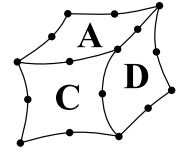




**UNIVERSITÄT  
BAYREUTH**

Fakultät für Ingenieurwissenschaften

Lehrstuhl für  
Konstruktionslehre und CAD  
Prof. Dr.-Ing. Frank Rieg



---

# Evolutionary Multi-Objective Topology Optimization for Engineering Problems

Masterarbeit

angefertigt von

**Felix Schleifer B. Sc.**

Betreuer

**Prof. Dr.-Ing. Frank Rieg**

**Kevin Deese M. Sc.**

**Juli 2017**

## Danksagung

An dieser Stelle möchte ich allen Personen danken, deren fachliche und persönliche Unterstützung zum Gelingen dieser Masterarbeit beigetragen hat.

Mein Dank gilt Herrn Prof. Dr.-Ing. Frank Rieg und seinen Mitarbeitern am Lehrstuhl für Konstruktionslehre und CAD, die es mir ermöglicht haben, dieses interessante Thema zu bearbeiten.

Besonders danken möchte ich Herrn Kevin Deese M. Sc., der diese Arbeit betreut hat und mir im Laufe der Arbeit zur Seite stand. Weiterhin gilt mein Dank Herrn Manfred Gramß, welcher mir bei der Entwicklung und Inbetriebnahme des Versuchsstandes tatkräftig zur Seite stand.

Des Weiteren gilt mein Dank meiner Familie sowie meinen Freunden und Kommilitonen, die mich während meines Studiums und der Anfertigung dieser Arbeit unterstützt, ermutigt und begleitet haben.

Besonderer Dank gilt hierbei Maria Muster B. A. und Maximilian Ries B. Sc., die diese Arbeit Korrektur gelesen haben.

# Contents

<b>List of Figures</b>	<b>IV</b>
<b>List of Algorithms</b>	<b>V</b>
<b>Nomenclature</b>	<b>VI</b>
<b>1 Introduction and Objective</b>	<b>1</b>
<b>2 Theoretical Framework</b>	<b>3</b>
2.1 Finite Element Method . . . . .	3
2.1.1 Linear Elasticity . . . . .	4
2.1.2 Stationary Thermal Conduction . . . . .	5
2.2 Numerical Optimization in Engineering . . . . .	5
2.2.1 Evolutionary Optimization . . . . .	6
2.2.2 Multi-Objective Optimization . . . . .	10
2.2.3 Structural Optimization . . . . .	14
<b>3 Optimizer Development</b>	<b>19</b>
3.1 Complexity of Evolutionary Topology Optimization . . . . .	19
3.2 Optimization Algorithm . . . . .	20
3.2.1 Binary Parameter Coding . . . . .	20
3.2.2 Choice of Algorithm . . . . .	21
3.2.3 Solution Clustering . . . . .	23
3.2.4 Parameter Setting . . . . .	24
3.3 Genotype-Phenotype Mapping . . . . .	25
3.3.1 Feasibility Filtering . . . . .	26
3.3.2 Removal of Isolated Material . . . . .	28
3.3.3 Ground Element Filtering . . . . .	31
3.3.4 Filter Combination . . . . .	32
3.4 C++ Implementation . . . . .	33
<b>4 Evaluation Setup</b>	<b>36</b>
4.1 Optimization Problems . . . . .	36
4.2 Experimental Setup . . . . .	42

<b>5</b>	<b>Results</b>	<b>46</b>
5.1	Optimizer Performance . . . . .	46
5.2	Influence of the Genotype-Phenotype Mapping . . . . .	50
5.3	Optimization Results . . . . .	52
5.3.1	Digital Image Correlation . . . . .	59
5.3.2	Simulations . . . . .	62
<b>6</b>	<b>Discussion and Outlook</b>	<b>66</b>
6.1	Discussion of the Results . . . . .	66
6.2	Evolutionary Algorithms in Topology Optimization . . . . .	68
6.3	Multi-Objective Optimization in Engineering . . . . .	69
6.4	Outlook . . . . .	69
<b>7</b>	<b>Summary</b>	<b>71</b>
<b>8</b>	<b>Zusammenfassung</b>	<b>72</b>
	<b>Bibliography</b>	<b>74</b>
	<b>Appendix</b>	<b>79</b>

# List of Figures

2.1	Discretization of a two dimensional continuum using quadrangular elements	3
2.2	Binary genetic operators to create offspring members	8
2.3	Domination and Pareto-optimality	11
2.4	Real and approximated Pareto-front	11
2.5	Non-dominated sorting and crowd distance calculation	13
2.6	The NSGA-II procedure adapted from [25]	14
2.7	The three types of structural optimization	15
2.8	Numerical instabilities of topology optimization	17
2.9	Two step smoothing of a 0-1 distribution structure	18
3.1	Comparison between three optimization approaches	22
3.2	Two exemplary structures found by wPBIL and NSGA-II	22
3.3	Projection from $\mathbb{S}$ to $\mathbb{Z}$	25
3.4	Three solutions for a design problem	26
3.5	Four steps of the filtering	27
3.6	Ground element filtering	32
3.7	Full Genotype-phenotype mapping cycle	34
4.1	Beam design domain of Problem 1.	37
4.2	Bridge-like design domain of Problem 2	38
4.3	Beam design domain of Problem 3.	39
4.4	Thermal design domain of Problems 4 and 5	40
4.5	Cubical design domain of Problem 6	41
4.6	Bearing mount-like design domain of Problem 7	42
4.7	Drawing of specimen based on design Problem 3	43
4.8	Test rig, mounting details and specimen details	44
5.1	Approximated Pareto-fronts at five different generations	46
5.2	Number of found solutions per generation	47
5.3	Variation over ten runs of NSGA-II	48
5.4	Five solutions to Problem 2 by NSGA-II and SIMP	49
5.5	Comparison between NSGA-II and SIMP approach	50
5.6	Pareto-fronts influenced by filtering	51
5.7	Filtered and unfiltered solution to Problem 3	52

5.8	Twelve solutions to Problem 3 . . . . .	53
5.9	Five solutions to Problem 6 . . . . .	54
5.10	24 solutions to Problem 4 . . . . .	55
5.11	Pareto-surface of solutions found for Problem 5 . . . . .	57
5.12	Seven solutions to Problem 5 . . . . .	57
5.13	Eight solutions to Problem 7 . . . . .	58
5.14	Unfeasible solution with random material outgrowth . . . . .	59
5.15	Five solutions to Problem 3 . . . . .	60
5.16	Digital image correlation process . . . . .	61
5.17	Results of the digital image correlation experiments . . . . .	61
5.18	Derived designs of two solutions to Problem 7 . . . . .	63
5.19	Solutions to Problem 7 in objective space . . . . .	63
5.20	Temperature distribution on bearing mount designs . . . . .	64

## List of Algorithms

2.1	Arbitrary evolutionary algorithm . . . . .	7
2.2	Classical PBIL algorithm . . . . .	9
3.1	Feasibility filtering algorithm . . . . .	29
3.2	Algorithm used to remove isolated material . . . . .	30
3.3	Genotype-phenotype mapping . . . . .	33

# Nomenclature

In this thesis the following conventions will be used for formula letters:

- Scalars will be indicated by a single letter, e. g.  $r$
- Vectorial quantities will be indicated by a single arrow, e. g.  $\vec{r}$
- Matrices will be indicated by a double arrow and upper case letters, e. g.  $\vec{\vec{R}}$
- Sets of any kind will be indicated by bold upper case letters, e. g.  $\mathbf{R}$
- Spaces will be indicated by double lined fonts, e. g.  $\mathbb{R}$
- Single entries of vectors, matrices or sets are indicated by a subscript, e. g.  $R_{ij}$

Proper names are depicted in small capitals and terms that refer to implementation details are depicted in a typewriter font. The asymptotic algorithmic complexity is indicated by the big  $\mathcal{O}$  notation. In the depicted algorithms the equal operator is separated into  $==$  and  $:=$ . The first one is a compare operator and the second one depicts the assignment of a value to a variable.

Abbreviation	Description
CPU	Central Processing Unit (Processor)
DD	Design Domain
DIC	Digital Image Correlation
DNA	Deoxyribonucleic Acid
FEA	Finite Element Analysis
GEF	Ground Element Filtering
moPBIL	Multi-Objective PBIL
NSGA-II	Non-Sorting Genetic Algorithm II
PBIL	Population Based Incremental Learning
SIMP	Solid Isotropic Material with Penalization
wPBIL	Weighted PBIL

Symbol	Dimension	Description
$A, B, C$	misc.	Arbitrary points in space
$\vec{A}$	L	Matrix used in GEF
$\mathbb{B}^N$	–	Binary decision variable space
$\vec{C}$	L	Matrix used in GEF
$d(A, B)$	L	Euclidean distance between points A and B
$e$	$ML^2T^{-2}$	Elastic energy
$e^*$	$L^2$	Simplified compliance objective
$E$	$ML^{-1}T^{-2}$	Elastic modulus, YOUNG's modulus
$f(\vec{x})$	misc.	Objective function
$F$	$MLT^{-2}$	Force
$\vec{F}$	$MLT^{-2}$	Force vector
$\mathbf{F}$	–	Set of solutions
$g(\vec{x})$	misc.	Constraining equation
$h$	–	HAMMING distance
$\tilde{h}$	–	Normalized HAMMING distance
$h(\vec{x})$	misc.	Constraining inequation
$i, j, k$	–	Arbitrary counter variables
$k$	$MT^{-2}$	Spring constant
$K$	–	Number of objectives
$\vec{K}$	$MT^{-2}$	Stiffness matrix
$L$	–	Size of a set $\mathbf{F}$
$L_R$	–	Learning rate
$m, n$	–	Arbitrary quantities
$M$	–	Number of nodes
$N$	–	Number of parameters
$\mathcal{O}$	–	Big O notation for computational complexity
$p$	–	Probability
$\vec{p}$	–	Probability vector
$\mathbf{P}$	–	Parental population
$\vec{q}$	$MT^{-3}$	Heat flux density
$\mathbf{Q}$	–	Offspring population
$r$	–	Relative mass



Symbol	Dimension	Description
$\vec{r}$	L	Center of mass
$\mathbf{R}$	–	Combined population
$\mathbb{R}^N$	–	Decision variable space, parameter space
$\mathbb{R}^K$	–	Objective space
$\mathbb{S}$	–	Feasible parameter space
$t$	–	Generation counter
$T$	$\Theta$	Temperature
$\vec{T}$	L	Matrix used in GEF
$u$	L	Displacement
$\vec{u}$	L	Displacement vector
$V$	$L^3$	Volume
$w$	–	Objective weighting factor
$W$	–	Bandwidth of a sparse matrix
$x, y, z$	L	Spatial coordinates
$\vec{x}$	misc.	Vector of decision variables
$\vec{z}$	–	Vector of objectives
$\mathbb{Z}$	–	Feasible objective space
$\alpha$	–	SIMP parameter
$\epsilon$	–	Neighborhood
$\varepsilon$	–	GEF parameter
$\lambda$	$ML^1T^{-3}\Theta^{-1}$	Thermal conductivity
$\mu$	–	Population size
$\nu$	–	POISSON's ratio
$\rho$	$ML^{-3}$	Density
$\vec{\rho}$	–	Density distribution
$\sigma$	$ML^{-1}T^{-2}$	VON MISES equivalent stress
$\omega$	–	Weighting factor
$\prec$	–	Domination
$\prec_n$	–	Crowded-comparison operator

# 1 Introduction and Objective

In engineering optimization is a mathematical strategy aiming at manipulating a model of a real system so that one or more properties and specific values of the system are improved. For the diverse demands of modern engineering a series of different specific optimization methods have been developed. The system response is provided by a computer simulation, so the procedure can be carried out fully automated. Computer-based optimization methods play an integral role in the design of e. g. production processes, air routes, control systems and geometric structures [1].

For the engineering design process specialized methods were developed aiming to find optimal distributions of solid material. An expensive trial and error based development can be avoided by such cost and time efficient structural optimization procedures. Nowadays, especially in the industrial sector of mobility, where lighter structures lead to better performing and less fuel consuming products, the use of computer aided optimization is omnipresent and almost inevitable [2]. The topology optimization itself is a fundamental design step that requires little to no a priori knowledge about the structure, yet only information about the available design space and the expected loading conditions [3].

Multi-objective optimization aims to find a set of solutions to a problem that can not be outperformed with respect to all optimization goals. The set of trade-off solutions that is the result of such an multi-objective optimization can lead to more specific insights about the design problem in comparison to classical approaches [4]. Amongst the approaches to solve a multi-objective optimization problem the evolutionary algorithms have proven to be promising choices. Evolutionary algorithms have already been used for multi-objective topology optimization [5–8] and have demonstrated the ability to provide the engineer with a set of near-application trade-off solutions [9–11].

The objective of this thesis is to develop a multi-objective topology optimization method for engineering design. The most promising algorithm from literature has to be chosen and, if necessary, expanded to account for problem-specific demands. Because most examinations have only yet focused on showing that multi-objective structural optimization is possible, in this thesis more near-application problems will be tackled. This includes large structures, non-rectangular design spaces, and three dimensional models. Optimization goals will be up to three independent objectives derived from mechanical and thermal calculations with up to three different constraints. Methods to reduce computational effort and to enhance the usability will be presented and implemented.

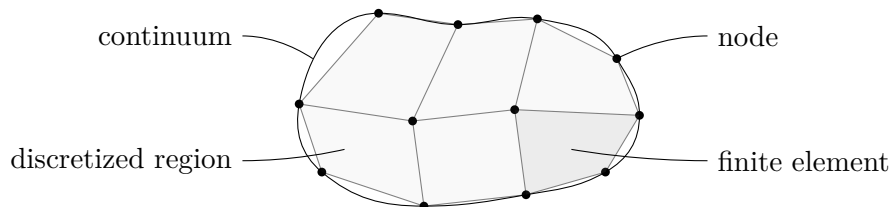
The found optimization results will be used to derive sets of trade-off design propositions. The found solutions will be discussed with respect to their qualities and the performance of the optimizer. Physical experiments and simulations will be carried out to evaluate, whether the developed optimization tool can reliably find feasible solutions to the proposed problems. The applicability of the presented method in an engineering design process will be discussed.

## 2 Theoretical Framework

In the following chapter the theoretical framework will be presented. First, the linear finite element method will be discussed as a means to provide system responses for mechanical and thermal models. After introducing the concepts of evolutionary algorithms and multi-objective optimization the state of art in numerical multi-objective optimization will be illustrated. The concepts of structural optimization and especially topology optimization in engineering shall be illustrated afterwards.

### 2.1 Finite Element Method

The finite element method is a numerical approach to solving boundary value problems of partial differential equations. An arbitrary shaped continuum is discretized by dividing it into a set of simple-shaped continua, the so-called finite elements. The entirety of the elements make up the mesh. The elements are defined by a set of points in space that are referred to as the nodes [12]. The discretization of an arbitrary continuum using quadrangular elements is illustrated in Figure 2.1.



**Figure 2.1:** Discretization of a two dimensional continuum using quadrangular elements.

The elements are chosen according to the demands of the meshing algorithm, assumptions about the state variables, arising numerical issues, and geometrical features. The discretization level and the appropriate choice of the elements have large influence on the correctness of the results and on the performance of the solver [13].

For the stationary problems that will be covered in this thesis, the partial differential equations will be approximated locally by a set of algebraic equations. These equations can be solved for every element and are compiled to an overall solution. For continuum elements it is possible to solve the problems by numerical methods. For this purpose, the state variables are approximated over the element volume by polynomial shape functions. The degrees of freedom are defined at the nodes. By this means, it is possible to describe the course of the state variables over the whole continuum [12]. In order to achieve realistic results, it is important to either model structure using a sufficient number of

elements or to use shape polynomials of a higher degree in order to correctly describe the local state variable field [13].

The finite element method proved to be a very effective tool for many applications, including mechanical, thermal, electrostatic, and fluid dynamic simulation. Many commercial software products are available and the method has become a standard tool in engineering and science. Mechanical finite element analysis (FEA) is a basic step of the strength calculation in engineering. The following chapters will cover details of the solution of linear elasticity problems and problems of stationary thermal conduction.

### 2.1.1 Linear Elasticity

In mechanical finite element calculations the basic principles of continuum mechanics are used to find equations to describe the deformation behavior of parts under mechanical loading. Apart from the conservation equations for mass, momentum, and energy, a material law is required. Linear elasticity describes material behavior in which deformation and loading are always proportional. Material failure, plasticity and other non-linearities are neglected [12, 14].

Linear elasticity is described by the three dimensional HOOKE's law. In the method of finite elements the deformation is approximated by the spatial displacement of the nodes that are the degrees of freedom of the system [12]. The one dimensional HOOKE's law, also known as the spring equation, is given in Equation 2.1, where  $u$  is the displacement or the deformation of a spring,  $k$  is the spring constant or the stiffness of the spring and  $F$  is the force applied to the spring.

$$u = k^{-1} \cdot F \quad (2.1)$$

For stationary problems the equation is a linear function of the force, when  $k$  is not a function of  $F$  nor  $u$  and  $F$  is not a function of  $u$ . The first condition is fulfilled for small displacements and the second condition is fulfilled for models without contact or spatially dependent loading conditions. Henceforth only linear problems with isotropic materials will be tackled [14]. Non-plastic deformation of poly-crystalline alloys at room temperature is a good example for a problem that fulfills all of these criteria.

For any element, the nodal displacement must be described as a function of the applied forces. The conservation equations and the material law can be transformed into a weak form, a set of algebraic equations, by variational calculus. The so-called element stiffness matrix  $\vec{K}^e$ , that maps all degrees of freedom to the applied nodal forces in all spatial directions, is the element-wise solution to these equations. For the definition of  $\vec{K}^e$ , two independent material properties and all the element's nodal coordinates are required. Usually the modulus of elasticity  $E$  and the POISSON's ratio  $\nu$  are used. All element stiffness matrices are subsequently compiled into a single system of equations and the boundary conditions are included. This results in a linear system of equations of the

form shown in Equation 2.2, where  $\vec{K}$  is the overall stiffness matrix and  $\vec{u}$  and  $\vec{F}$  are the nodal displacements and forces, respectively [12].

$$\vec{K}\vec{u} = \vec{F} \quad (2.2)$$

There are two possible types of boundary conditions: forces and displacements. In order to obtain a unique solution, the degrees of freedom in every direction have to be prescribed at at least one node. The linear system of equations can be solved by classical methods as well as by more sophisticated ones, that exploit the sparse occupation of the overall stiffness matrices. Direct and iterative solvers are available. In a subsequent step, the displacement field can be used to calculate local strains and stresses, in order to gain additional information about the mechanical properties of the system [12].

### 2.1.2 Stationary Thermal Conduction

Thermal conduction is the predominant way of heat transfer in solids. In steady state, heat is transferred according to FOURIER's equation given in Equation 2.3. The heat flux density  $\vec{q}$  is defined as the thermal power per area.  $\nabla T$  is the local temperature gradient and  $\lambda$  is a material constant, the so-called thermal conductivity [15].

$$\vec{q} = -\lambda\nabla T \quad (2.3)$$

Analogous to the linear elasticity, Equation 2.3 can also be solved by the finite element approach. Nodal temperatures are the degrees of freedom of the system and heat sources and drains can be modeled by applying nodal heat fluxes. If  $\lambda$  and the boundary conditions are independent of the temperature, the problem is also linear and can be solved analogously to a linear elastic problem. Any surface that has no boundary conditions applied to it, is treated as if it was perfectly isolated [15].

In real life problems these assumptions are rarely justified. Boundary conditions that model any kind of natural heat drains depend strongly on the surface temperature and  $\lambda$  is fundamentally a function of the temperature. The linear assumptions are only valid for very small temperature deviations.

## 2.2 Numerical Optimization in Engineering

Current problems in the applied sciences can only be solved by using specialized optimization methods. Some problems can be handled with skilled heuristic approaches but computational optimization methods have proven to be very useful tools [16].

