2006.

Saleh, J. H., Mark, G., and Jordan, N. C. Flexibility: A Multi-Disciplinary Literature Review and a Research Agenda for Designing Flexible Engineering Systems. *Journal of Engineering Design*, 20(3):307–323, may 2009.

Salomon, S., Avigad, G., Fleming, P. J., and Purshouse, R. C. Optimization of Adaptation - A Multi-Objective Approach for Optimizing Changes to Design Parameters. In Purshouse, R., editor, *7th International Conference on Evolutionary Multi-Criterion Optimization*, volume 7811 of *Lecture Notes in Computer Science*, pages 21–35. Springer Berlin Heidelberg, 2013a.

Salomon, S., Avigad, G., Goldvard, A., and Schütze, O. PSA A New Scalable Space Partition Based Selection Algorithm for MOEAs. In Schütze, O., Coello Coello, C. A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., and Legrand, P., editors, *EVOLVE 2013*, volume 175 of *Advances in Intelligent Systems and Computing*, pages 137–151. Springer Berlin Heidelberg, 2013b.

Salomon, S., Avigad, G., Fleming, P. J., and Purshouse, R. C. Active Robust Optimization - Enhancing Robustness to Uncertain Environments. *IEEE Transactions on Cybernetics*, 44(11):2221–2231, 2014.

Salomon, S., Purshouse, R. C., Avigad, G., and Fleming, P. J. An Evolutionary Approach to Active Robust Multiobjective Optimisation. In Gaspar-Cunha, A., Henggeler Antunes, C., and Coello, C. C., editors, *8th International Conference on Evolutionary Multi-Criterion Optimization*, volume 9019 of *Lecture Notes in Computer Science*, pages 141–155. Springer International Publishing, 2015.

Salomon, S., Avigad, G., Purshouse, R. C., and Fleming, P. J. Gearbox design for uncertain load requirements using active robust optimization. *Engineering Optimization*, 48(4):652–671, apr 2016a.

Salomon, S., Purshouse, R. C., Giagkiozis, I., and Fleming, P. J. A Toolkit for Generating Scalable Stochastic Multiobjective Test Problems. In *Proceedings of the 2016 Annual Conference on Genetic and Evolutionary Computation*, pages 597–604, Denver, CO, USA, 2016b.

Savsani, V., Rao, R. V., and Vakharia, D. P. Optimal Weight Design of a Gear Train Using Particle Swarm Optimization and Simulated Annealing Algorithms. *Mechanism and Machine Theory*, 45(3):531–541, mar 2010.

Schaffer, J. D. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, 1985.

Scheffe, H. Experiments With Mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):344–360, 1958.

Schott, J. R. *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.* Ms thesis, Massachusetts Institute of Technology, 1995.

Schütze, O., Esquivel, X., Lara, A., and Coello, C. A. C. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *Evolutionary Computation, IEEE Transactions on*, 16(4):504–522, 2012.

Sethi, A. K. and Sethi, S. P. Flexibility in Manufacturing: A Survey. *International Journal of Flexible Manufacturing Systems*, 2(4):289–328, jul 1990.

Shafer, G. *A Mathematical Theory of Evidence.* Princeton university press, Princeton, New Jersey, 1976.

Shim, V. A., Tan, K. C., Chia, J. Y., and Al Mamun, A. Multi-objective Optimization with Estimation of Distribution Algorithm in a Noisy Environment. *Evolutionary computation*, 21(1):149–177, mar 2013.

Siddiqi, A., De Weck, O. L., and Iagnemma, K. Reconfigurability in Planetary Surface Vehicles - Modelling Approaches and Case Study. *Journal of the British Interplanetary Society*, 59(12):450–460, 2006.

Srinivas, N. and Deb, K. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, sep 1994.

Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation, and Application.* John Wiley and Sons, Inc, 1986.

Storn, R. and Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4): 341–359, 1997.

Swantner, A. and Campbell, M. I. Topological and Parametric Optimization of Gear Trains. *Engineering Optimization*, 44(11):1351–1368, nov 2012.

Syberfeldt, A., Ng, A., John, R. I., and Moore, P. Evolutionary Optimisation of Noisy Multi-Objective Problems Using Confidence-Based Dynamic Resampling. *European Journal of Operational Research*, 204(3):533–544, aug 2010.

Taguchi, G. *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*, volume 1. UNIPUB/Kraus International Publications White Plains, NY, 1987.