An optimization aims to minimize an objective function  $f(\vec{x})$  by finding a vector of  $N$  optimal parameters  $\vec{x}^* \in \mathbb{R}^N$ .  $\mathbb{R}^N$  is the decision variable space or the parameter space. In application this is mostly done considering one or more constraining equations  $g_i(x)$  or inequations  $h_i(x)$  as shown in Equation 2.4. If not stated otherwise, all following

optimization problems will be minimization problems. Maximization of an objective function  $f(\vec{x})$  can be done by minimizing  $-f(\vec{x})$  [17].

$$\min\{f(\vec{x})|\vec{x} \in \mathbb{R}^N|g_i(\vec{x}) = 0, i = 1, \dots, n; h_j(\vec{x}) \leq 0, j = 1, \dots, m\} \quad (2.4)$$

A point  $\vec{x}^* \in \mathbb{R}^N$  is called a global minimum of  $f(\vec{x})$ , if it satisfies Equation 2.5 [17].

$$f(\vec{x}^*) = \min\{f(\vec{x})|\vec{x} \in \mathbb{R}^N\} \quad (2.5)$$

If the point  $\vec{x}^*$  satisfies Equation 2.5 only for certain  $\vec{x}$  with  $|x_i - x_i^*| < \epsilon$  (for a constant positive  $\epsilon$ ),  $\vec{x}^*$  is called a local minimum of  $f(\vec{x})$  [17].

Optimization has been an important part of the industrial design process for a long time. For instance, at the automotive industry, where a steadily growing demand of more capable models is to be met. Engineers benefit from modern optimization tools that automatically conduct an iterative process that constantly yields an improved design. These mathematical methods allow the designers to efficiently solve complex problems in a way that the human mind would not be capable of [17].

### 2.2.1 Evolutionary Optimization

Evolutionary optimization algorithms are a class of meta-heuristics that mimic the basic principles of evolutionary biology: natural selection and genetic recombination [18].

In ecosystems it is assumed that those members of a population have the best chances of reproduction, that are most fit to their environment. Their genetic code, and thus their characteristics, are more likely to be passed on to their offspring and subsequent generations, respectively. By this means, over the time species adapt to their environment. A classical simplified example is the giraffe's neck. Ancestors of the giraffe had shorter necks and therefore could not reach fruit on high trees. When their population grew, food supply eventually became short. Yet, those members of the populations with longer necks could reach fruit in higher trees and therefore suffered less from the food shortage. Their better nourishment lead to greater physical strength that enabled those individuals to succeed over mating rivals and to pass on their genetic material. This mechanism is called natural selection [19].

All characteristics of an animate being are defined on molecular level by its genetic code. The deoxyribonucleic acid (DNA) is a linear macro molecule made up from four different components. The sequence of these components determines all the characteristics of the organism. The genetic code is set during the parents' reproduction. The offspring's genetic code is made up in parts of the mother's and the father's genetic code. It is also possible that due to environmental influences (e. g. chemicals, radiation), certain components of the DNA are changed. These so-called mutations happen rarely but can change the organism's biochemistry significantly [19].

## Evolutionary Algorithms

Abstractions of the mechanisms described in the previous section can be used to solve non-linear optimization problems. The variables are encoded in a string to resemble the genetic code inside the DNA. Initially, a population of parents is created randomly so that each member (or individual) has their own unique genetic code. In every iteration (usually called a generation), offspring solutions are generated via recombination of the parental genetic code. Subsequently, a new parental population is created from the former offspring to create another generation of new members. Which parents are to be preferred depends on their performance that is commonly referred to as the fitness. Evolutionary algorithms usually differ in terms of how the fitness is determined and how the offspring is generated. Most common EAs use selection, crossover and mutation operators to create an offspring population [20]. The arbitrary evolutionary algorithm is shown in pseudo-code in Algorithm 2.1.

The terms fitness and selection go hand in hand and resemble the process of natural selection. The fitness is a measure of the gene code's quality in respect to the optimization goal. In single-objective optimization the fitness can for instance be the objective function itself. A selection operator that mimics mating rivals fighting for a reproduction partner is the tournament. The classical approach is to randomly choose two members of the parental population and to then compare their fitness. The one with the better fitness is chosen for reproduction [20].

The two most common processes to create offspring genetic code are crossover and mutation operators. When crossover operators are used, two members are chosen by tournament and their gene code is then combined. In Figure 2.2 (a) a two-point crossover is shown. The genetic code of the parents is cut at two random positions. The first parent's genetic code between the cuts is subsequently replaced by the code of the second parent. Figure 2.2 (b) shows the mutation of a binary chromosome. Two of ten bits are

---

### Algorithm 2.1 Arbitrary evolutionary algorithm

---

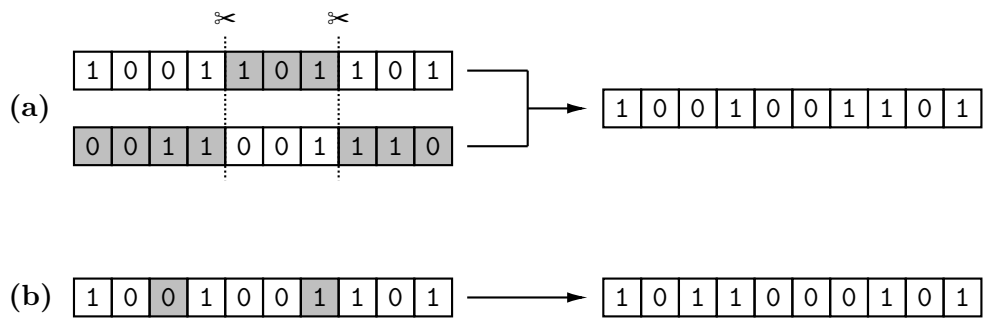
```

begin
  create_initial_population
  assign_fitness
  while no abortion criterion is met do
    for population size do
      selection
      recombination
      mutation
      (genotype-phenotype mapping)
      assign_fitness
    end
    compile_new_parental_population
  end
  output
end

```

---





**Figure 2.2:** Binary genetic operators to create offspring members. In (a) a recombination scheme using two gene codes, the two-point crossover, is shown. In (b) the mutation of a gene is shown.

flipped which corresponds roughly to a probability of mutation of one fifth. These two operations can be either consecutive (as it is implied in Figure 2.2) or exclusive. In the latter instance one would have to define a fraction of the offspring population that is first derived by crossover and subsequently mutated while the other part is only mutated. There are modified operators to work with real-number parameters but the presented binary operators can also be used to work with different parameter encodings [20].

After creating an offspring population by the genetic operators the objective functions are to be evaluated and the fitness is to be assigned. If in the following generation the whole offspring replaces the former parental population, one may lose some very fit solutions. To avoid this the concept of elitism is used. This ensures, that very fit solutions are kept as parents. The compilation of a new parental population from both, parents and offspring is therefore another important step of an evolutionary algorithm [20].

To use an evolutionary algorithm as described in this section, one has to define four parameters: population size  $\mu$ , crossover and mutation probabilities, and an abortion criterion. An ordinary abortion criterion is the definition of a maximum number of generations  $t_{max}$ . In order to avoid unfeasible solutions and to enhance the algorithm's performance genotype-phenotype mapping can be used. This concept is also taken from nature, where different genes (genotypes) may result in the same characteristics (phenotype) of the individual [19, 20]. Because this mapping is very problem specific, it will be discussed later in Section 3.3.

Evolutionary algorithms show great success in finding global optima, because they are less prone to getting lost in local optima compared to gradient-based approaches. For this purpose, a genetically diverse population is desired throughout the optimization. To exploit this concept it is crucial to avoid effects that narrow down the genetic diversity and to establish an appropriate amount of random mutations. The advantages of a large population can be wasted by using a too severe selection operator or by applying too strict constraint handling. Non-optimal or constraint-violating solutions can contain interesting geometrical features and parameter blocks that lead to efficiently finding a global optimum [20].

The ability to find global optima, of course, does not guarantee repeatable success, due to the massive use of probabilistic operators and a series of runs may have to be conducted to get reliable results. Also, it is not possible to estimate any valid abortion criteria because during the optimization a period of no improvement may occur. There are widely acknowledged guidelines as to when and how genetic algorithms may outperform other optimization approaches [20].

### Population Based Incremental Learning

Population Based Incremental Learning (PBIL) is a meta-heuristic that combines features of evolutionary algorithms and other methods from the domain of artificial intelligence for single-objective optimization. Like the classical evolutionary algorithm described in Section 2.2.1 PBIL is a population based method. But unlike the explicit storage of a parental population crucial for any evolutionary algorithm, in this case the genetic code is saved implicitly via a vector of probabilities. Every entry of this vector represents the probability of the corresponding binary parameter being a 1. Other formulations for real-coded parameters are available [18].

The probability vector  $\vec{p}$  is initialized with all entries being 0.5. Every member of the offspring population is created directly from this vector. After evaluating the objective function, every probability  $p_i$  is updated as shown in Equation 2.6. In the original formulation of the algorithm mutations are also implemented directly to the vector of probabilities [18].

$$p_i^{t+1} = (p_i^t \cdot (1 - L_R)) + (L_R \cdot x_i) \quad (2.6)$$

$L_R$  is the so-called learning rate that controls how much the probability vector is changed in every generation and  $\vec{x}_i$  is the  $i$ -th gene of the currently best member that was found during the  $t$ -th generation. The classical PBIL algorithm is shown in Algorithm 2.2.

---

#### Algorithm 2.2 Classical PBIL algorithm

---

```

begin
  initialize_probabilities := 0.5
  create_initial_population
  assign_fitness
  while no abortion criterion is met do
    update_probabilities
    mutate_probabilites
    generate_population
    assign_fitness
  end
  output
end

```

---

Several modifications of the updating scheme exist, some that also include a negative learning rate or some that also use second or third best solutions to update the probability [18]. During the optimization the vector of probability is being narrowed down to

eventually approach an optimum structure [21]. For problems that can be solved using evolutionary algorithms PBIL is an effective strategy that is less prone to getting lost in local minima because population diversity vanishes slower [21] and that is very straight forward to implement [18]. However, at least three parameters have to be set, whose influence is not as well understood as that of a classical evolutionary algorithm. The PBIL method can also be applied to multi-objective optimization problems. For this purpose, a set of probability vectors and a way to allocate members to them have to be established [8, 21].

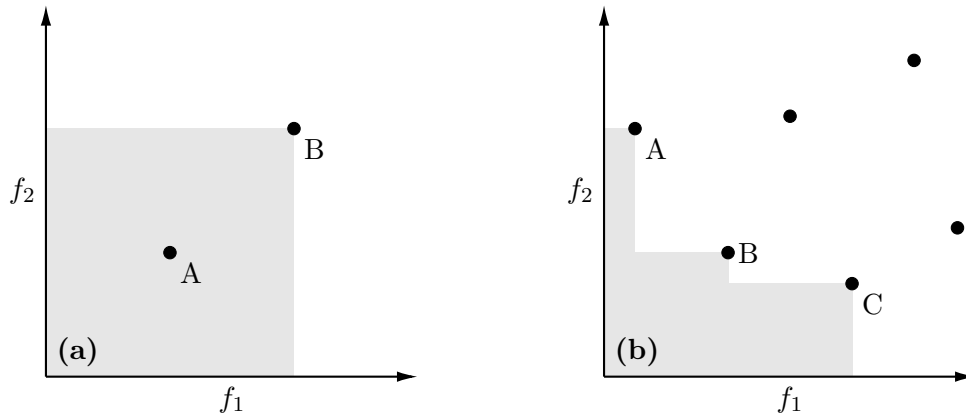
### 2.2.2 Multi-Objective Optimization

In real applications there is usually more than one independent objective function that has to be taken into account. These can either be considered as constraints to a classical optimization approach or as further objective functions. An optimization with respect to more than one objective is called a multi-objective optimization. Problems of four or more objective functions are also referred to as many-objective optimization problems [22]. The mathematical formulation of an optimization as shown in Equation 2.4 can be expanded to represent multi-objective problems. This is shown in Equation 2.7, in which  $K$  is the number of objectives and  $\mathbb{S}$  is a subset of feasible solutions inside the binary decision variable space  $\mathbb{B}^N$  formed by the constraint functions [23].

$$\min\{f_1(\vec{x}), f_2(\vec{x}), \dots, f_K(\vec{x}) | \vec{x} \in \mathbb{S}\} \quad (2.7)$$

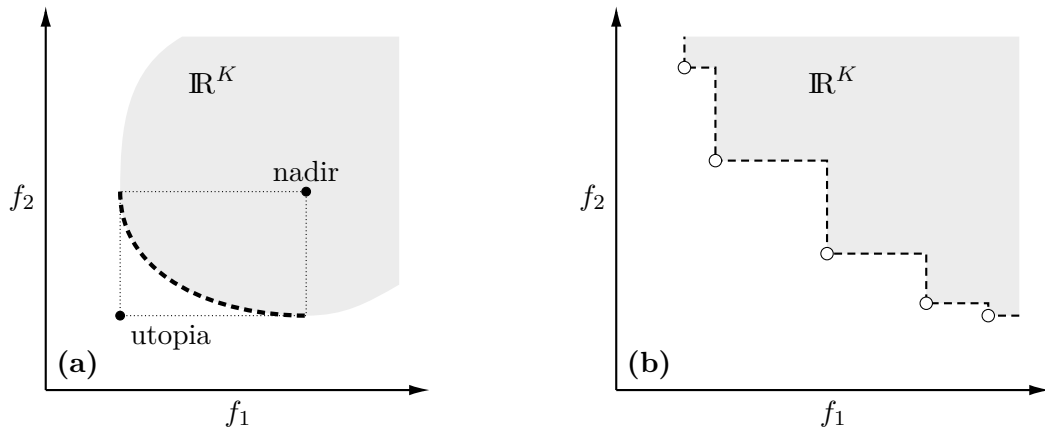
Most objective functions work in some way exclusionary to each other for instance minimization of a structure's mass and simultaneously its compliance. In general, reduction of mass leads to a less stiff structure and vice versa. Therefore it is impossible to define a single optimal point in objective space (henceforth referred to as a solution) and the concepts of domination and Pareto-optimality have to be introduced. Solution A dominates another solution B if it outperforms B in at least one objective function with the other objective values being at least equal. B is then dominated by A or in other words B is in no way better than A. If there does not exist any solution that dominates A, it is non-dominated and a single trade-off solution of the multi-objective optimization. Such a solution is called Pareto-optimal to account for the fact that an improvement of one objective function can only be achieved by accepting a decrease in another objective function [4, 24]. Domination and Pareto-optimality in two dimensional objective space are illustrated in Figure 2.3. An example for a symbolic syntax for domination,  $A \prec B$  (A dominates B), is given in [25] and will henceforth be used.

The solution of any multi-objective optimization is a set of trade-off solutions called the Pareto-front. In two dimensional objective space this front is a steadily decreasing curve, for  $K = 3$  it is a surface and in many-objective problems the Pareto-front is a hyper-surface in objective space. For a problem as shown in Equation 2.7 the Pareto-



**Figure 2.3:** Domination (a) and Pareto-optimality (b). In (a) the gray space indicates the area where potential solutions would dominate solution B. A is dominating B. In (b) solution A, B and C are non-dominated and Pareto-optimal.

front is the convex border of the feasible objective space  $\mathbb{R}^K$  [4]. Two special points can be defined: the utopia point and the nadir point. The utopia point is defined as the point in objective space made up from the best coordinates of all Pareto-optimal solutions. The utopia point can by definition never be reached. The nadir point is somewhat defined similarly. It is made up of the least optimal objective values. These concepts are shown in Figure 2.4 (a) for two objectives.



**Figure 2.4:** Real Pareto-front (a) with utopia and nadir point. Non-dominated solutions with approximated Pareto-front (b) in objective-space. The region of feasible solutions is shown in gray and the Pareto-front is shown as a dashed line. The approximated Pareto-front is the border of the region of solutions dominated by the Pareto-solutions.

An approximated Pareto-front is a set of non-dominated trade-off solutions. It represents the approximated border of the feasible objective space. Connecting the solutions with straight lines does not take into account the possibility of non-existence of solutions on this line and is therefore not suited as an illustration. Figure 2.4 (b) shows a correct approximation of a constant border. This scheme is henceforth used to plot approximated Pareto-fronts for two objective functions.

### Single-Objective Approaches

For approximating the Pareto-front it may be useful to turn the multi-objective problem into a set of single-objective problems. These can then be solved using classical single-objective optimization approaches with their specific advantages and disadvantages. The most straight forward approach is to simply weight the objective functions by factors  $\omega_i \in [0; 1]$  as shown in Equation 2.8 in order to reduce the problem to a single-objective problem.  $K - 1$  weighting factors are required [26].

$$f(\vec{x}) = \sum_{i=1}^K f_i(\vec{x}) \cdot \omega_i \quad (2.8a)$$

$$\sum_{i=1}^K \omega_i = 1 \quad (2.8b)$$

The solution of such a single objective optimization problem is expected to be member of the Pareto-front so the Pareto-front is approximated by sweeping through the weighting factors [24,26]. While this approach may seem intuitive and a simple way to approximate the Pareto-front, it has a great disadvantage. This strategy does only find convex points off the Pareto-front. Using this method for arbitrary problems may result in misleading solutions. Also, normalization of the objective functions is necessary but requires a priori knowledge about the absolute values of the objectives [24].

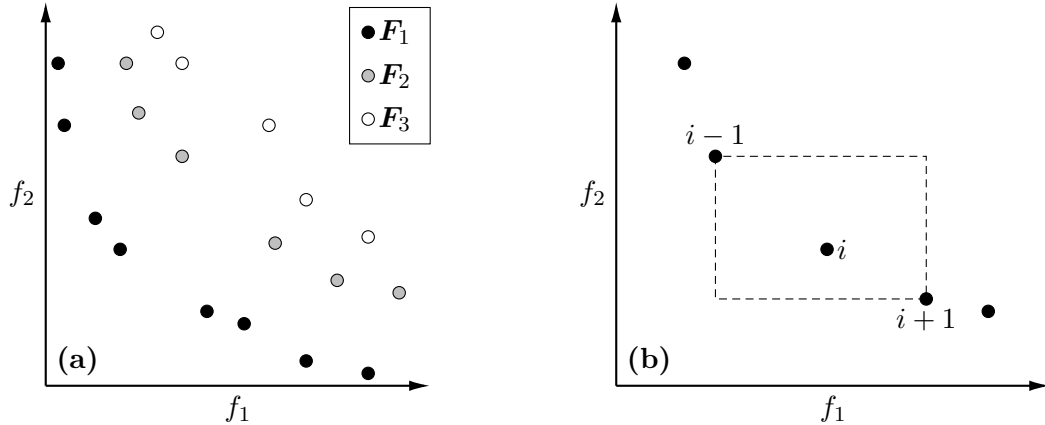
Amongst others, another notable approach to gain single-objective problems would be to keep an objective and to treat the other objectives as constraints. This is the so-called  $\epsilon$ -constraint method. But like the weighting function approach, this method has also proven to be unable to provide fully satisfying solutions to multi-objective optimization problems [24]. All approaches that project multi-objective problems to a set of single-objective problems suffer from the necessity of a discretization in objective space and potential losses due to inappropriate definitions of that discretization.

### Non-Sorting Genetic Algorithm II

More promising approaches to approximating the Pareto-front of a multi-objective optimization problem are population based. In this section an evolutionary method called Non-Sorting Genetic Algorithm II (NSGA-II) will be described that has proven to be very promising in multi-objective optimization [25] and especially in multi-objective topology optimization [5–8].

As described in Section 2.2.1 an evolutionary algorithm has to be adapted to specific problems. This happens in terms of how the objectives are scalarized into a fitness value. For multi-objective optimization one may want to assign better fitness to members of the population that are not dominated by another member [27]. The approach that NSGA-II chooses is allocating every solution to a front  $F_i$  meaning that the Pareto-front is the first front  $F_1$ . The second front  $F_2$  is defined as the set of solutions that would be non-dominated if the Pareto-front would have been removed from the offspring population.

This allocation is repeated for  $F_{i+1}$  until all solutions are allocated to a front [25]. The size of each front is  $L_i$ . This scheme is illustrated in Figure 2.5.



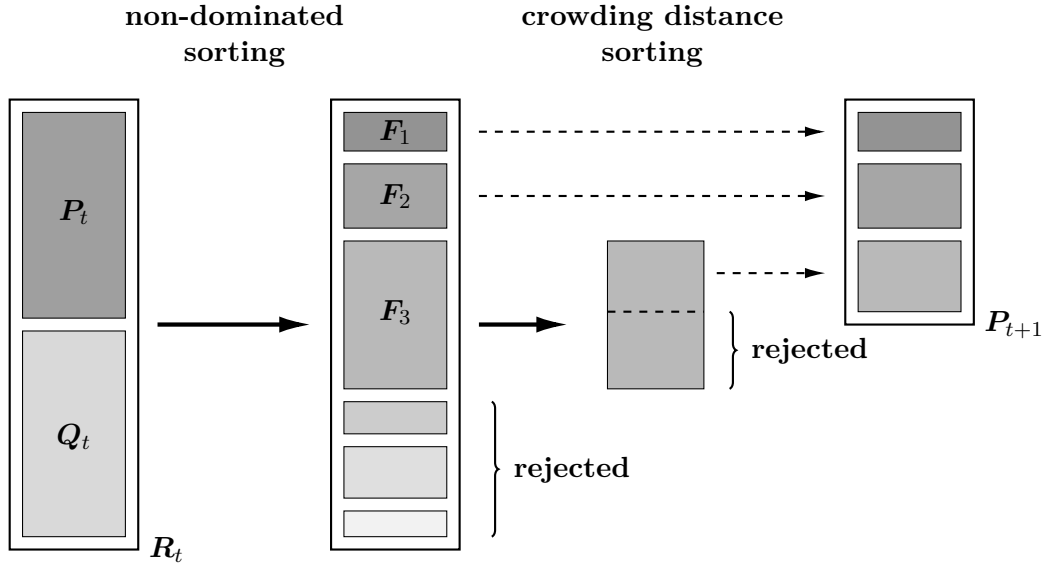
**Figure 2.5:** Non-dominated sorting (a) and crowd distance calculation (b). In (a) each solutions is assigned to a front  $F_1$  to  $F_n$  due to their non-domination level. In (b) the crowding measure is shown for the  $i$ -th member of a front ( $1 < i < L$ ). Adapted from [25].

Subsequently, the solutions are assigned integer ranks corresponding to the fronts they are allocated to. Pareto-elements therefore have a rank of 1, members of  $F_2$  have rank 2 and so on. The efficient algorithm, called non-dominated sorting, used to allocate solutions to the fronts, requires  $\mathcal{O}(K \cdot \mu^2)$  computations and is given in [25].

Apart from the ranks, another scalar is used to determine the fitness of a found solution. The crowding distance is a measure for the distribution density of solutions inside each front. For the  $i$ -th solution it is determined by adding up the distances between its closed neighbors when sorted according to each objective. Preferring solutions with a larger crowding distance may lead to a more equally dense populated Pareto-front. The solutions with minimum or maximum objective values are assigned an infinite crowding distance in order to preserve them [25]. The concept of the crowding distance is also illustrated in Figure 2.5 for a set of  $L = 5$  solutions.

Both, rank and crowding distance, together make up the fitness function used in NSGA-II. In order to compare two solutions during the tournament phase of an evolutionary algorithm, the so-called crowded-comparison operator  $\prec_n$  is introduced. Its definition is as follows:  $A \prec_n B$  if the rank of A is lower than the rank of B. If both have equal ranks,  $A \prec_n B$  if the crowding distance of A is larger than B's [25].

In NSGA-II the overall procedure of a generation, after the first population is initialized randomly, is as follows: From the parental set of solutions an offspring population is generated by tournament and recombination techniques as described in Section 2.2.1. Both populations are then merged into a combined population in order to establish elitism. All members of the combined population are then assigned ranks and crowding distances. The combined population is then sorted according to the crowded-comparison operator and subsequently truncated to fit the defined population size by removing the elements that are least fit [25]. This procedure is illustrated in Figure 2.6.



**Figure 2.6:** The NSGA-II procedure adapted from [25]. The parental population  $P_t$  and the offspring population  $Q_t$  are combined to a population  $R_t$ . The members of  $R_t$  are subsequently assigned to the fronts  $F_1$  to  $F_n$  according to their non-domination and sorted with respect to the crowding distance. The parental population of the following generation  $P_{t+1}$  is created from the most fit members of  $R_t$ .

In the original paper there is also given a straight forward approach to implement constraint handling. This is done during the tournament phase. Prior to the actual crowding-comparison the constraints are considered. If both solutions do not violate any constraints, the crowding-comparison operator applies as usual. If one solution does violate the constraints, while the other does not, the non-violating solution is preferred. If both randomly chosen solutions violate one or more constraints, the solution with the smaller overall constraint violation is to be chosen [25].

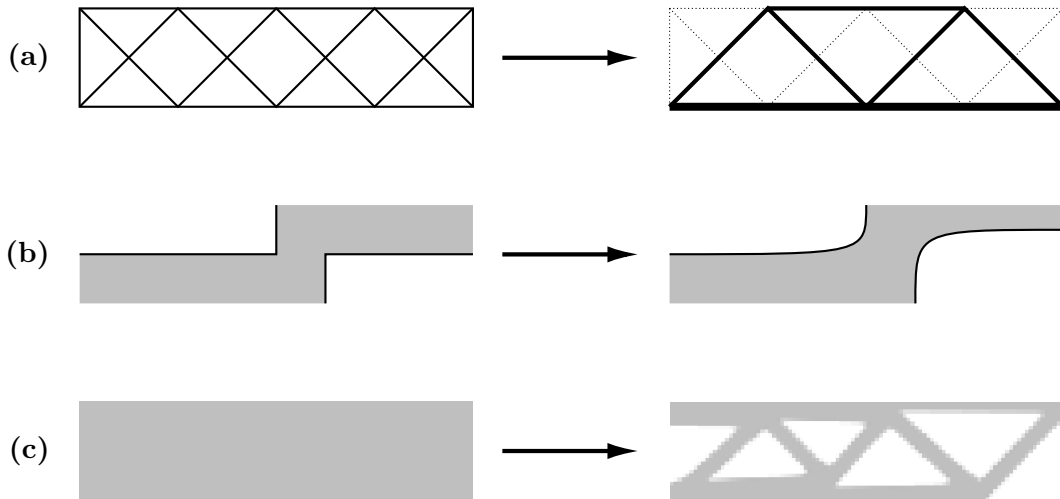
Even though NSGA-II has proven to be a very promising evolutionary approach to multi-objective optimization there are plenty of other approaches published, each with its own strengths and weaknesses. Amongst these approaches are classical algorithms as presented in [28] as well as multi-objective particle swarm optimization [29], the multi-objective version of the PBIL [8] and the many-objective version of NSGA-II [22, 30]. Each of these approaches has proven to be an appropriate choice for a special problem.

### 2.2.3 Structural Optimization

Structural optimization is an engineering method used to generate optimal structure of parts or modules. In every case, the problem aims to distribute material in an optimal way. The only a priori required quantities are an initial structure and the loading conditions. Usually constraints are defined to gain feasible results or because they are required by the optimization method [3, 31].

One can distinguish between three basic types of structural optimization: sizing, shape optimization, and topology optimization. Sizing describes an approach where the struc-

ture itself is in many ways already predefined. E. g. non-continuum elements such as bars, trusses or plates are defined in terms of their location and connections. Inside this design domain only the element parameters are to be optimized, i. e. diameters, thicknesses or element types are to be defined. This is depicted in Figure 2.7 (a), where an optimal suspension is found by appropriately setting the trusses' thicknesses [31].



**Figure 2.7:** The three types of structural optimization. Sizing (a) only affects element parameters such as the beam cross-sectional area. The shape optimization shown in (b) can effectively reduce peak stresses at notches by redistributing material. Topology optimization allows for holes to be created in the design space as it is shown in (c). Gray areas indicate continua. Based on [31].

Shape optimization however works without a prefixed design domain. The borders of the material are the design variable and are chosen appropriately. This does not allow for holes to be created but can effectively reduce stress peaks and set radii of curvature as it is illustrated in Figure 2.7 (b) [31].

Topology optimization explicitly allows for holes to be created and aims to find an optimum distribution of solid material and void inside a predefined design domain. It clearly distinguishes from sizing by explicitly aiming to find a strict 0-1 distribution of the material [31]. An example is given in Figure 2.7 (c) and this method is discussed in detail in the subsequent section.

All the problem formulations can be solved by various optimization schemes. Amongst them are purely mathematical approaches as the moving asymptotes method or empirical approaches that e. g. aim to find a fully stressed design. The aim of such optimizations is to find an optimum set of variables that makes up the resulting structure. This structure usually contains a number of geometrical features, that are distinctly shaped domains of solid material that can ultimately make up guidelines for a subsequently derived engineering design.



### Topology Optimization

In topology optimization the size, shape and topology of the structure are initially unknown. They are described by a set of distribution functions over a fixed design domain. The design domain is usually discretized by uniform finite elements with fixed position and parameters. Henceforth, only isotropic materials are taken into account to avoid the necessity of optimizing the principal direction of the material [31].

Most commonly, the desired discrete 0–1 distribution is replaced by a set of real-valued variables and the problem is solved as a sizing problem. The design variables are interpreted as the material’s physical density and penalized to steer to a 0–1 distribution again. The goal of a topology optimization is then to find an optimal vector of element densities  $\vec{\rho}$  [31]. The very popular solid isotropic material with penalization method (SIMP) maps the element density  $\rho_i \in [0, 1]$  to the local elastic modulus  $E_i$  as shown in Equation 2.9.  $E_0$  is the assigned material property of the solid material and  $\alpha$  is a parameter usually greater than 3.0 [31].

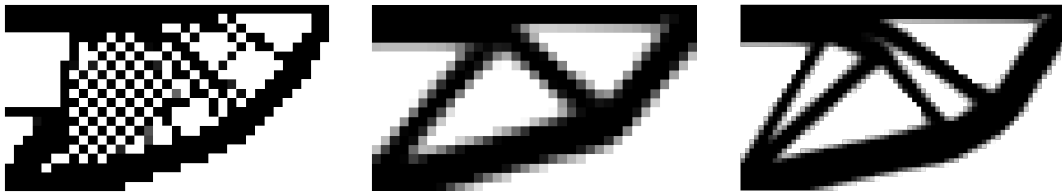
$$E_i = \rho_i^\alpha E_0 \quad (2.9)$$

Usually, the target volume is to be predefined, too. After a first guess of a homogeneous density distribution an iterative update scheme is applied to find the optimal distribution. This is done considering box constraints giving upper and lower values for the densities. Mathematically appropriate optimization algorithms are required that can cope with a large number of design variables and a moderate number of constraints. Many other formulations to solve the topology optimization problem exist beside this most common one. For practical reason one might sometimes want to prescribe regions in which no optimization is to be conducted. Initially selected variables can be set to always be 1 or 0. Henceforth, the term fixed element is going to be used for elements that are prescribed to always be solid material.

Topology optimization in general and the SIMP approach in particular are prone to a number of numerical instabilities. Apart from the possibility of converging towards a local optimum, there is the so-called checkerboard effect that occurs in most simple implementations of topology optimization. A pattern of alternating solid and void elements shows significantly greater stiffness per mass than any other structure. This is a numerical effect that has no physical meaning. Several methods exist to avoid checkerboard patterns such as the use of higher degree shape functions or filters that average the element densities in a defined neighborhood [32].

Apart from the checkerboard effect, mesh dependencies can occur due to non-uniqueness or non-existence of optimum mass distributions. This means that for the same design problem the same optimizer may obtain fundamentally different structures. For finer meshes the structure also tends to become finer and to converge towards a microstructure. This effect, like the checkerboarding, is not resulting in helpful design propositions

and is therefore to be avoided by filtering or specialized constraints [32]. An example of mesh discretization dependence and for the checkerboard effect is given in Figure 2.8. The images were generated by an 88 lines MATLAB code<sup>1</sup>. The first structure was obtained on a  $35 \times 20$  elements mesh and shows regions in which checkerboarding occurs. The second structure was found on the same mesh but checkerboarding was suppressed with a filter. The last structure was found using the same parameters as the previous structure just on a  $70 \times 40$  elements mesh. The two structures visibly differ in terms of the arrangement of truss-like features.



**Figure 2.8:** Numerical instabilities of topology optimization. Left: Structure with distinct checkerboards structures. Mid: Structure with checkerboarding suppressed by a filter. Right: Structure found on a finer mesh.

In [33] a list of challenges is published that topology optimization methods are supposed to meet in order to prove their potential in real life application. A universally applicable topology optimization method must perform well on all the following issues:

- Efficiency on three dimensional large scale problems
- Applicability to arbitrary physics problems and numerical methods
- Stable physical and geometrical constraint handling
- Mesh and starting-guess independence
- Easy to use and to set parameters

### Further Processing of the Results

The results that are obtained by topology optimization are rarely instantly usable. Even if near-application constraints were considered, direct production is not possible. Even when using additive manufacturing, it is not always advisable to use the unprocessed results of an optimization. The structures found by an optimization procedure should rather be inspiration to the designer and seen as an initial design guess. Engineering features, such as holes, threads or connectors may have to be added and manufacturing limitations are to be considered as well as the estimated costs and the aesthetic appeal. Taking these points into consideration one may obtain a well performing design proposition.

<sup>1</sup><http://www.topopt.dtu.dk/?q=node/751>

When the discretization of the design domain is too coarse or if noisy features have to be eliminated, one might want to smooth the resulting structure. When real-valued density distributions are considered, one might want to define a cut-off density first to gain a well defined voxel-like solid body. Further processing might benefit from the findings of computer graphics and medical three dimensional imaging. A way to gain a smoothed surface is the application of a modified marching cubes algorithm [34]. The resulting triangular surface mesh can be further processed using explicit or implicit iterative methods developed from signal processing methodology [35,36]. These methods show significantly better volume preservation and less shrinkage than classical approaches such as GAUSSIAN filtering. Nevertheless, smoothing is to be avoided when checkerboarding occurs [32]. A two step approach is shown in Figure 2.9, in which a two dimensional structure has been smoothed as described above.



**Figure 2.9:** Two step smoothing of a 0-1 distribution structure. First a modified marching cube algorithm is applied and second an implicit iterative fairing algorithm is used.

The typical development process using structural optimization consists of preliminary considerations about the design domain, the underlying physics and suitable numerical models. A topology optimization with subsequent smoothing might give first insights into a possible solution and further shape optimization can be applied. Further calculations and testing must be conducted to check the functionality of the derived design.

## 3 Optimizer Development

In the following sections the development of a multi-objective optimizer is described. The chosen approach will combine an evolutionary algorithm with an element-wise design parametrization  $\vec{\rho}$  known from methods such as the SIMP approach.

The first section will cover an estimate of the computational effort to solve multi-objective topology optimization problems with evolutionary methods. Therefore, in the subsequent chapters, special attention is given to the appropriate choice of the optimization algorithm and to an adequate genotype-phenotype mapping to reduce computational effort. Details about the implementation are given in a subsequent section.

### 3.1 Complexity of Evolutionary Topology Optimization

While the finite element meshes used will be small in terms of usual engineering problems, the number of finite element analyses to be carried out will be the predominant source of hardware limitations. Therefore, only the computational complexity, not memory complexity, will be considered.

Direct finite element sparse matrix solvers used to evaluate the objective functions have a computational complexity of  $\mathcal{O}(M \cdot W^2)$  [37].  $M$  is the number of nodes in the finite element mesh and  $W$  is the bandwidth of the stiffness matrix, which makes up the linear system of equations that has to be solved. Both parameters increase with a rising overall number of elements.  $W$  also depends strongly on the numbering of elements and nodes. This yields an overall complexity of  $\mathcal{O}(M \cdot W^2 \cdot \mu \cdot t_{max})$  for any population based structural optimization. Taking the density of every element as an independent binary parameter, a mesh of  $n$  elements results in a problem of  $N$  parameters. The number of nodes, bandwidth, population size and the required number of generations are usually increasing functions of  $N$  [13, 37, 38]. A deeper analysis of the proper parameter tuning will be presented in a following chapter. Now assuming  $N \propto M$ ,  $N \propto W^2$ ,  $N \propto \mu$  and  $N \propto t_{max}$  the above relation yields a overall complexity estimate of  $\mathcal{O}(N^4)$ . This is in good accordance with preliminary runs with differently sized problems that revealed that the examined problems are at least of  $\mathcal{O}(N^3)$  complexity. This is, in algorithmic theory, considered calculable but in engineering application such behavior is not tolerable. The effort for solving larger problems rather quickly reaches the limits of any, even high-end, stock computer [39].

In order to be able to solve near-application problems, a significant reduction of the computational effort is essential. There are several ways to achieve this: First, the com-

plexity can be reduced by adequate choice of algorithms. As standard multi-objective optimization algorithms are evolutionary [40], a reduction of the complexity is not directly possible. Computation time can therefore only be reduced by a constant factor. The evaluation of different members' objective functions in a generation is completely independent of the other members so parallel computing can be used to reduce not CPU but wall time (i. e. time that passes on a physical clock on a wall) significantly. Also classical modeling techniques aiming to reduce costs of the finite element analysis, as exploiting symmetries, are used. Apart from that the principle of the genotype-phenotype mapping can be used to increase the performance of an evolutionary algorithm.

## 3.2 Optimization Algorithm

This section will tackle the choice of an appropriate optimization strategy. A fitting representation of the variables will be discussed first, then the choice of a fitting optimization algorithm is presented. Further sections will tackle solution clustering and preliminary considerations concerning appropriate parameter setting.

### 3.2.1 Binary Parameter Coding

For any optimization it is crucial to know the species of the parameters that are to be optimized. In the present instance either a real-numbered representation of the element densities or a binary representation were considered. The real-numbered approach is crucial to any classical, gradient-based optimization technique, such as the SIMP approach. Whereas classical operators used in an evolutionary algorithm use binary coding (see Section 2.2.1). Although specialized operators for real-valued parameters exist these require additional parameters. A binary representation forces a strict 0-1 distribution of the material. This distribution would otherwise have to be established by additional operators that would further complicate the optimization.

The reasons listed above indicate strongly that in the present case a binary parameter coding is superior to a real-valued one. Henceforth a binary distribution of the element density will be assumed in which a 0 stands for a hole in the material and a 1 stands for solid material. No intermediate values are allowed at all. The material properties are distributed to be the predefined material property for a solid element and a sufficiently small non-zero value for a hole. This ensures that numerical issues during the finite element analysis due to arising singularities are avoided [31]. As an example for any material property the distribution of the elastic modulus is shown in Equation 3.1 in which  $E_0$  is the material property of the solid material.

$$E(\rho) = \begin{cases} E_0 & \text{for } \rho = 1 \\ 10^{-16} & \text{for } \rho = 0 \end{cases} \quad (3.1)$$

A distribution of this kind will further on be used for any material parameter. The binary approach allows for the recombination operators two-point crossover and mutation to be implemented directly as they are presented in Section 2.2.1. Crossover and mutation will be consecutive operations. The exchange of blocks of well-performing genes is established by periodic numbering of the elements. This ensures that consecutive bits inside the genetic code encode connected elements. Other traits of the binary representation will be exploited in the following sections when a clustering method and a genotype-phenotype mapping are presented. Fixed elements can be defined prior to the optimization in order to force regions to always consist of solid material.

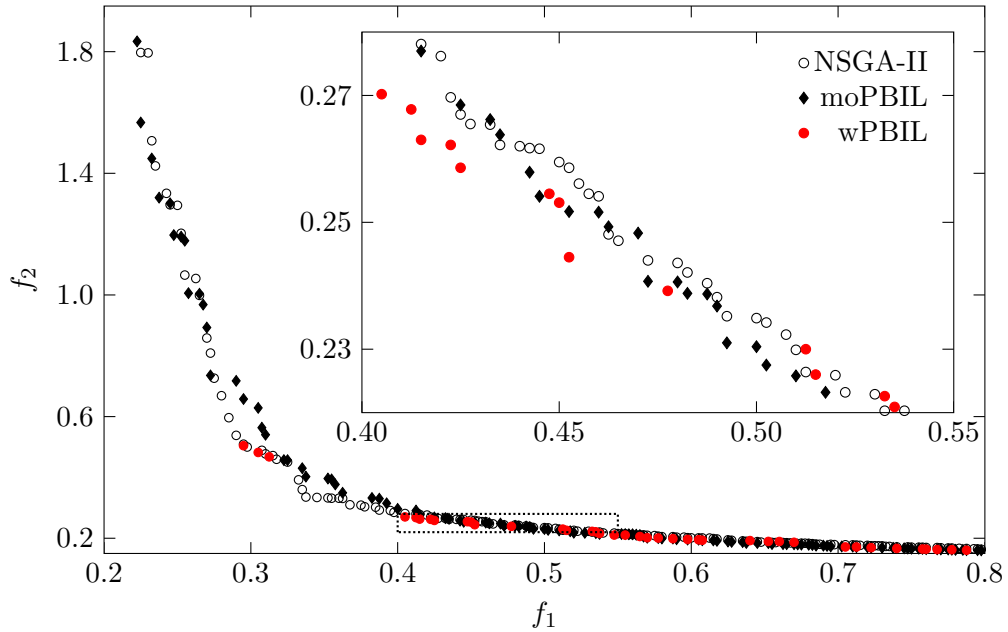
### 3.2.2 Choice of Algorithm

In preliminary studies three promising evolutionary approaches to solving multi-objective topology optimizations were implemented and tested. A single objective weighted sum approach using PBIL (wPBIL) was tested against NSGA-II and multi-objective PBIL (moPBIL). As PBIL originally works with binary parameters and has a very good reputation to be able to find global minima at least as efficient as regular evolutionary algorithms [18] it was chosen for the weighted sum approach. NSGA-II and moPBIL were both found very promising to solve topology optimization tasks [5,8] with moPBIL having a slight advantage over NSGA-II [7].