Tan, Y. Y., Jiao, Y. C., Li, H., and Wang, X. K. A Modification to MOEA/D-DE for Multiobjective Optimization Problems with Complicated Pareto Sets. *Information Sciences*, 213(x):14–38, 2012.

Teich, J. Pareto-Front Exploration with Uncertain Objectives. In Zitzler, E., Thiele, L., Deb, K., Coello Coello, C., and Corne, D., editors, *Evolutionary Multi-Criterion Optimization SE - 22*, volume 1993 of *Lecture Notes in Computer Science*, pages 314–328. Springer Berlin Heidelberg, 2001.

Thompson, D. F., Gupta, S., and Shukla, A. Tradeoff Analysis in Minimum Volume Design of Multi-Stage Spur Gear Reduction Units. *Mechanism and Machine Theory*, 35(5):609–627, may 2000.

Thunnissen, D. P. *Propagating and Mitigating Uncertainty in the Design of Complex Multidisciplinary Systems.* PhD thesis, California Institute of Technology, 2005.

Triantis, A. J. and Hodder, J. E. Valuing Flexibility as a Complex Option. *The Journal of Finance*, 45(2):549–565, jun 1990.

Trojanowski, K. and Wierzchoń, S. T. Immune-based Algorithms for Dynamic Optimization. *Information Sciences*, 179(10):1495–1515, apr 2009.

Trosset, M. W. Taguchi and Robust Optimization. Technical report, Department of Computational \& Applied Mathematics, Rice University, Houston, TX, 1996.

Van Veldhuizen, D. A. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.* Phd dissertation, Graduate School of Engineering of the Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA, 1999.

Vasile, M., Minisci, E., and Wijnands, Q. Approximated Computation of Belief Functions for Robust Design Optimization. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences, pages 1–18. American Institute of Aeronautics and Astronautics, apr 2012.

Walpole, R. E., Meyers, R. H., Meyers, S. L., and Ye, K. *Probability & Statistics for Scientists & Engineers.* Pearson Prentice Hall, New Jersey, 8th edition, 2007. ISBN 0131877119.

Wang, H.-P. H. Optimal Engineering Design of Spur Gear Sets. *Mechanism and Machine Theory*, 29(7):1071–1080, oct 1994.

Wang, R., Zhang, T., and Guo, B. An Enhanced MOEA/D Using Uniform Directions and a Pre-Organization Procedure. *2013 IEEE Congress on Evolutionary Computation, CEC 2013*, (70971132):2390–2397, 2013.

Wiener, N. The Homogeneous Chaos. *American Journal of Mathematics*, 60(4):897–936 CR – Copyright © 1938 The Johns Hopk, oct 1938.

Wolpert, D. H. and Macready, W. G. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

Wood, C. D., Lewis, P. K., and Mattson, C. A. Modular Product Optimization to Alleviate Poverty: An Irrigation Pump Case Study. In *38th Design Automation Conference*, pages 455–462. ASME, 2012.

Xu, Y., Chen, Y., Zhang, G., and Gu, P. Adaptable Design of Machine Tools Structures. *Chinese Journal of Mechanical Engineering*, 21(3):7–15, 2008.

Xue, D., Hua, G., Mehrad, V., and Gu, P. Optimal Adaptable Design for Creating the Changeable Product Based on Changeable Requirements Considering the Whole Product Life-Cycle. *Journal of Manufacturing Systems*, 31(1):59–68, jan 2012.

Yao, W., Shi, Y., He, L., and Pamarti, S. Joint Design-time and Post-silicon Optimization for Digitally Tuned Analog Circuits. In *Proceedings of the 2009 International Conference on Computer-Aided Design*, ICCAD '09, pages 725–730, New York, NY, USA, 2009. ACM.

Yokota, T., Taguchi, T., and Gen, M. A Solution Method for Optimal Weight Design Problem of the Gear Using Genetic Algorithms. *Computers \& Industrial Engineering*, 35(34):523–526, dec 1998.

Zadeh, L. Fuzzy Sets. *Information and Control*, 8(3):338–353, jun 1965.

Zadeh, L. Fuzzy Algorithms. *Information and Control*, 12(2):94–102, feb 1968.

Zhang, J. *Adaptable Design of Mechanical Products With Robust Performance*. PhD thesis, Calgary, 2014.

Zhang, J., Chen, Y., Xue, D., and Gu, P. Robust Adaptable Design of Mechanical Products. In *18th Design for Manufacturing and the Life Cycle Conference*, Portland, Oregon, USA, 2013.