The results of the preliminary tests indicated that the wPBIL approach was able to find a couple of very well advanced non-dominated solutions for every set of weighted parameters. Nevertheless it proved to be very computationally extensive as a population based optimization had to be carried out repeatedly. The solutions were not equally distributed over the objective space either as finding the right weights is not a trivial task. The two other explicitly population based approaches performed way better in terms of the number of found solutions and equal distribution of the solutions over the objective space for comparable computational effort. In terms of domination wPBIL performed especially well compared to the other approaches. This observation is demonstrated by three Pareto-fronts in Figure 3.1.

The structures found, nevertheless, did not differ much from those found by NSGA-II, for instance. This leads to the conclusion that both approaches were heading towards the same optimum, yet wPBIL did so faster. This is depicted in Figure 3.2, where two exemplary structures are shown. As in engineering, the results of the optimization will be subject to a number of subsequent design steps, small changes in the structure may not be of interest. In this instance, an evenly filled set of Pareto-optimal solutions may provide further insights than a few well advanced solutions. It showed that the same quality of structures could be reached by the other algorithms too, by accepting a greater computational effort.

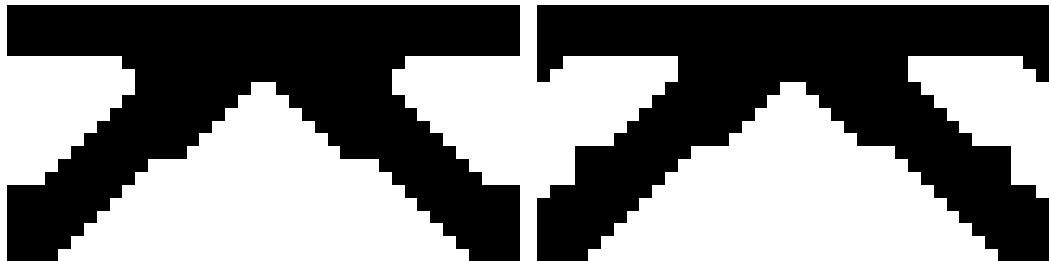
Comparing the two directly population based approaches, NSGA-II and moPBIL, it showed that, unlike it is predicted in [7], NSGA-II seems to perform slightly better



**Figure 3.1:** Comparison between three optimization approaches. Approximated Pareto-fronts are shown for NSGA-II, wPBIL and moPBIL. The local superior performance of wPBIL is emphasized in a detailed plot.

in terms of how far the Pareto-front advanced. This and the fact that NSGA-II goes with a minimum number of parameters that have to be set seem to make NSGA-II the preferable choice as an optimization algorithm. In comparison to NSGA-II, moPBIL has more parameters that have to be chosen appropriately and to which no reliable literature is available. As all parameters that have to be set for NSGA-II are common for most evolutionary algorithms, one can benefit from previous studies concerning optimum parameter tuning. This is in some ways also true for wPBIL. The issue of appropriate parameter tuning is tackled in detail in a Section 3.2.4.

A combination of an evolutionary algorithm and a subsequent weight-based local optimization was introduced in [5]. This approach was not implemented because the additional computational effort to carry out multiple local searches did not seem justified



**Figure 3.2:** Two exemplary structures found by wPBIL (left) and NSGA-II (right) with equal relative mass. The structure on the right performs slightly better in objective space but both structures' shapes seem to not differ significantly and might eventually result in the same engineering design.

taking into account the considerations outline above. As the computational effort to solve larger problems is predicted to be very large one might want to find a reasonable compromise in sacrificing precision of the results for a significantly reduced computing time.

Considerations basing on preliminary studies show that both, wPBIL and NSGA-II, seem to perform well in finding an approximated Pareto-optimal set of solutions. NSGA-II was chosen for further use, as it may provide insights into the solution space that are more relevant to engineering. Apart from that, NSGA-II is also less prone to poorly set parameters.

### 3.2.3 Solution Clustering

As described earlier, it is desirable to not only find some unevenly distributed solutions but to find a densely filled approximated Pareto-front with as many trade-off solutions as possible. With a rising number of binary parameters the actual number of fundamentally different solutions may get obscured by the amount of Pareto-optimal variations of one and the same structure. This is due to the fact that a single inverted parameter may lead to small changes in the objectives and might result in another trade-off solution. Of course, for practical reasons, in this case it may not be beneficial to regard all structures found during the optimization. Therefore, in order to reduce the number of found structures it may be best to merge very similar structures into a cluster of solutions.

A clustering method that works in objective space is presented in [5]. In topology optimization this might not be the best approach, as it is possible to ignore interesting structures only because they lie close to another solution in objective space. A clustering method that works in parameter space will be presented here in order to emphasize the fundamental structures that make up the Pareto-front. For this purpose, a trait of the representation of the structure as a binary string is exploited. In information technology the HAMMING distance  $h$  is used to find transmission errors by comparing the received string with members of a set of expected strings. In order to calculate  $h$  a bit-wise XOR operator has to be used on both structures. The HAMMING distance is then defined as the count of all 1's in the resulting binary string. If two binary string are equal  $h = 0$ . The lesser  $h$ , i. e. the less parameters differ, the more similar are two strings [41]. The calculation of the HAMMING distance is shown for two arbitrary binary strings in Equation 3.2.

$$\begin{array}{r}
 1000101001 \\
 \text{XOR } 1010001011 \\
 \hline
 0010100010 \Rightarrow h = 3 \\
 \Rightarrow \tilde{h} = 0.3
 \end{array} \tag{3.2}$$

As the structures can be expressed as binary strings, it is possible to calculate  $h$  for two structures. To make this scalar independent of the length of the strings or of the



size of the structure, respectively,  $h$  will be normalized to  $\tilde{h}$  by the number of binary parameters used. For the structures given in Figure 3.2  $h$  would be 24 and  $\tilde{h}$  would be 0.03.

The clustering method now uses  $\tilde{h}$  to decide whether two solutions are treated as being similar. For this purpose, a maximum  $\tilde{h}$  is to be defined that still qualifies as similarity. In this thesis a value of 0.1 was chosen for two dimensional structures and 0.2 for three dimensional structures. The solutions are sorted by one objective first, then  $\tilde{h}$  is calculated for consecutive structures and then similar solutions are added to the same cluster set. One solution per set is output in order to represent the whole cluster. The representing structure is chosen according to [5] amongst all solutions inside the cluster as the structure closest to the cluster's center in objective space. Henceforth, presented structures are taken from clustered solutions, if not stated otherwise.

### 3.2.4 Parameter Setting

Using NSGA-II and the presented genetic operators as the basis to develop an optimizer the user has to define four parameters: mutation rate, crossover probability, population size, and a stopping criterion. In this section an estimate about correct parameter setting is to be made for problems of a considerably high number of binary parameters. Problems of few real-numbered parameters might need considerably differently parameter setting.

The mutation rate is set to a value that results in one fitness-altering parameter flip per structure [42] and the crossover probability is set to a value between 0.9 to 1.0 to ensure that most of the results are coming from the diversity preserving crossover operations [42]. In the scope of this thesis a mutation rate of  $p_{mut} \approx N^{-1}$  and a crossover probability of 0.99 were used. [38] proposes that the optimal population size is in the same magnitude as the number of binary parameters. Smaller population sizes may initially perform better but will probably get stuck in local optima due to insufficient genetic diversity [18, 43].

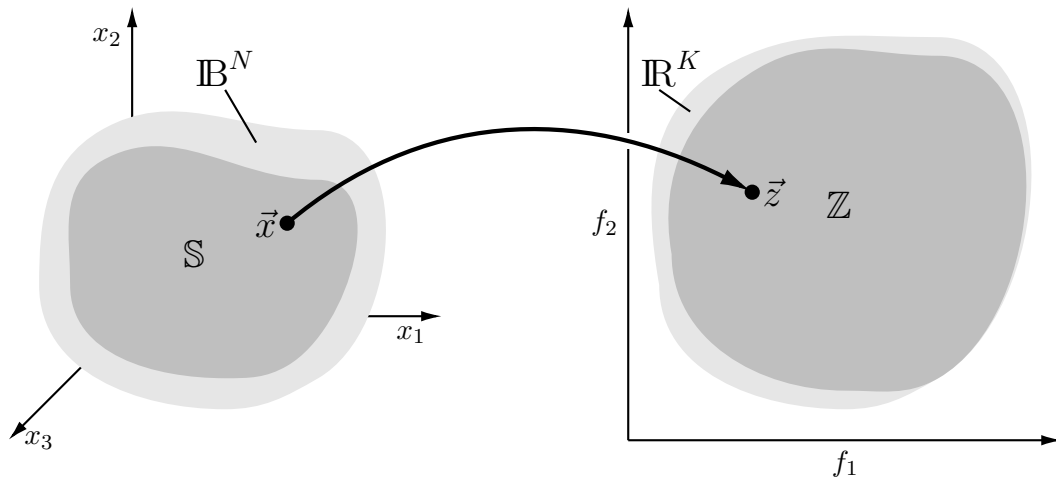
The simple stopping criterion for evolutionary algorithms is to just stop after a predefined number of iterations. Apart from this, there are no real stopping criteria found to be practical [33]. The definition of a residuum is impossible for evolutionary algorithms, as no steady convergence is observable. For multi-objective optimization the definition of a sophisticated abortion criterion is even more complicated, as multiple objectives are to be taken account for. In the scope of this thesis the maximum number of generation was set proportional to the population size. For more than two objectives the number of iterations had to be increased even further. A rule of thumb was developed for an appropriate setting of the population size and the maximum number of iterations given in Equation 3.3 with  $\mu \approx N$ .

$$\mu \cdot t_{max} \geq (K - 1) \cdot N^2 \quad (3.3)$$

Henceforth, if not stated otherwise, the above values are used. In order to avoid problems arising from too short optimization duration it is possible to continue the optimization after its abortion, if required.

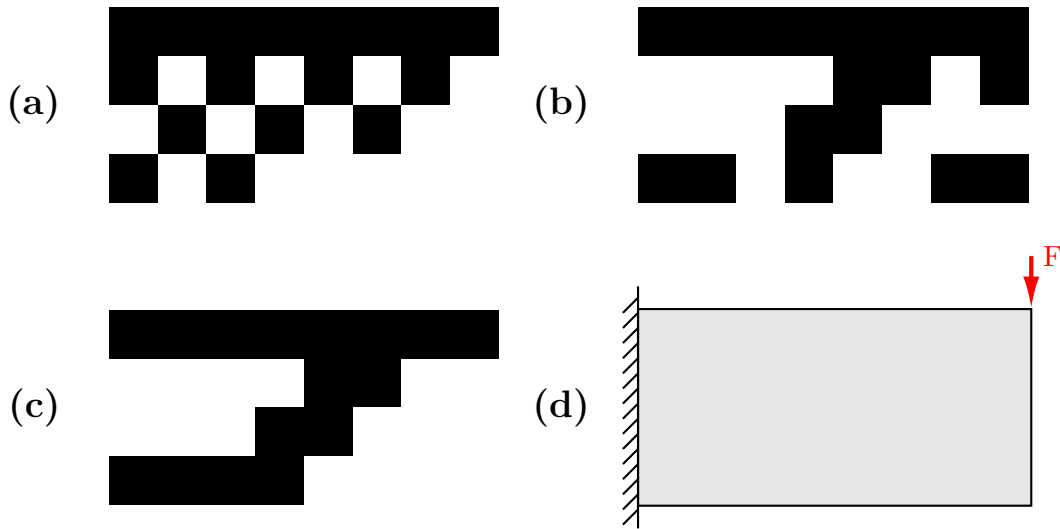
### 3.3 Genotype-Phenotype Mapping

In order to avoid unfeasible solutions, e. g. due to the checkerboard effect (see Section 2.2.3) or loose material floating in space, an additional step can be introduced that maps the binary variable bit string to the finite element mesh used for evaluating the objective functions. Mathematically this is another geometrical constraint to the optimization resulting in further reduction of the parameter space  $\mathbb{B}^N$  to the subspace of feasible solutions  $\mathbb{S}$  (see also Equation 2.7). The projection of a member  $\vec{x} \in \mathbb{S}$  to the feasible objective space  $\mathbb{Z}$  is shown in Figure 3.3. As  $\mathbb{Z}$  itself is again a subspace of  $\mathbb{R}^K$ , the mapping can affect the Pareto-front negatively (also shown in Figure 3.3).



**Figure 3.3:** Projection from  $\mathbb{S}$  to  $\mathbb{Z}$ .  $\mathbb{S}$  and  $\mathbb{Z}$  are subspaces of  $\mathbb{B}^N$  and  $\mathbb{R}^K$ , respectively. A point  $\vec{x} \in \mathbb{S}$  is projected to a point  $\vec{z} \in \mathbb{Z}$  in objective space. Based on [4].

The necessity for such a mapping is depicted in Figure 3.4. Three different binary solutions (a)-(c) to a simple design problem (d) on a coarse grid are shown. Even though solutions (a) and (b) might perform well in objective space, they are not desirable results for an engineer. Solution (a) shows a checkerboard pattern and (b) shows random geometrical features and material floating in space. Both solutions may contain interesting geometrical features but their position in objective space probably does not concur with the performance of a potentially derived design. A result that might be more desirable is solution (c). It may perform worse in objective space than (a) but it may also have more potential to provide insights about an optimal structure. Solution (b) performs way worse in objective space than (c), however, it contains interesting geometrical features. Solutions (a), (b) and (c) belong to  $\mathbb{B}^N$  but only  $\vec{x}_{(c)} \in \mathbb{S}$ .



**Figure 3.4:** Three solutions (a)-(c) for a design problem (d). Mass and compliance are to be minimized. Domination order in  $\mathbb{R}^K$  is  $(a) \prec (c) \prec (b)$ .

There are two work-arounds for the presented complications: Instead of simply discarding (b), its mass and compliance could be significantly improved by small changes in parameter space and instead of accepting (a) as an optimal solution one might prefer solution (c). Even though this means worse results in objective space, solution (c) is more near-net shape than the other two and may therefore be better suited as a design recommendation. This means that from an engineering point of view both, parameter and objective space, can not be viewed independently when evaluating multi-objective optimization results. Sometimes worse solutions in objective space are to be preferred due to their superiority in parameter space.

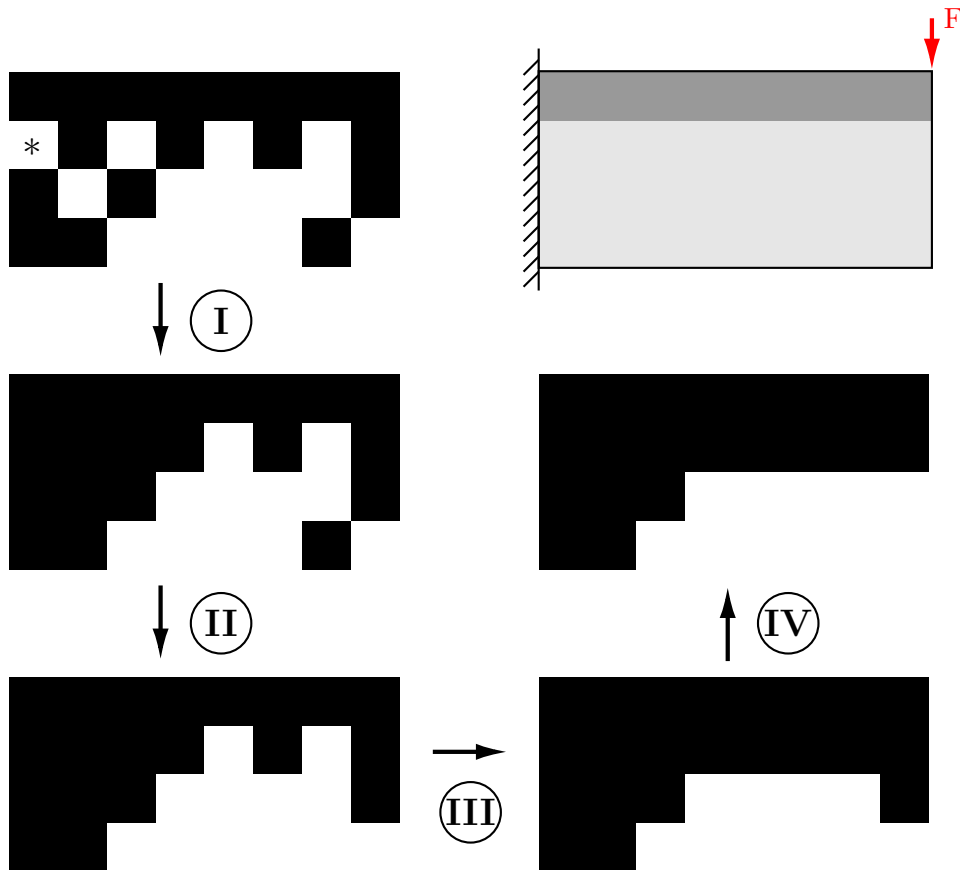
The preceding considerations suggest to avoid infeasible solutions. In order to improve performance of the evolutionary algorithm one might want to not discard the solutions but to improve them. By the proposed genotype-phenotype mapping several infeasible parameter combinations found by the optimizer are mapped to a single feasible mesh. Another approach to improve performance is the ground element filtering technique (GEF, also referred to as approximate density distribution [44–46]) used to reduce the number of parameters and therefore computational effort. The used problem-specific genotype-phenotype mapping is presented in detail in the following sections.

### 3.3.1 Feasibility Filtering

Preliminary tests with binary representation of element densities and nature-inspired meta-heuristics showed that found solutions tend to seem noisy and to have random features. This is due to the extensive use of probabilistic methods (see Section 2.2.1). Other than the checkerboard effect, these do not show up in gradient-based optimization approaches. Their influence on the optimization has been discussed in the prior section. A way to overcome these detrimental effects shall be described here.

The binary representation of the element densities allows to easily classify any parameter as either a solid or a void element. The approach chosen tries to smooth the surface of the structure by step-wise filling holes and removing unconnected solid elements. In order to be able to determine an element's role inside the structure, it is necessary to obtain information about its surrounding. For every element inside the mesh a list of its neighbor elements (henceforward simply referred to as its neighbors) is created in an initial step. For this purpose, it is analyzed with how many elements the element shares a surface. Using hexahedral elements, it is possible to gain information about the dimensionality of the structure by determining the maximum number of neighbors. If all elements have no more than four neighbors the structure is of a two dimensional shape. If elements with up to six neighbors exist, the design space is three dimensional.

The multi step procedure is shown in Figure 3.5 for an arbitrary beam-like structure. In the first step, in order to avoid checkerboarding, all void elements that are surrounded on all sides by solid elements are filled. Therefore, the number of solid neighbors is



**Figure 3.5:** Four steps of the filtering. Top right: Exemplary two dimensional design problem with fixed elements in a darker shade. Top left: The parameter set initially found by binary selection, recombination and mutation operations. I: Holes are filled. The asterisk marks a boundary condition exception. II: Unconnected solid elements are removed. III: Elements surrounded by 3 solid elements are filled. IV: Solid elements that are only connected on one side are removed. A fifth step is used for three dimensional problems.

compared to the maximum number of neighbors counted earlier (4 for two dimensional case, 6 for three dimensional, when using hexahedral elements). An exception is made for elements to which boundary conditions were applied. Any surface with boundary conditions is treated as another solid neighbor element. This ensures connection of the structure to boundary conditions. The following steps are similar to the first one. In the second step all solid elements that are not connected to another solid element over at least one surface are removed. Here, also surfaces with boundary conditions are taken into account. The third and fourth step fill holes that are surrounded by solid material on all but one side and remove solid elements that are connected to other elements over one side only, respectively. A fifth step fills void elements that are surrounded by four solid ones only if the structure was found to be three dimensional.

The general procedure is depicted in pseudo-code in Algorithm 3.1. The operation `count_solid_neighbors` includes the exception made for surfaces with boundary conditions. In order to avoid dependency of the element numbering, remove or fill operations are carried out on a copy of the input structure, that only replaces the actual input structure after every step. After steps that remove solid elements fixed elements are restored. As the filtering described in this section leads to structures that seem to be more feasible design suggestions for an engineer, the filtering will henceforth be referred to as feasibility filtering.

### 3.3.2 Removal of Isolated Material

In the present instance the optimization aims at finding a set of suggestions to a certain engineering design problem. In topology optimization one might want to receive a single continuous body without additional parts of unconnected material floating in space. This problem is unique to the binary representation of the element density. In gradient-based real-numbered optimization approaches such behavior usually does not occur. Even though the isolated material does not necessarily negatively influence the results of the finite element analysis, it leads to overestimation of the relative mass, which may result in underrating the fitness of a member. Thus, the main structure should be kept, while all other clusters of solid elements are to be removed. The approach described in this section removes islands of material that are not connected to any fixed parameter. Hence, at least one element density should be fixed to make use of this procedure.

Connections of solid elements to fixed domains are determined using a maze solving strategy similar to TRÉMAUX's algorithm [47]. The basic idea is that from any element inside the main structure there exists a continuous path through solid elements to an element flagged `fixed`. If such a path does not exist, the elements are not connected to the main structure and can therefore be removed. The deterministic algorithm presented here will try all possible links. If it eventually reaches a fixed element all elements that were part of the path will be flagged as `keep` and will therefore not be removed. If all possible paths were tried and no fixed element was encountered, all elements in the

**Algorithm 3.1** Feasibility filtering algorithm

---

```

begin
  if then
    | max_neighbors := 4
  else
    | max_neighbors := 6
  end
  for every void element do
    count_solid_neighbors
    if solid_neighbors == max_neighbors then
      | element := solid
    end
  end
  for every solid element do
    count_solid_neighbors
    if solid_neighbors == 0 then
      | element := void
    end
  end
  end
  set_fixed_elements_to_solid
  for every void element do
    count_solid_neighbors
    if solid_neighbors == max_neighbors - 1 then
      | element := solid
    end
  end
  end
  for every solid element do
    count_solid_neighbors
    if solid_neighbors == 1 then
      | element := void
    end
  end
  end
  set_fixed_elements_to_solid
  if 3D then
    for every void element do
      count_solid_neighbors
      if solid_neighbors == max_neighbors - 2 then
        | element := solid
      end
    end
  end
end
end

```

---

path will be flagged **delete**. The algorithm starts at the lowest numbered solid element that is not fixed. After the first path is flagged either **keep** or **delete**, another starting point is to be determined. Again, the lowest numbered solid element without any flag is chosen and the procedure is repeated until either an element with a **fixed** or **keep** flag is encountered or a **delete** flag has to be set. The procedure is repeated until no

**Algorithm 3.2** Algorithm used to remove isolated material

---

```

begin
  for every solid element do
    evaluate_number_of_solid_neighbors
    set_keep_flag_if_fixed
  end
  for every solid element without any flag do
    starting_position := element
    path := {}
    add_current_position_to_path
    while no break flag is set do
      counter_at_current_position := counter_at_current_position + 1
      find_neighbor_with_lowest_counter
      if Nowhere left to go then
        all_members_of_path := delete
        set_break_flag
      else
        move_to_next_element
        add_new_position_to_path
        if A neighbor has keep flag then
          all_members_of_path := keep
          set_break_flag
        end
      end
    end
  end
end
remove_elements_flagged_delete
end

```

---

suitable starting points can be found and all solid elements are flagged, respectively. Subsequently, all parameters with a **delete** flag are set to 0 and therefore removed from the structure.

Prior to the start of the process for every solid element the number of its solid neighbors has to be evaluated. A list of every element's neighbors has already been generated for the feasibility filtering. Every element is assigned a counter initialized to be 0. The paths are found by step-wise moving the current position from an element to one of its neighbors. Every time the current position is changing to another element, that element is added to a set **path** and its counter is raised by 1. From all neighboring elements the current position moves to the element that has the lowest counter. If multiple elements share the lowest counter, the one with the lowest numbering is chosen. It is only possible to move to elements whose counter is lower than its number of solid neighbors. If no neighbor fulfills this requirement anymore, all paths from that starting point were tried and every element in the **path** set receives a **delete** flag. The basic procedure is depicted in Algorithm 3.2.

With a rising number of elements the effectivity of this step increases. For full functionality it may be best to assign fixed elements to surfaces of the design domain at which

boundary conditions are applied. That way it is made sure that the main structure is always connected to those areas.

### 3.3.3 Ground Element Filtering

Ground element filtering (GEF, also known as approximate density distribution) is a simple numerical technique exploiting surface spline interpolation for projecting density distributions from a coarsely discretized mesh (henceforth referred to as parameter mesh) to another mesh of higher resolution (henceforth referred to as finite element mesh, as it is used for the finite element analysis) [46]. The technique can be used to avoid checkerboarding effects and to reduce the number of parameters that are to be optimized. Even though the projection of a structure to a finer mesh does not improve the resolution of the structure, it does improve the results of the finite element analysis used as objective functions as it refines the spatial discretization. Also structures of elements only connected over their edge nodes are avoided, which is especially necessary for thermal calculations. Finer meshes consist of a greater number of finite elements and therefore raise the computational effort. The following description of the GEF process for binary density distributions is based on [44–46].

In a first step, a real-valued density distribution  $\vec{\rho}_{real}$  on the finer mesh can be computed from the binary density distribution of the coarse mesh  $\vec{\rho}_{GEF}$  using the relations given in Equation 3.4. The parameter mesh consists of  $m$  elements and the finer mesh consists of  $n$ , respectively.

$$\vec{\rho}_{real} = \vec{C}\vec{A}^{-1}\vec{\rho}_{GEF} = \vec{T}\vec{\rho}_{GEF} \quad (3.4a)$$

$$\vec{A} = [a_{ij}]_{m \times m} = [d(\vec{r}_i, \vec{r}_j)] \quad (3.4b)$$

$$\vec{C} = [c_{kj}]_{n \times m} = [d(\vec{r}_k^*, \vec{r}_j)] \quad (3.4c)$$

$$d(\vec{r}_i, \vec{r}_j) = \sqrt{(\vec{r}_i - \vec{r}_j)^T(\vec{r}_i - \vec{r}_j)} \quad (3.4d)$$

The function  $d(\vec{r}_i, \vec{r}_j)$  returns the euclidean distance between two element's volume centroids  $\vec{r}$ . Centroids of elements of the finer mesh are indicated by an asterisk. In order to again obtain a binary density distribution  $\vec{\rho}$ , the transformation given in Equation 3.5 is used with  $\varepsilon \in [0, 0.5]$ .

$$\vec{\rho}_i = \begin{cases} 0 & \text{for } \rho_{real} + \varepsilon \leq 0.5 \\ 1 & \text{for } \rho_{real} + \varepsilon > 0.5 \end{cases} \quad (3.5)$$

An example of ground element filtering is given in Figure 3.6. The real-value density distribution and two meshes with a 0-1 distribution are shown for an already filtered structure that was found during a run of an evolutionary algorithm.  $\varepsilon$  has a significant influence on the derived finite element mesh. Higher values of  $\varepsilon$  lead to a bulkier structure. Henceforth, the value of  $\varepsilon$  will be 0.1, if feasibility filtering is used, and 0.35, if no further filtering is used. This compensates for the fact that the feasibility filtering



itself leads to bulkier structures as it forbids elements that are only connected to another element over their edges-nodes.



**Figure 3.6:** Ground element filtering. Top left: Initial binary parameters on a coarse mesh. Top right: Corresponding real-values density distribution. Bottom: Two derived binary finite element meshes for  $\varepsilon = 0.05$  and  $0.35$ , respectively.

Ground element filtering is straight forward to implement and universally applicable together with all types of finite elements and any number of spatial dimensions. It can be useful to suppress checkerboarding and to reduce the number of parameters in topology optimization. Nevertheless, it has two disadvantages: First there is no a priori way to determine an adequate value of  $\varepsilon$ . As this value can have a strong influence on the final structure [46], various preliminary tests may have to be conducted. This circumstance is tackled by the prior application of feasibility filtering. Second, as depicted in Equation 3.4, it is necessary to invert an  $m \times m$  matrix. Matrix inversion is very computationally extensive due to high algorithmic complexity of the underlying numerics. In the scope of this thesis the size of the matrices never rose to a point where the inversion might have had noticeable influence on the overall runtime of the optimizer.

### 3.3.4 Filter Combination

The feasibility filtering, the removal of isolated material, and the GEF together make up the problem specific genotype-phenotype mapping used in this thesis. The application of the mapping works as a strict geometric constraint to the optimization. In Algorithm 3.3 the combination of the three steps is shown. The genotype-phenotype mapping method takes a set of parameters as input, processes it into a feasible structure and returns the derived binary element density distribution for the finite element mesh. Original parameters, structure, and mesh are saved for every generated member.

**Algorithm 3.3** Genotype-phenotype mapping

---

```

begin
  structure := input_parameters
  while structure changes do
    | structure := filter(structure)
  end
  remove_isolated_material
  mesh := gef(structure)
  while mesh changes do
    | mesh := filter(mesh)
  end
  return mesh
end

```

---

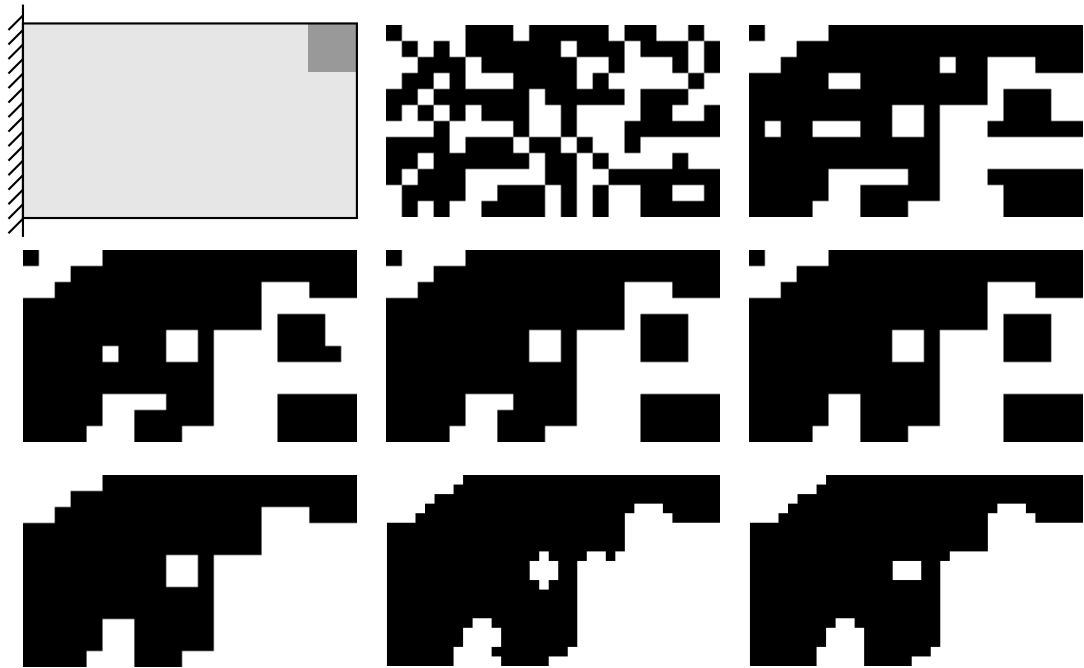
The feasibility filter described in Section 3.3.1 is applied to the structure not only once but several times. As shown in Algorithm 3.3, the filtering process is repeated until the structure is no longer influenced by the filtering. This allows for the selection and recombination operators of an evolutionary algorithm to use the structure rather than the original parameter set and suppresses long outgrowths of material. Additional multi-stage feasibility filtering is applied to the mesh in order to avoid artifacts from non-optimal GEF parameters. All stages of a genotype-phenotype mapping are shown in Figure 3.7 applied to a random parameter set. Feasibility filtering is repeated four times until further application would not have any more influence.

The features feasibility filtering and removal of isolated material can be bypassed in order to evaluate their influence (see Section 5.2). Henceforth, if not stated otherwise, both are executed during optimization. GEF is always used but could also be bypassed by using the same input files for the parameter structure and for the finite element mesh. As the removal of isolated material is working only together with a set of fixed elements most problems contain at least one fixed element. It is especially reasonable to fix elements to which boundary conditions are applied or at which objective functions are evaluated.

By the presented genotype-phenotype mapping, the complexity of the optimization problem is reduced from approximately  $\mathcal{O}(N^4)$  to approximately  $\mathcal{O}(N^2 \cdot n^2)$ . Here  $N$  is the number of binary parameters in and  $n$  is the number of finite elements to which the parameter mesh is projected by the GEF technique.

### 3.4 C++ Implementation

The features covered in this chapter were implemented in C++ 11 under WINDOWS 10 PRO x64. The code was compiled in VISUAL C++ 2015 using optimization to decrease runtime. Selection and genetic operators, the finite element analysis, and the objective functions run simultaneously using the `std::thread` class in order to reduce computation



**Figure 3.7:** Full genotype-phenotype mapping cycle of a randomly generated binary structure. From top left to bottom right: First image shows the design domain with fixed elements and boundary conditions. Second image shows randomly generated binary parameters. The following four structures show the progress of repeated application of the feasibility filter. Subsequently, the images in the bottom row show that unconnected material is removed and GEF is used to project the structure to a mesh of higher resolution. The final structure is obtained after another application of the feasibility filter.

time. This works efficiently because data races do not occur. This is due to the fact that no shared memory has to be written on during these steps.

The original implementation of NSGA-II outputs all members of the parental population with rank 1. Different from that, in the present case, an external Pareto-set is used that is updated after every generation. All members of the updated parental generation that do not violate any constraints are added to the set and non-Pareto-elements are removed subsequently. This way, it is ensured that the Pareto-front contains all Pareto-optimal solutions found during the optimization even if their number exceeds the population size. Clustering is applied directly to this external Pareto-set. The Pareto-set and the clustered set are implemented using the container class `std::map` in order to be able to easily add and remove solutions and to efficiently search for entries. To solve the equations of the GEF (see Section 3.3.3), the linear algebra header-only template library EIGEN<sup>2</sup> was used.

In order to avoid solutions being found twice, an archive of already found solutions is implemented. Every solution that was at least found once is added to a `std::map`. During the generation of a new solution it is checked, if this structure has already been found. If this is the case, the generation process is repeated until a new unique structure

<sup>2</sup><http://eigen.tuxfamily.org/>

is found. This is efficiently possible due to the logarithmic time complexity for searching entries inside a sorted list [39].

Objective functions are evaluated by finite element analysis. For this purpose, a modified version of the free ware finite element programm z88<sup>3</sup> was provided that reads structural information and information about the boundary conditions in a directory `Z880pt` and prints files containing state variables and the structure's mass in a directory `temp`. The system call to run the finite element analysis is modified so that only the calculations are done and only the output files are written that are explicitly required for the prescribed objectives. These outputs are subsequently read and processed into the objective values. Details about the implementation of the objective functions are given in the next chapter.

The objective functions, their parameters, and the parameters for NSGA-II are read from a control file `NSGAI1.ctrl` in the directory `input` together with a list of fixed elements `fix.txt` and the structural data of the parameter mesh. An exemplary control file is given in the appendix. Objective functions are added to the evaluation when the identifier `OBJECTIVE` is read. Parameters and constraints for this objective can be passed afterwards. A list of implemented objective functions and their parameters is given in the appendix. Not all objective functions are automatically optimized. Because some objective functions can be evaluated and used only as a constraint (e.g. the maximum stress), the number of objectives that are to be optimized has to be defined too. Apart from the objective functions, also material parameters and the parameters of NSGA-II have to be assigned in the control file.

The results of the optimization are output in several files inside the `output` directory. The log file `log.txt` includes all information that was read from the control file as well as information about starting time and duration of the matrix inversion required by GEF. In every generation the starting and end time of the generation, the number of found Pareto-elements, clustered elements, and the maximum rank inside the parental population are output. Similar output is printed to the screen. The log file also includes error messages and information about the overall optimization time. Pareto-solutions and clustered solutions are output in two separate files of the same kind. Every line represents a found structure. The first number is a counter starting at 1. The following real numbers are the values of all evaluated objective functions. After the objective values the binary string that encodes the finite element mesh is added.

In order to halt the optimization and to restart it later, after every generation, the whole parental population is output as well as the archive. These files can be pasted to the `input` directory and will then be used together with the Pareto-set as an initial point for the optimization. An overview over the folder and file structure of the program is given in the appendix.

---

<sup>3</sup><http://en.z88.de/>

## 4 Evaluation Setup

In order to evaluate the presented method and to gain benchmark results, several optimization problems were created. Linear elasticity and stationary thermal conduction are regarded and up to three objectives are considered. In the following section the optimization problems are described. To evaluate the Pareto-optimal solutions found by the optimizer, the found structures are further tested. Pareto-optimal design propositions are to be derived and their performance is to be examined. The methodology used to derive designs from the results of NSGA-II as well as the experimental setup is described in the subsequent section.

### 4.1 Optimization Problems

All models were initially set up and meshed in ABAQUS CAE 6.18<sup>4</sup>. The finite element meshes consist of linear hexahedral elements (element type C3D8 in ABAQUS and element type 1 in z88AURORA, respectively). The models were subsequently imported into z88AURORA, where node and element sets were created, boundary conditions were applied and material data was allocated in order to get the required input file format. Even if two dimensional problems are solved, no actual two dimensional elements were used. All models expand in three dimensions and are meshed with hexahedral elements. A tabular listing of the problems, their parameters and constraints is given in the appendix.

In order to minimize the mass of the structure, the concept of relative mass is used. The definition of the relative mass  $r$  is given in Equation 4.1, where  $n$  denotes the number of finite elements,  $V_i$  is the volume, and  $\rho_i$  is the physical density of the  $i$ -th element.  $V_{DD}$  is the volume of the design domain.

$$r = \frac{\sum_{i=1}^n V_i \rho_i}{V_{DD}} \quad (4.1)$$

Throughout the following sections the design domain will be shown in a light gray color, whereas fixed elements will be colored in a darker shade of gray. In all cases, the term stress refers to a positive scalar that is the VON MISES equivalent stress. The optimization problems under purely mechanical loading are typically about minimization of mass and compliance. Compliance can be minimized either via minimization of certain absolute nodal displacement values or by maximizing the first eigenfrequency of a struc-

---

<sup>4</sup><https://www.3ds.com/products-services/simulia/products/abaqus/>

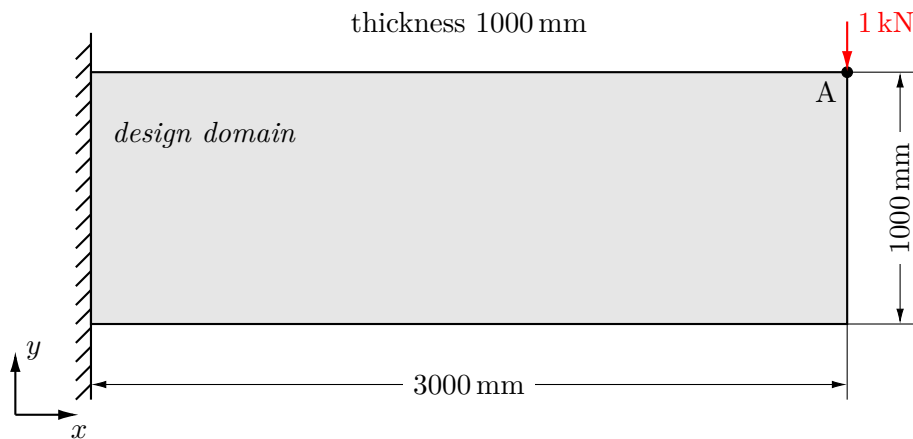
ture [7]. The SIMP approach usually maximizes the stiffness by minimizing the elastic energy  $e$  that is the sum over the elastic energy of all  $N$  elements. Its definition is given in Equation 4.2 with  $\vec{u}_i$  being the element's vector of displacement and  $\vec{K}_i^e$  being the element's stiffness matrix [31].

$$e = \frac{1}{2} \sum_{i=1}^N \vec{u}_i^T \vec{K}_i^e \vec{u}_i \quad (4.2)$$

For the mechanical models conventional engineering symbols will be used to depict the boundary conditions. Surfaces to which temperatures are applied are indicated by thick lines. Heat flows are indicated using arrows. The isotropic linear material data consists of the elastic YOUNG's modulus  $E$ , the POISSON's ratio  $\nu$  and the thermal conductivity  $\lambda$ . The maximum tolerable stress  $\sigma_t$  can be considered indirectly by constraining the maximum element VON MISES equivalent stress.