Zhang, J., Chen, Y., Xue, D., and Gu, P. Robust Design of Configurations and Parameters of Adaptable Products. *Frontiers of Mechanical Engineering*, 9(1):1–14, 2014.

Zhang, J., Xue, D., and Gu, P. Adaptable Design of Open Architecture Products with Robust Performance. *Journal of Engineering Design*, 26(1-3):1–23, mar 2015.

Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N., and Zhang, Q. Multiobjective Evolutionary Algorithms: A Survey of the State of the Art. *Swarm and Evolutionary Computation*, 1(1):32–49, mar 2011.

Zitzler, E. and Thiele, L. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.

Zitzler, E., Thiele, L., and Bader, J. On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79, feb 2010.

Zitzler, E. *Evolutionary Algorithms for Multiobjective Optimization : Methods and Applications*. Phd dissertation, Swiss Federal Institute of Technology Zurich, 1999.

Zitzler, E. and Künzli, S. Indicator-Based Selection in Multiobjective Search. In Yao, X., Burke, E., Lozano, J., Smith, J., Merelo-Guervós, J., Bullinaria, J., Rowe, J., Tio, P., Kabán, A., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN VIII SE - 84*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer Berlin Heidelberg, 2004.

Zitzler, E. and Thiele, L. Multiobjective Optimization Using Evolutionary Algorithms A Comparative Case Study. In Eiben, A., Bäck, T., Schoenauer, M., and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature PPSN V SE - 29*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer Berlin Heidelberg, 1998.

Appendix A

Calculation of the q_{ϵ_+} Indicator

The q_{ϵ_+} indicator (Zitzler et al., 2003) is a symmetric binary indicator for comparison between two sets of vectors. It is based on the concept of ϵ_+ *dominance* (Laumanns et al., 2002). A vector $\mathbf{a}$ is said to ϵ_+ dominate another vector $\mathbf{b}$, denoted as $\mathbf{a} \preceq_{\epsilon+} \mathbf{b}$, iff $\mathbf{a} \preceq \mathbf{b} + \epsilon$, where ϵ is a real number. The value of ϵ defines the dominance relation; a positive value allows a vector to ϵ_+ dominate another non-dominated vector, while a negative value requires stronger domination than the common definition.

For two sets of vectors $\underline{\mathbf{a}}, \underline{\mathbf{b}} \in \mathbb{R}^n$, the binary measure $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ is defined as the minimal value of ϵ required for every vector $\mathbf{b} \in \underline{\mathbf{b}}$ to be ϵ_+ dominated by at least one vector $\mathbf{a} \in \underline{\mathbf{a}}$. A negative value of $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ implies that all vectors in $\underline{\mathbf{b}}$ are dominated by vectors in $\underline{\mathbf{a}}$. A positive value implies that at least one vector in $\underline{\mathbf{b}}$ dominates a vector in $\underline{\mathbf{a}}$. For a minimization problem, without loss of generality, the mathematic definition of $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ as given in Zitzler et al. (2003) is:

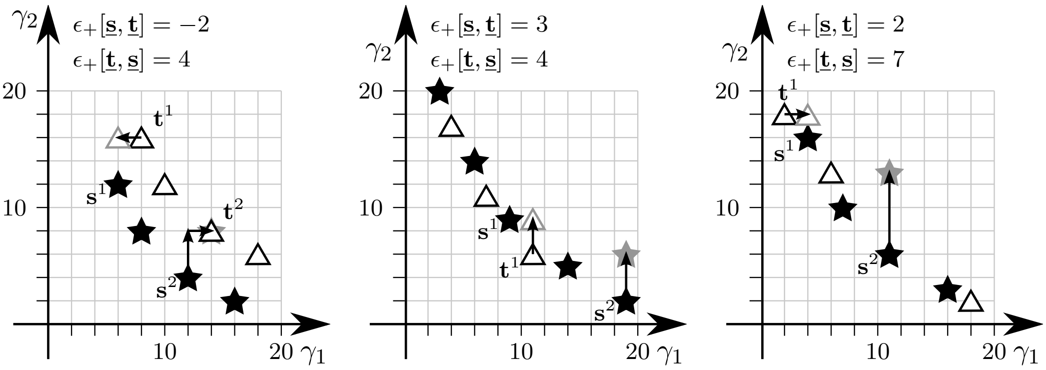
$$
\begin{aligned}
&\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = \inf_{\epsilon \in \mathbb{R}} \{\forall\, \mathbf{b} \in \underline{\mathbf{b}}\ \exists\, \mathbf{a} \in \underline{\mathbf{a}} : \mathbf{a} \preceq_{\epsilon+} \mathbf{b}\} \\
\text{where } &\mathbf{a} \preceq_{\epsilon+} \mathbf{b} \text{ if and only if:} \\
&a_i \leq b_i + \epsilon \quad \forall\, i \in \{1, \ldots, n\}
\end{aligned}
\tag{A.1}
$$