### Problem 1

The first problem resembles a two dimensional beam. It will be used as a benchmark problem to compare the results obtained by the presented method with the findings presented in [7], where the problem is referred to as MOP1 (for multi-objective problem 1). The design domain with the applied boundary conditions is shown in Figure 4.1. Deflections in all spatial dimensions are constraint on the left end of the beam. A force of 1 kN is applied in negative  $y$ -direction at point A. The design domain is subdivided into a parameter mesh of  $18 \times 6 = 108$  elements resulting in 108 binary parameters. The corresponding finite element mesh consists of  $30 \times 10 = 300$  elements. A steel-like material is used with  $E = 200$  GPa,  $\sigma_t = 200$  MPa and  $\nu = 0.3$ .

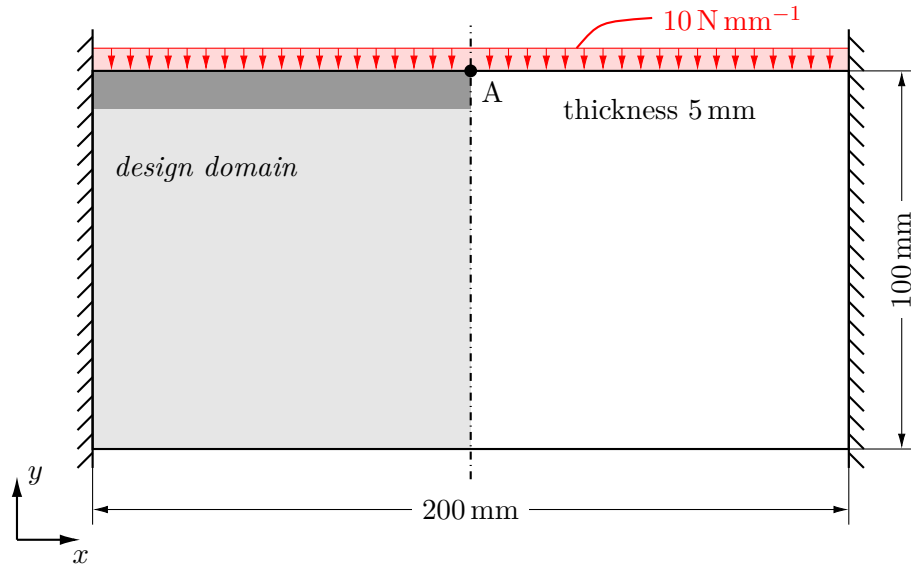


**Figure 4.1:** Beam design domain of problem 1. All displacements are constrained on left hand side and loaded with a point load of 1 kN on right upper edge. Adapted from [7] where it is referred to as MOP1.

The design problem is set to minimize both the relative mass  $r$  and the absolute displacement  $u_A$  at the contact point of the force A. The relative mass is constrained to be in the range of 0.5 to 0.8. This resembles the constraints presented in [7].

### Problem 2

The second problem is a two dimensional bridge-like structure loaded with a constant line force of  $10 \text{ N mm}^{-1}$  in negative  $y$ -direction. The structure and the boundary conditions are symmetric to a vertical axis and therefore the design domain only covers half of the structure. The problem dimensions and boundary conditions are shown in Figure 4.2. The symmetric axis is taken account for by constraining displacements in  $x$ -direction. The parameter mesh consists of  $10 \times 10 = 100$  and the finite element mesh consists of  $20 \times 20 = 400$  elements. The upper row of parameter elements is fixed so there is a total of 90 binary parameters. Similar to problem 1, steel-like material data is used.



**Figure 4.2:** Bridge-like design domain of problem 2. The structure and its boundary conditions are symmetric to a vertical axis. A constant line load is applied to the 10 fixed elements at the top.

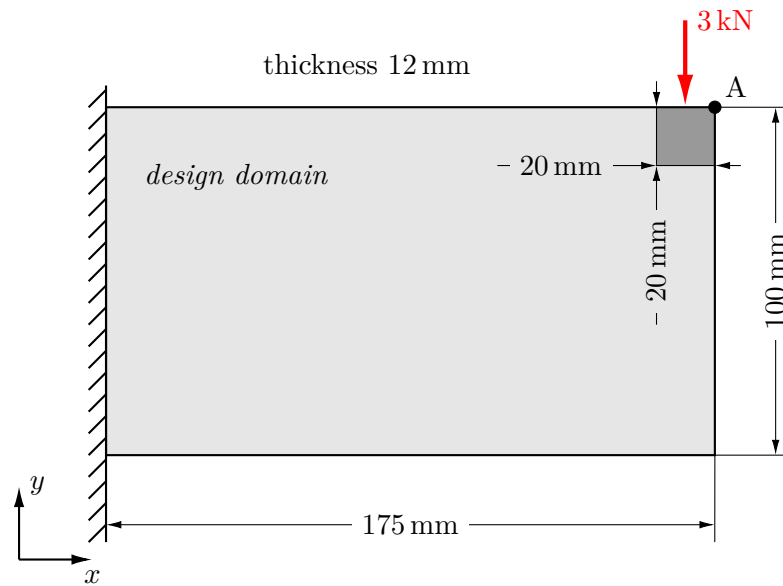
The problem is set to minimize both the relative mass  $r$  and the elastic energy  $e$ . Because the mesh consists of all uniform cubical elements, the definition of the elastic energy of Equation 4.2 was used to derive a simplified objective  $e^*$  to account for a minimization of the elastic energy. For cubical elements and a strict 0-1 distribution the element stiffness matrix can be canceled out as well as the constant factor of a half so that  $e \propto e^*$  applies. The definition of the simplified objective is given in Equation 4.3.

$$e^* = \sum_{i=1}^N \rho_i \vec{u}_i^T \vec{u}_i \quad (4.3)$$

In order to compare the results with solutions found by the SIMP approach, the deflection at point A is also output but not explicitly minimized.

### Problem 3

Problem 3 is a beam-like structure with a mechanical loading similar to problem 1. Its design domain was chosen as a basis for the further physical experiments that are described in Section 4.2. The parameter mesh consists of  $21 \times 14 = 294$  elements and the finite element mesh is made up of  $35 \times 20 = 700$  elements. An area of  $20 \times 20$  mm is fixed and the loading of 3 kN in negative  $y$ -direction is applied to its center. The problem contains a total of 243 binary parameters.



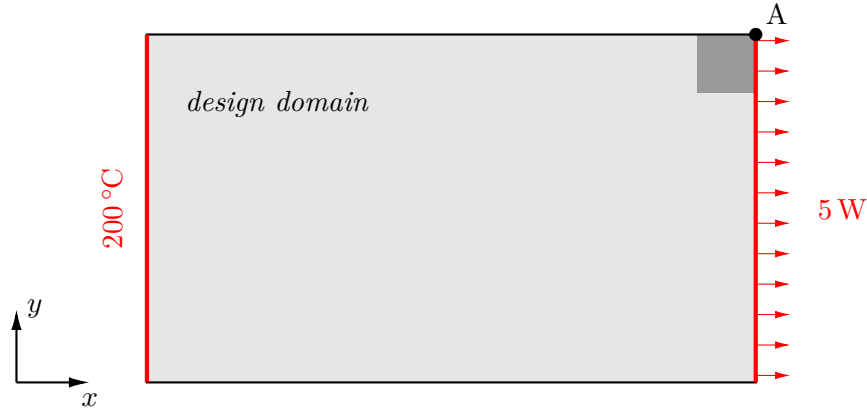
**Figure 4.3:** Beam design domain of problem 3. All displacements are constrained on left hand side and loaded with a point load of 3 kN on Point A. Around the loading there is a set of fixed elements.

Apart from  $r$ , the displacement of point A in  $y$ -direction is to be minimized. The mass is constrained to a range of 0.25 to 0.75. The stress was constrained to a maximum of 200 MPa. An aluminum-like material with  $E = 70$  GPa and  $\nu = 0.27$  was assumed with similar properties to the material used to produce the derived parts.

### Problems 4 and 5

Problems 4 and 5 are both based on the design domain of problem 3 shown in Figure 4.3. The meshes, fixed elements and physical dimensions are kept and thermal boundary conditions are applied. On one side a temperature of 200 °C and on the other side a heat drain of 5 W is applied. A drawing of the design domain with the thermal boundary conditions is given in Figure 4.4. The material is assumed to have a thermal conductivity of  $200 \text{ W m}^{-1} \text{ K}^{-1}$ .





**Figure 4.4:** Thermal design domain of Problems 4 and 5 based on the structure of Problem 3 as shown in Figure 4.3. The left edge of the structure has a constant temperature of  $200\text{ }^{\circ}\text{C}$  and the other edge is connected to a heat drain of  $5\text{ W}$ .

Problem 4 is the optimization with respect to  $r$  and to  $T$  at point A with  $r$  being restricted to be less than 0.7. Taking a nodal temperature as an objective function is connected with a series of complications. If an element is connected only to the heat drain and has no solid connection to the initial temperature, the resulting temperature may be close to the numerical minimum temperature, which has no physical meaning and yet would outperform any other solution by many orders of magnitudes. If this solution also had a very low relative mass, this would lead to a failure of the NSGA-II algorithm because no real solution would ever be able to outperform it. A simple constraint would have no effect because the solution would presumably never vanish from the parental population. Such behavior is avoided simply by discarding any solution that shows an unrealistically low temperature.

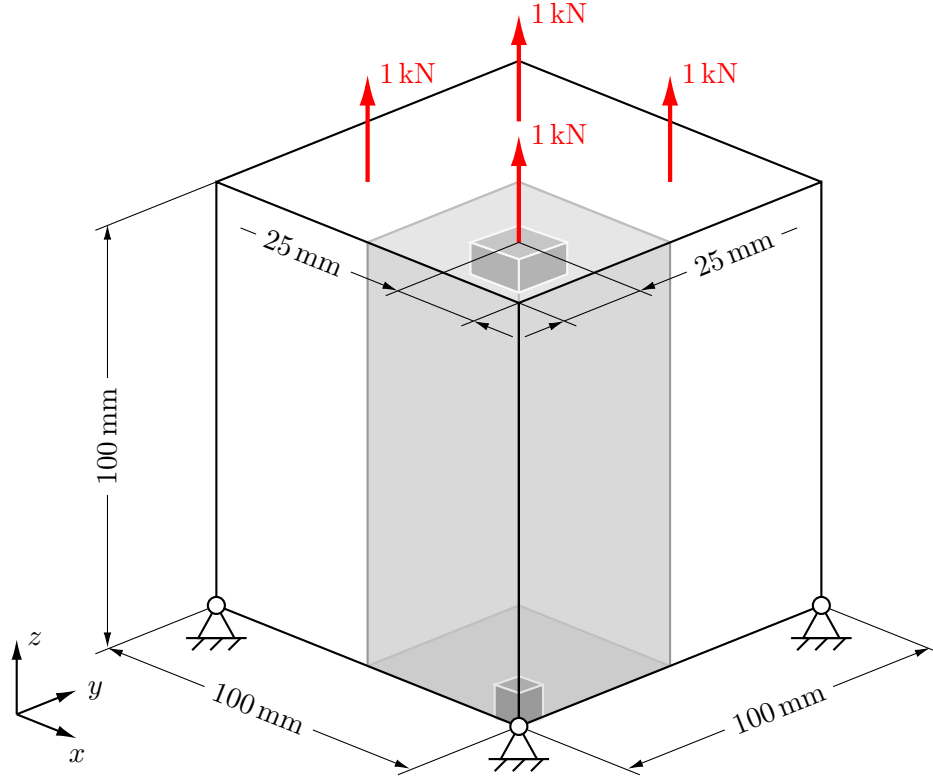
Problem 5 is a superposition of problems 3 and 4. Both mechanical and thermal calculations are done to optimize  $r$  as well as  $T$  and  $u_y$  at point A. This means that three independent objective functions are to be minimized.  $r$  is constrained to be less than 0.4, the nodal displacement  $u_y$  is constrained to be less than 1.0 mm, and the maximum stress is set to 200 MPa.

### Problem 6

Problem 6 is a three dimensional optimization problem that is based on a model presented in [3, 48]. It is of a cubical shape with displacements at the lower vertices being constrained in all directions. A four-fold rotational symmetry is exploited in order to reduce the computational effort and to ensure symmetrical results. The design domain together with the applied boundary conditions is shown in Figure 4.5.

The design domain consists of  $6 \times 6 \times 12 = 432$  elements and the finite element mesh is made up of  $10 \times 10 \times 20 = 2000$  elements. Four point loads in positive  $z$ -direction of 1 kN each are applied to the top plane of the cube. A single element is fixed at the constrained node. A steel-like material is used,  $r$  is constrained to a maximum of 0.8

and the maximum permitted stress is  $\sigma_t = 200$  MPa. The displacement in  $z$ -direction at the loaded node and the relative mass are to be optimized.

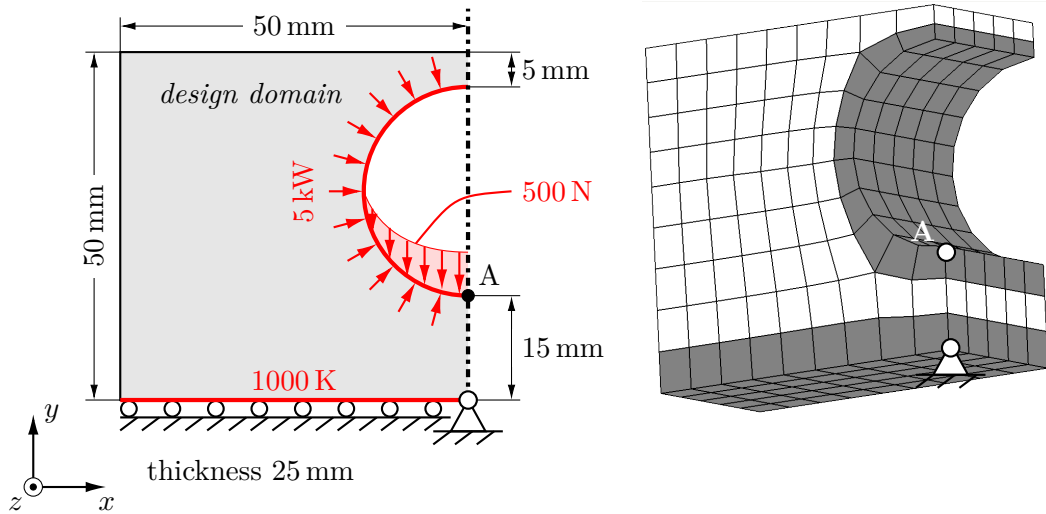


**Figure 4.5:** Cubical design domain of problem 6. The structure and its boundary conditions have a four fold rotational symmetry with respect to the  $z$ -axis. One fourth of the structure makes up the design domain with the element at the support and another four elements around the loading being fixed. Based on [3, 48].

### Problem 7

Problem 7 is the most near application problem tackled in this thesis. It resembles a bearing mount under mechanical and thermal loading. A fitting example would be the mounting of the nose cap of a space shuttle. The design domain is a rectangular block with a hole of 30 mm diameter that is supposed to hold an axle. The axle presses into the bearing with a force of 500 N and works as a heat drain that conducts 5 kW. The contact to the nose cap has a temperature of 1000 K. An axial symmetry is exploited. The design domain is depicted in Figure 4.6. The parameter mesh consists of 395 elements with fixed elements at the bottom and at the hole in order to ensure functionality of the resulting structures. The finite element mesh consists of 1099 elements.

Objectives are the relative mass constrained to be  $\leq 0.3$ , the deflection of point A in  $y$ -direction, and the contact temperature to the axle. This temperature is defined as the maximum temperature of all nodes in contact with the axle. Analogous to the complication described for problem 4 and 5, numerical marginal cases have to be taken care of in order to avoid unrealistic objective values. The displacement is constraint to be



**Figure 4.6:** Bearing mount-like design domain of problem 7. Left: Drawing of the bearing mount-like design domain with mechanical and thermal boundary conditions. The lower surface of the design space is constrained from moving in  $y$ -direction. A single point is fixed in all directions and symmetrical boundary conditions are applied. A cylindrical cavity with a diameter of 30 mm is loaded with a surface force by a shaft. The shaft itself is acting as a heat drain. Right: Parameter mesh with fixed elements emphasized in gray.

less than 0.001 mm. The material data resembles an isotropic fibre reinforced ceramic typically used in advanced aerospace engineering application. The elastic modulus is 60 GPa,  $\nu = 0.3$  and  $\lambda = 15 \text{ W m}^{-1} \text{ K}^{-1}$ . Stresses are constraint not to exceed 80 MPa.

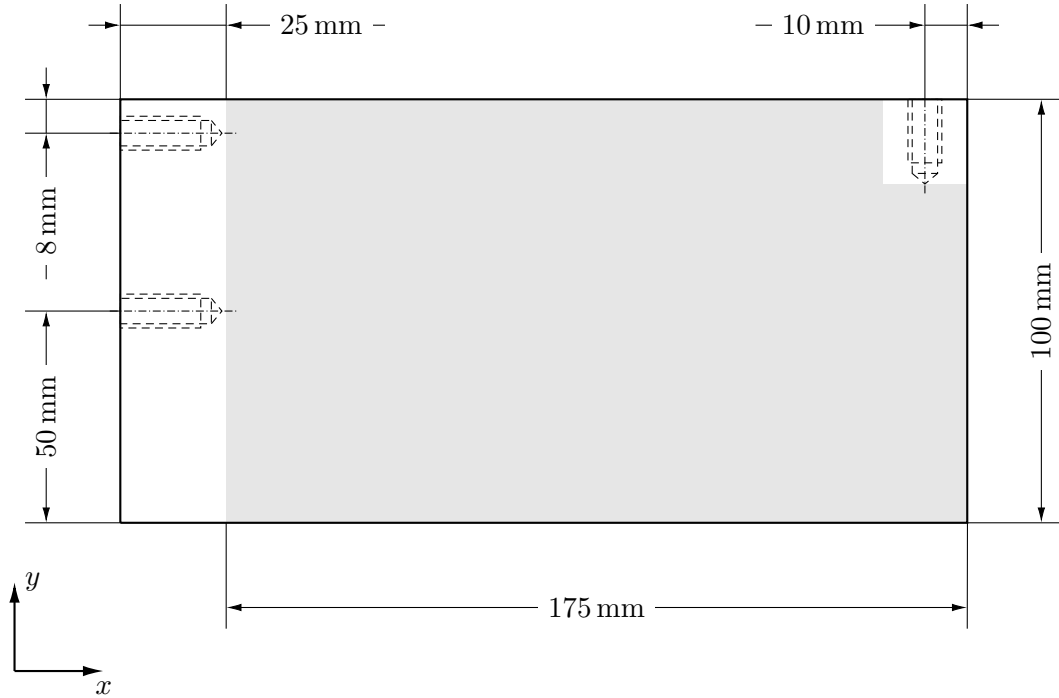
## 4.2 Experimental Setup

In order to find out whether it is possible to obtain Pareto-optimal designs from the presented optimizer, sets of design propositions are to be derived, machined and tested. The parts are to be loaded by a hydraulic press and the displacement in  $y$ -direction at a certain point is to be evaluated by digital image correlation (DIC).

Problem 3 was chosen to be used for the tests. The design domain of Figure 4.3 is used as a basis to design five specimens of different relative masses. A prototypical drawing is given in Figure 4.7. The original design domain is unchanged and shown in gray. The parts are to be mounted to a solid support via two screw joints. The location of the joints was chosen to account for the bearing reaction due to the bending force. The force itself is to be applied via a locator that is screwed onto the part. By that means, it is ensured that the point of contact does not change significantly during loading. All screw threads are ISO M8. A smoothing tool as described in Section 2.2.3 was provided and used to derive part designs. The default parameters were used. The parts were designed in CREO PARAMETRIC 3.0<sup>5</sup> with a minimum inside radius of 5 mm and machined out of 100 mm  $\times$  12 mm aluminum bar stock. A temperable AlCuMgPb

<sup>5</sup><https://www.ptc.com/en/cad/creo/parametric>

alloy was used. A detailed data sheet is given in the appendix. The design process will be further illustrated in Section 5.3.1.

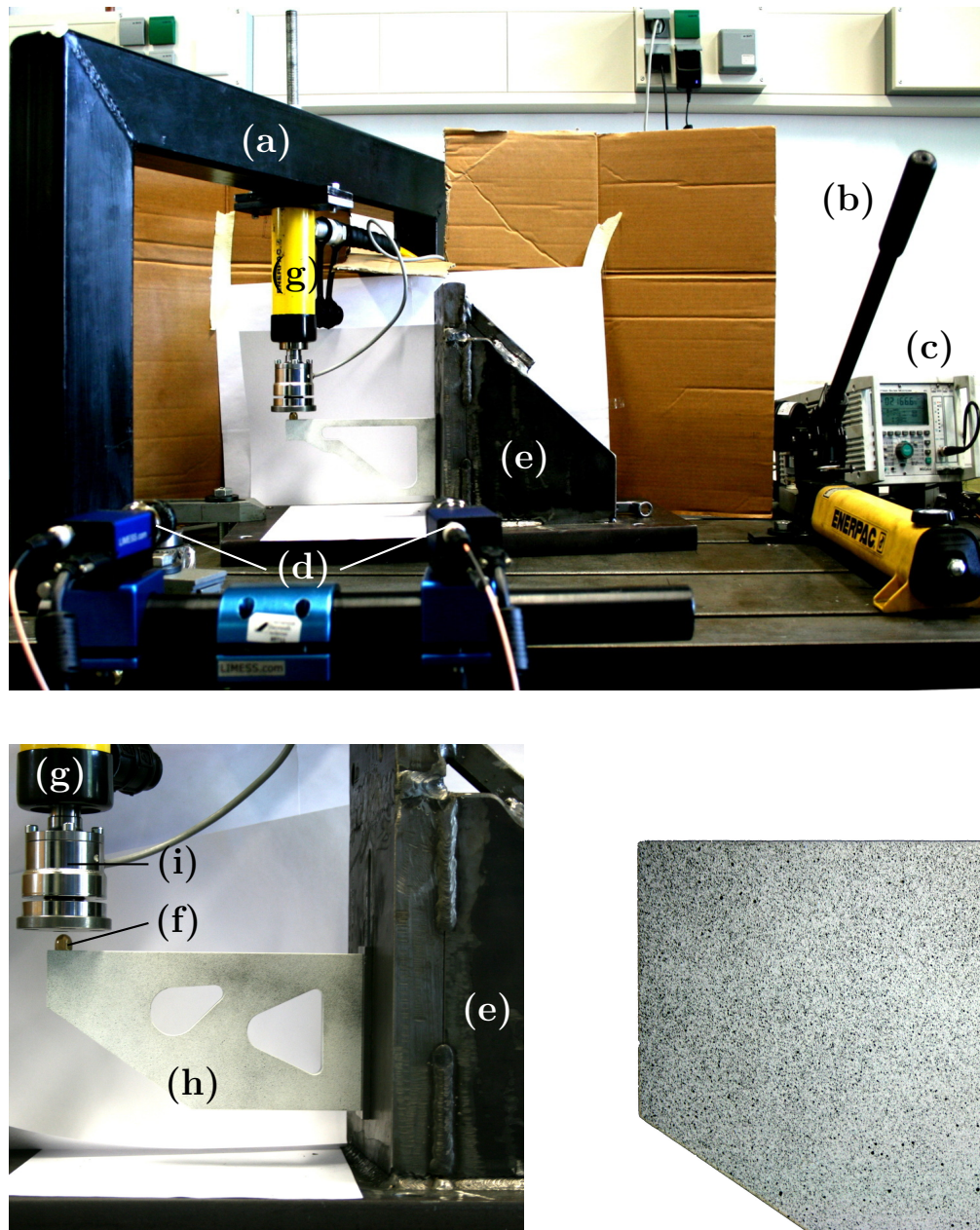


**Figure 4.7:** Drawing of specimen based on design problem 3. The gray area corresponds to the design domain specified in Figure 4.3. All screw threads are ISO M8.

The whole testing rig is shown in Figure 4.8. The hydraulic press stud is mounted to a steel frame (a). The force is applied manually via a lever (b) and a measuring amplifier (c). In the foreground two industrial cameras (d) are shown mounted to a tripod. Both cameras point to the region of interest with an angle of  $5^\circ$  to  $25^\circ$  towards each other. The cameras are connected to each other and to a measuring amplifier. The amplifier is connected to a computer with the DIC software used to calibrate the cameras, take images and to do the image correlation.

To evaluate the performance of the derived part designs, the manufactured parts are to be mounted to a support and loaded with a hydraulic press. Details of the mounting are shown in Figure 4.8. All fixed parts are mounted to a machine bed. The supporting structure (e) is made from welded 30 mm construction steel in order to reduce its contribution to the system compliance. A brass locator (f) transfers the force from the hydraulic press (g) into the specimen (h). A pressure cell (i) records the overall force applied. The specimens are bolted to the mounting support with a torque wrench set to 27 N m and the locator is manually screwed hand-tight into the specimen.

In order for the DIC to obtain a surface displacement field, it maps patterns found on the specimen's undeformed surface to the deformed surface. These patterns have to be applied to the structure manually [49]. The specimens were spray painted white and then speckled with black spray paint. By that means, a fine, even speckle pattern is



**Figure 4.8:** Test rig (top), mounting details (bottom left) and prepared specimen surface (bottom right). (a) supporting frame, (b) hydraulic lever, (c) measuring amplifier, (d) industrial cameras on tripod, (e) mounting support, (f) locator, (g) hydraulic press, (h) specimen, (i) pressure cell.

obtained. This speckle pattern is also shown in Figure 4.8. To achieve even lighting and to avoid the need for further light sources the background is masked with white paper.

The DIC system used was LIMESS Q-400-3D<sup>6</sup> together with the software ISTR 4D X64 4.3.0.48. In order to obtain correct results, the cameras were focused on the region of interest, the aperture was set appropriately and the system was calibrated [49]. Three different camera setups were used, each with different camera angles, distance to the

<sup>6</sup><http://www.limess.com/en/products/digital-image-correlation>

specimen and aperture settings. Exemplary images for each camera setup are given in the appendix. With each setup three image correlations could be carried out. One for each camera and another one in three dimensions. The procedure was repeated twice for every sample including the mounting process in order to compensate for systematical errors. By that means, for every sample 18 displacement fields were obtained and evaluated at the desired point (see Figure 4.3).

Apart from the physical testing, the design propositions were also subject to further simulations. Problem 7 was also basis for further analysis. The smoothed structures as well as two derived designs were imported into Z88AURORA, ABAQUS CAE 6.18 and COMSOL 5.2<sup>7</sup> and meshed. Mechanical and thermal simulations were carried out with meshes of different element sizes and shape functions in order to find a suitable discretization.

---

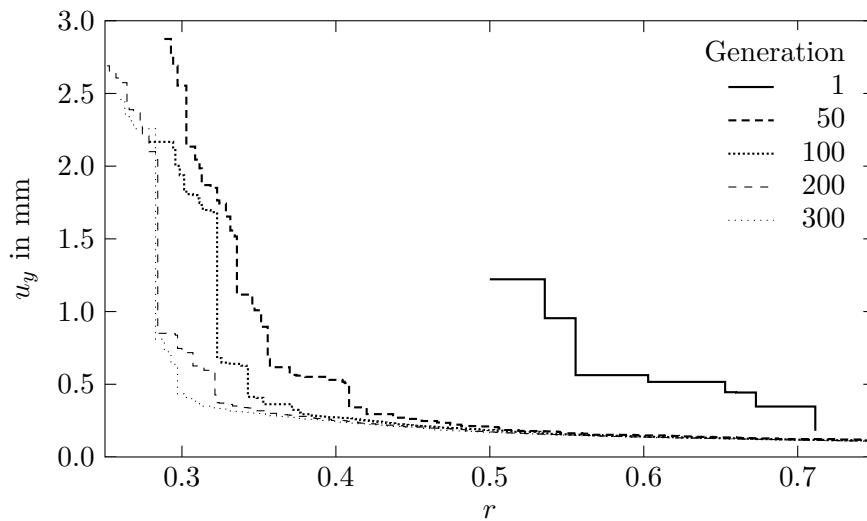
<sup>7</sup><https://www.comsol.com/>

## 5 Results

In this chapter the results of the application of the previously described optimizer will be shown and discussed. In the first section the overall performance of the presented method will be evaluated by exemplary structures and subsequently the influence of the genotype-phenotype mapping will be shown. In the last chapter optimization results for problems of different sizes, number of objectives, and physics will be presented. Both physical experiments and further calculations were conducted in order to verify the optimization results and to evaluate the derived part designs. Optimization results will be compared to other approaches and to results from literature.

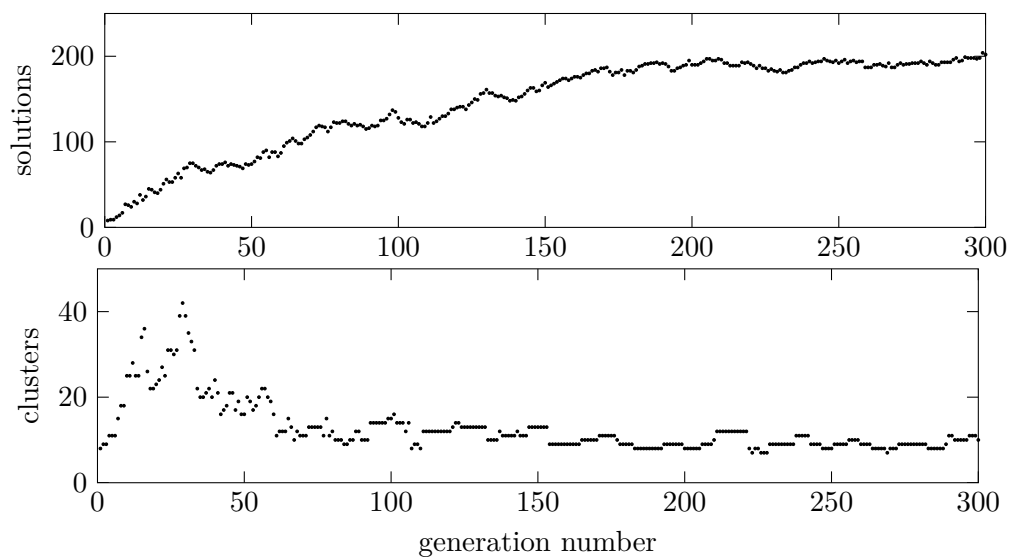
### 5.1 Optimizer Performance

In Figure 5.1 approximated Pareto-fronts for problem 3 are shown at five different generations. They were found using NSGA-II ( $\mu = 300$ ,  $t_{max} = 300$ ). One can see that over the course of the optimization the approximated Pareto-front seems to converge towards a curve in objective space. The term convergence is misleading because no convergence in a mathematical sense does occur [33]. Nevertheless, the relative improvement per generation decreases over the course of the optimization, whereas the fronts steadily advance. Apart from that, one may observe that the coverage in objective space and the number of found solutions also improve over the time. In generation one there are



**Figure 5.1:** Approximated Pareto-fronts for problem 3 at five different generations.

only eight solutions, whereas in the last generation 202 trade-off solutions were found. A greater number of solutions also increases the quality of the approximation of the Pareto-front. The number of found solutions and the number of clustered solutions are plotted over the number of generations in Figure 5.2. The size of the Pareto-set of solutions rises almost steadily over the course of the optimization and eventually seems to settle at a fixed value of around 200. The number of clustered solutions first alternates but over time also seems to level off at a fixed value. This behavior has been found for all multi-objective optimizations carried out and is in good accordance with findings from literature [50]. Multiple runs with the same parameters also showed similar amounts of found solutions and clusters.

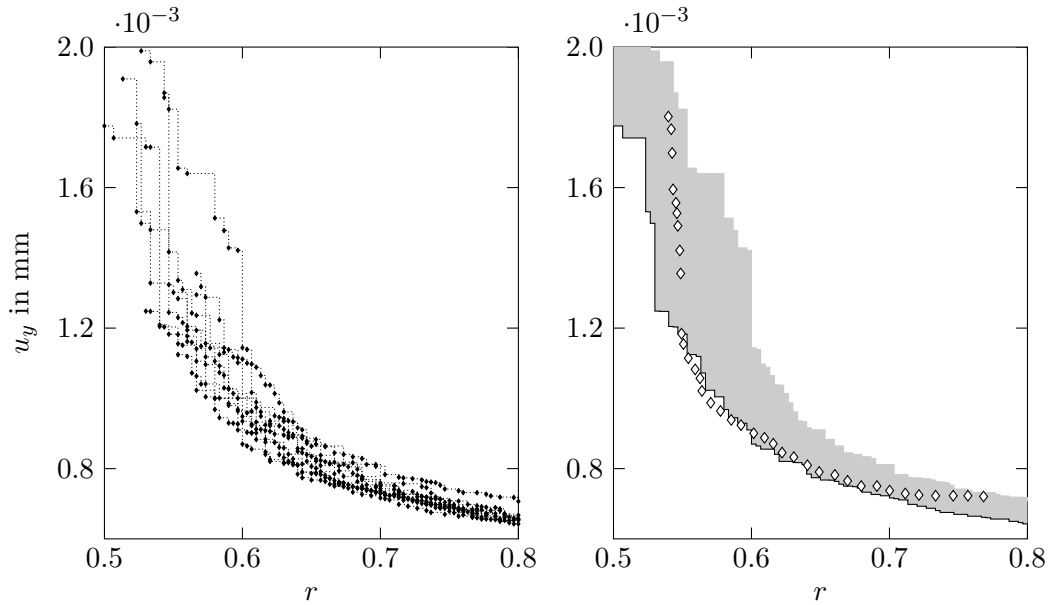


**Figure 5.2:** Number of found solutions per generation. Top: Number of Pareto-optimal solutions found up until the current generation. Bottom: Number of solution clusters.

Performance evaluation of population based optimizers, especially evolutionary algorithms, is not a simple task. The frequent use of probabilistic operators leads to an unpredictable course and statistical variation of the results [20]. In Figure 5.3 the obtained Pareto-fronts of ten independent runs of NSGA-II are shown. Problem 1 was solved applying the same parameters used in [7] ( $\mu = 30$ ,  $t_{max} = 120$ ) without feasibility filtering. It can be seen that above  $r = 0.65$  most solutions seem to cumulate. However, below  $r = 0.65$  the variation is significantly bigger. The displacement varies with up to a factor of two at a fixed relative mass and cumulations of solutions are hard to identify. Interestingly, not a single solution was found during more than one run. It can be concluded that, in order to evaluate the performance of an multi-objective evolutionary algorithm and its features, comparing two single runs can not yield significant insights. A number of runs has to be conducted and variation as well as the best Pareto-front have to be compared.

The region of solutions found for problem 1 is shown in Figure 5.3 together with a Pareto-front taken from [7], which was assembled from non-dominated solutions of



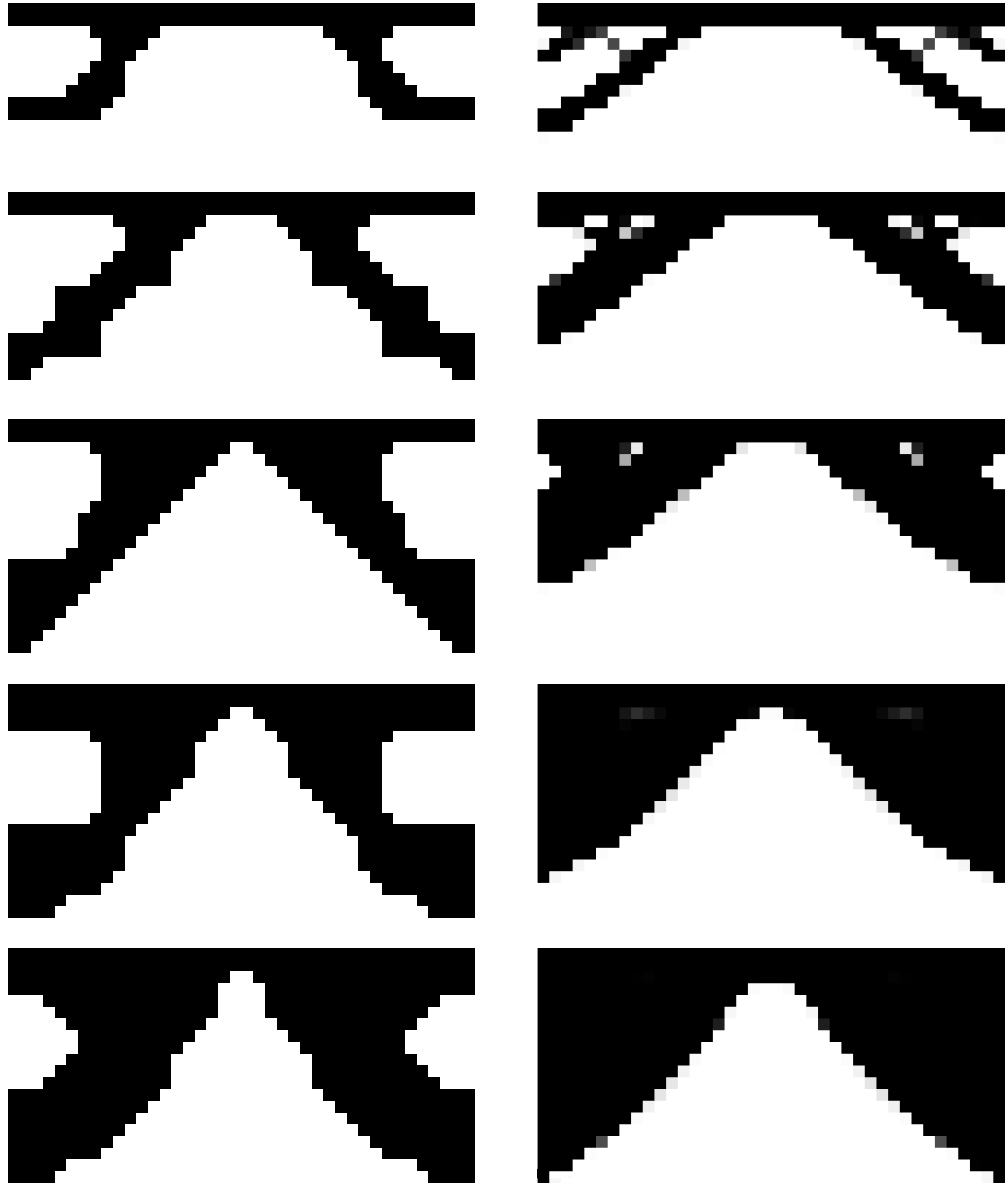


**Figure 5.3:** Variation over ten runs of NSGA-II. Left: Pareto-fronts for problem 1 found by ten optimizer runs with the same parameters without filtering. Right: Big white marks indicate solutions presented in [7]. Gray coloring indicates the area between the overall Pareto-front and the region that was dominated by all found solutions.

ten independent runs. It can be seen that the Pareto-front obtained by the presented optimizer lies very close to the one presented in literature. Slight differences may be results of different finite element implementations or statistical variations as described earlier in this section. More severe deviations may be results of differently implemented constraint handling. All in all, the optimizer presented in this thesis proved to be able to find similar results in the same time as optimizers presented in the literature.

Taking into account the statistical variations and thus comparing overall Pareto-fronts, another difficulty arises. It is the definition of the quality of obtained solutions. When comparing two approximated Pareto-fronts multiple factors may be considered, for instance the number of solutions, even spreading of the solutions inside the front, diversity of the solutions in parameter space (see Section 3.2.3). These measures do not describe how far the Pareto-front advanced. This may be described in terms of how much of front one is dominated by another or the area of objective space not dominated by an approximated front [7, 25]. Considering the above points, tuning an evolutionary algorithm's parameters itself is a multi-objective optimization problem because sometimes many evenly spread solutions may contain more valuable information than a few single solutions even though these might be better in terms of domination (see also Section 3.2.2).