The value of $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ can be calculated by:

$$
\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = \max_{\mathbf{b} \in \underline{\mathbf{b}}} \min_{\mathbf{a} \in \underline{\mathbf{a}}} \max_{1 \leq i \leq n} a_i - b_i \tag{A.2}
$$

It can be explained as the smallest Chebishev distance that the vectors in $\underline{\mathbf{b}}$ must be displaced, in order to be all weakly dominated by vectors in $\underline{\mathbf{a}}$ (denoted as $\underline{\mathbf{a}} \preceq \underline{\mathbf{b}}$).

A demonstration of calculating ϵ_+ is given in Figure A.1. The set of stars is denoted as $\underline{\mathbf{s}}$ and the set of triangles as $\underline{\mathbf{t}}$. In Figure A.1(a) all of the vectors in $\underline{\mathbf{t}}$ are strong



(a) Dominance (b) Span (c) Distribution

Figure A.1: $\epsilon+$ comparison of two sets with apparent difference in quality for different criteria.

dominated by vectors in $\underline{\mathbf{s}}$. Therefore the vectors in $\underline{\mathbf{t}}$ can be improved and still be weakly dominated by vectors in $\underline{\mathbf{s}}$. The smallest ϵ value for which $\underline{\mathbf{s}} \preceq \underline{\mathbf{t}}$ is -2 (i.e. $\epsilon_+[\underline{\mathbf{s}}, \underline{\mathbf{t}}] = -2$). It can be seen in the figure that the triangle marked as $\mathbf{t}^1$ can be translated by -2 along the γ_1 objective and still be weakly dominated by $\mathbf{s}^1$. To allow for $\underline{\mathbf{b}} \preceq \underline{\mathbf{a}}$, the vectors in $\underline{\mathbf{a}}$ has to be translated by 2 along the γ_1 objective and 4 along the γ_2 objective. Therefore the ϵ value for this case (i.e. $\epsilon_+[\underline{\mathbf{t}}, \underline{\mathbf{s}}]$) is 4 which is the maximum among the objectives. It can be seen in the figure, as the translation of $\mathbf{s}^2$ to be weakly dominated by $\mathbf{t}^2$. The sets in Figures A.1(b) and A.1(c) are non-dominated, and therefore both $\epsilon_+[\underline{\mathbf{s}}, \underline{\mathbf{t}}]$ and $\epsilon_+[\underline{\mathbf{t}}, \underline{\mathbf{s}}]$ yield positive values. The member's translation in each set that defines the ϵ_+ value is depicted in a similar manner to Figure A.1(a).

A single ϵ_+ comparison between two sets is usually not enough to decide which one of them is better. A positive value of $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ merely implies that the set $\underline{\mathbf{b}}$ is not dominated by $\underline{\mathbf{a}}$, but as seen in Figures A.1(b) and A.1(c), it does not provide any additional information on its own. Performing a double comparison $\epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}]$ and $\epsilon_+[\underline{\mathbf{b}}, \underline{\mathbf{a}}]$ can support a decision which of the sets should be preferred. As discussed in the beginning of this section, the set $\underline{\mathbf{s}}$ is superior to $\underline{\mathbf{t}}$ in any of the panels of Figure A.1. It can be observed that for all three examples $\epsilon_+[\underline{\mathbf{s}}, \underline{\mathbf{t}}] < \epsilon_+[\underline{\mathbf{t}}, \underline{\mathbf{s}}]$. Following this observation, a comparison between two sets can be based on the value of the quality indicator

$$q_{\epsilon_+}[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = \epsilon_+[\underline{\mathbf{a}}, \underline{\mathbf{b}}] - \epsilon_+[\underline{\mathbf{b}}, \underline{\mathbf{a}}]\,. \tag{A.3}$$

It is a symmetric indicator, i.e.,

$$q_{\epsilon_+}[\underline{\mathbf{a}}, \underline{\mathbf{b}}] = -q_{\epsilon_+}[\underline{\mathbf{b}}, \underline{\mathbf{a}}]\,. \tag{A.4}$$

Therefore, a positive value means that $\underline{\mathbf{b}}$ is better than $\underline{\mathbf{a}}$, and vice versa.