In Figure 5.4 five pairs of structures of equal mass are shown found by NSGA-II ( $\mu = 150$ ,  $t_{max} = 150$ ) for problem 2 together with structures found by the SIMP approach ( $\alpha = 4$ ). The software used was z88ARION, using sensitivity filtering and the default parameters. This optimization was conducted with the same mesh that was used by the

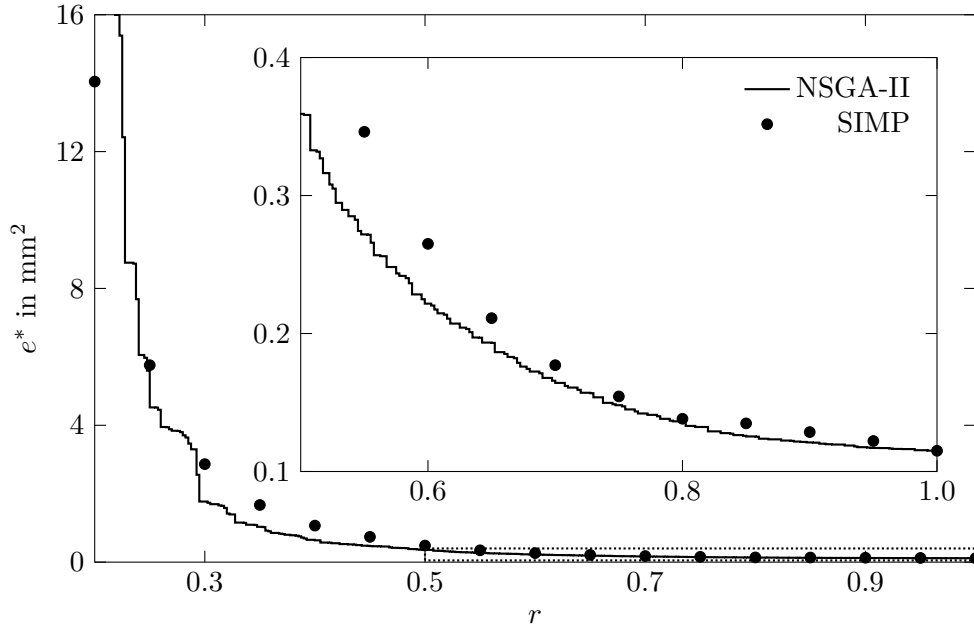


**Figure 5.4:** Five solutions to problem 2. From top to bottom the relative mass increases from 0.2 to 0.6 in steps of 0.1. On the left hand side the results obtained by NSGA-II are shown and on the right the structures found by the SIMP approach are given.

evolutionary algorithm. It can be seen that in the depicted range of  $r$  the found solutions are of fundamentally different shapes. The lightest structure found by z88ARION shows considerably finer geometrical features than its counterpart but also shows areas in which no discrete 0-1 distribution was achieved. The results found by the optimizer based on NSGA-II inherently show a strict 0-1 distribution but also show more coarse features due to the filtering described in Section 3.3. All structures show diagonal supports that tend to get bulkier for greater relative masses. The point of contact with the walls is located significantly lower in structures found by NSGA-II.

In Figure 5.5 the Pareto-front for problem 2 is shown. To achieve a Pareto-set, 19 runs of SIMP were conducted with relative target volumes from 0.1 to 1.0. The simpli-

fied compliance objective  $e^*$  was calculated from the corresponding displacements after Equation 4.3. It can be seen that for  $r > 0.25$  the Pareto-front found by NSGA-II dominates the SIMP solutions. This may be due to the lowered contact points of the support structure. The worse results of NSGA-II in objective space below  $r = 0.25$  may have occurred due to the fact that the SIMP approach found solutions without a 0-1 distribution.



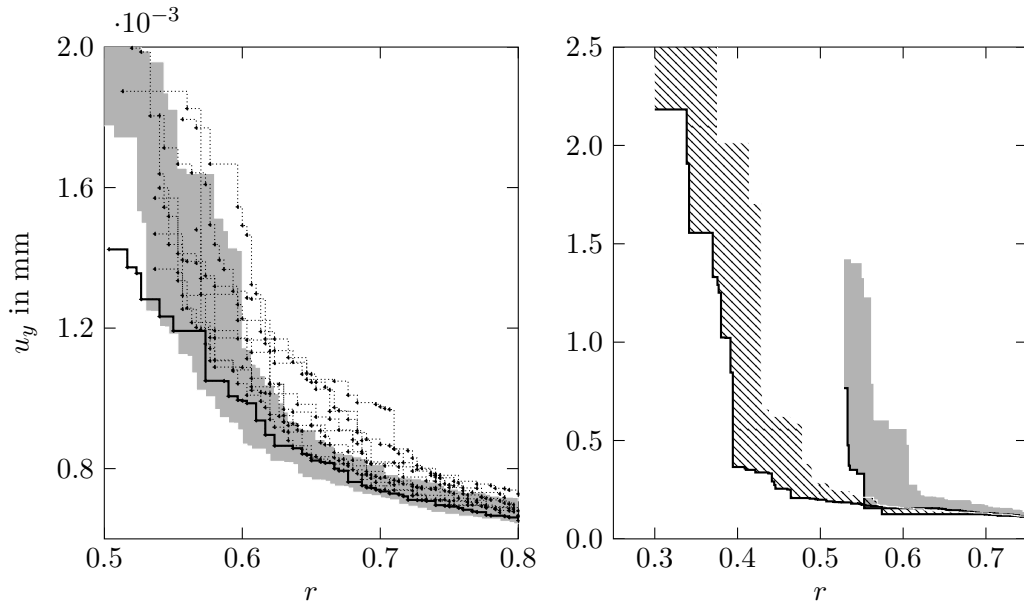
**Figure 5.5:** Comparison between Pareto-fronts found by NSGA-II and an SIMP approach for problem 2. The simplified compliance objective  $e^*$  is plotted against the relative mass.

Findings discussed in this section indicate that the presented optimizer basing on NSGA-II is able to find better solutions than a classical SIMP approach when optimizing structural stiffness. The SIMP optimizer took around 1000 finite element calculations to find 19 solutions. It took NSGA-II 22 500 FEA runs to find 189 non-dominated solutions.

## 5.2 Influence of the Genotype-Phenotype Mapping

The influence on the Pareto-fronts by the genotype-phenotype mapping described in Section 3.3 is shown in Figure 5.6. The Pareto-fronts and the corresponding variations with and without filtering are compared for problem 1 (left) and problem 3 (right). As described in Section 5.1, the variations over ten runs are drawn as areas in order to illustrate the statistical variations by the evolutionary algorithm.

The unfiltered results for problem 1 are the same that are shown in Figure 5.3. Only taking into account the results in objective space, it seems as if the filtering would have had a negative effect on the results. It can be seen that the results with activated filtering show a broader area of variation and, above  $r = 0.54$ , show a less advanced Pareto-front.



**Figure 5.6:** Pareto-fronts influenced by filtering. Variation depicted as areas described in Section 5.1. Left: Pareto-fronts to problem 1. Gray area as shown in Figure 5.3. Dotted lines show ten runs with activated filtering. Solid line shows overall Pareto-front found with filter. Right: Pareto-fronts for ten runs of problem 3. Filtering was again deactivated for the gray area and activated for the striped area.

The latter is probably due to the filtering working as an additional geometrical constraint to the optimization resulting in a reduction of the objective space (see Figure 3.3).

For problem 3, in contrast to the above, a much more positive effect caused by the filtering can be observed. For a relatively low computational effort ( $\mu = 30$ ,  $t_{max} = 120$ ) the optimizer together with the filter was able to find a more advanced Pareto-front (with exception of one single point). Without the use of the filtering no results were found for  $r < 0.55$ . A similar effect can also be observed on the left plot. Below  $r = 0.54$  the filter was able to find significantly better solutions in one instance. The positive effect on problem 3 is probably due to its greater amount of elements compared to problem 1. Thus, the constraint imposed on the problems by the filtering seems disadvantageous for smaller problems. For large enough problems the filter does not only lead to a more advanced Pareto-front but also helps to more quickly spread the Pareto-front over the desired range of relative volume.

The influence of the filter in parameter space is shown in Figure 5.7. Both structures were members of the approximated Pareto-front after 120 generations and both have a relative volume of around 0.68. The filtered structure shows smooth outlines and seems more near-design compared to the structure that was found without filtering. The unfiltered structure seems more bulky than the filtered one, even though it has the same relative mass. This is due to the many small holes that were generated. Also, some material is not connected to the beam, which leads to a wrong position of the solution in objective space. Both of these problems are addressed directly by the filter. It may



**Figure 5.7:** Filtered (left) and unfiltered (right) solution to problem 3 after 120 generations ( $r \approx 0.68$ ).

be possible to generate a smoother structure by a more fitting choice of  $\varepsilon$ , but there is no general method to determine this factor a priori.

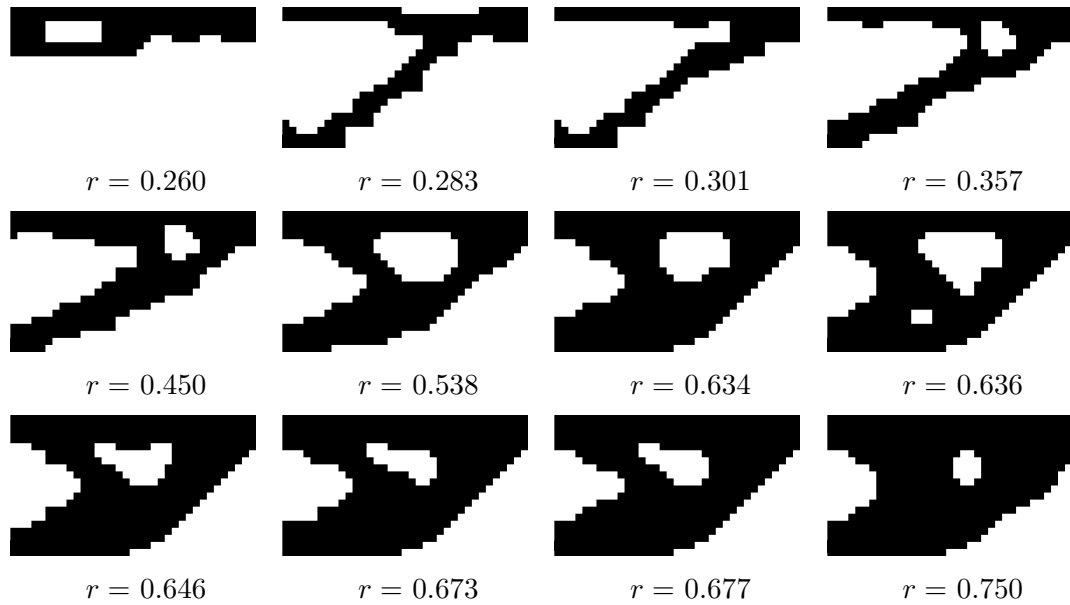
All in all, the findings presented in this section indicate that the genotype-phenotype mapping described in Section 3.3 is significantly advantageous for optimizing larger structures using a binary representation of a structure together with an evolutionary algorithm. For a large amount of binary parameters the filtering process strongly improves the approximation of the Pareto-front. Found solutions not only perform better in objective space but also tend to be more feasible as a basis for further engineering design. Negative effects due to the filtering can be observed for smaller structures. As it is the scope of this thesis to apply multi-objective evolutionary optimization to larger structure, all further optimization runs presented will have made use of the filtering.

### 5.3 Optimization Results

As presented in Section 4.1, the developed optimizer was used to find sets of Pareto-optimal trade-off solutions for different problems. In this section the found structures are presented and their performance in objective space is discussed. Special attention is given to engineering designs derived from the found solutions. First purely mechanical problems will be discussed and subsequently more thermal calculations will also be taken into account.

#### Mechanical Objectives

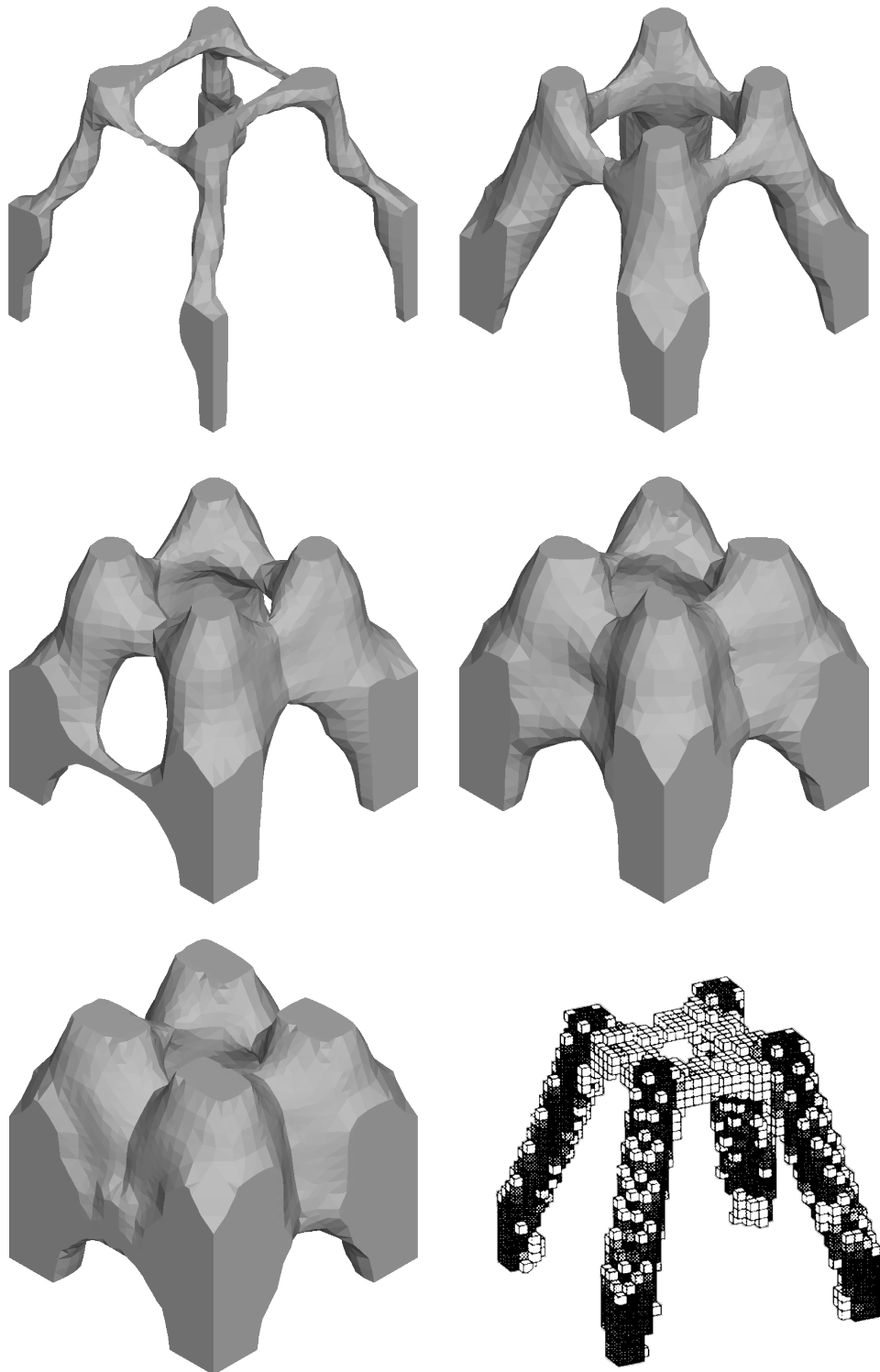
In Figure 5.8 twelve structures are shown found for problem 3. The results were obtained after 300 generations with a population size of 300. The first solution with the least relative mass fundamentally distinguishes from the others because it is basically a horizontal beam with a lengthy hole close to the supporting wall. The next heavier solutions consist of a horizontal beam and a diagonal supporting beam. The fourth and fifth structure show a different topology and an additional support structure. The heaviest solutions are rather similar to each other and only the thickness of the structure seems to increase.



**Figure 5.8:** Twelve solutions to problem 3. From top left to bottom right the mass of the structure increases while the displacement at the upper right corner of the structure decreases.

Problem 6 was the largest structure that was subject to a multi-objective optimization in the scope of this thesis, in terms of computational effort and number of elements that make up the finite element mesh. 446 single results and 8 clusters were found. A run with  $\mu = 450$  over 400 generations took over 49 h on a desktop computer with a four core CPU at 3.2 GHz on WINDOWS 10 PRO x64.

In Figure 5.9 five smoothed results are shown for problem 6. The smoothed triangular surface mesh was mirrored in order to show the whole structure. Additionally, an optimal structure obtained by [48] on a mesh of  $30 \times 30 \times 30$  elements is given. The optimum structure from literature is described as a quadropod solution with solid legs transferring the point loads to the nearest supports and additional material interconnecting the legs [48]. Similar results were obtained by NSGA-II. Four leg-like connections between the fixed elements and interconnecting material between neighboring legs can be observed with the interconnections being in the upper part of the design space. Over the whole permitted range of  $r$ , changes of the structures mostly consist of further thickening of the structure and larger interconnections. Because symmetry with respect to a plane not parallel to a coordinate plane is not implementable, two symmetry planes were implemented instead of the possible four that the design domain holds. Nevertheless, three of the five presented solutions show additional symmetry that was not a priori imposed to the system. The above points might be indicators for solutions close to optimal structures. Still some non-symmetrical geometry can be observed. The lightest structure presented in Figure 5.9 also seems to show a mesh dependent appearance that indicates that the spatial discretization of the density distribution started failing at such low relative masses.



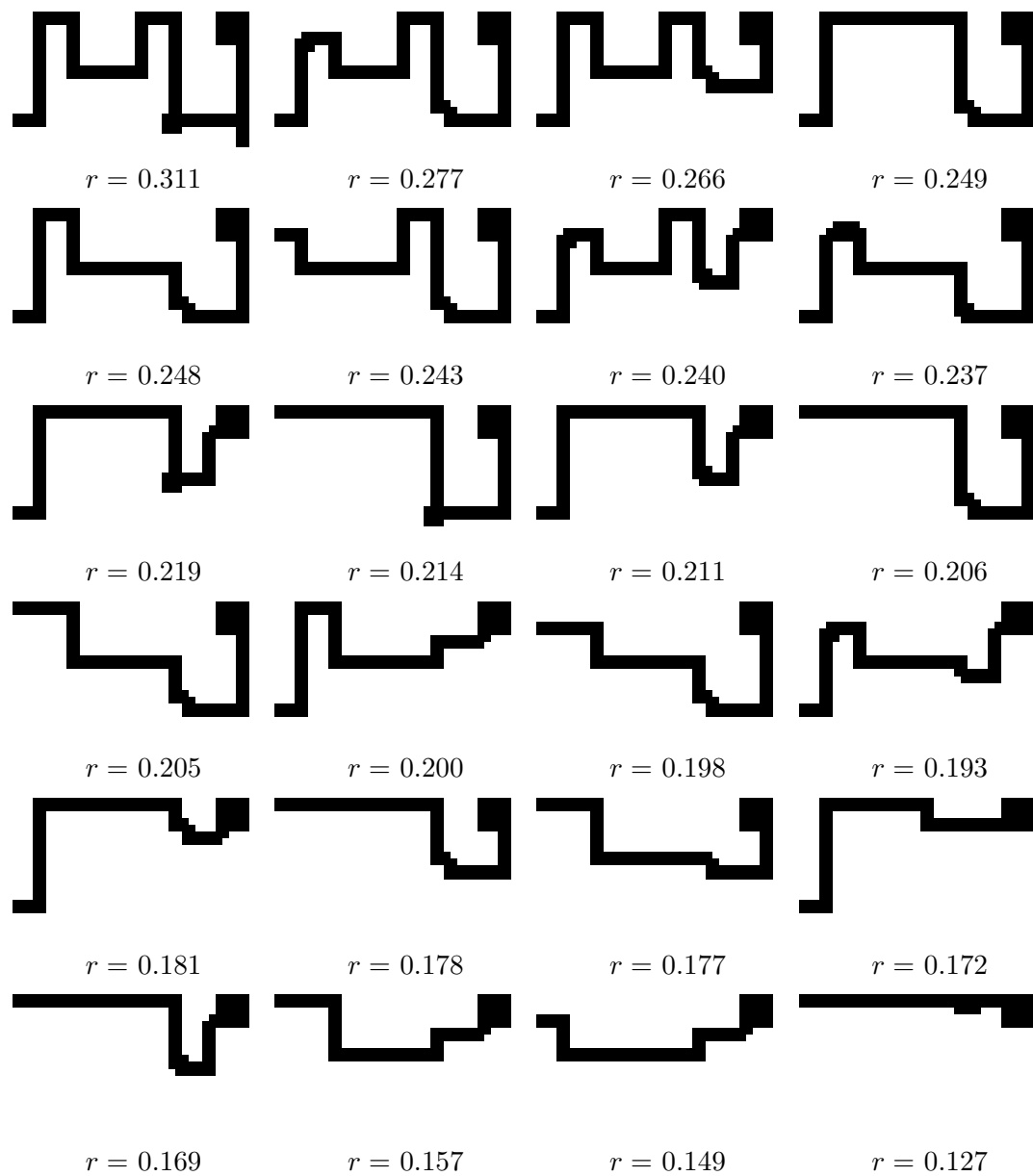
**Figure 5.9:** Five smoothed solutions to problem 6. From top left to bottom the relative mass increases from 0.1 to 0.5. Bottom right shows results presented in [48].

There are clear indicators that show that the results to problem 6 are likely to be very close to an optimal material distribution. When talking about additional unprescribed symmetries and similar shapes, one has to keep in mind the characteristics of the evo-

lutionary optimizer used. Due to the probabilistic generation of the parameters slight unevenness always occurs. This effect is attenuated by the use of a smoothing tool.

### Thermal Objectives

In Figure 5.10 all clustered solutions are shown for problem 4. The overall Pareto-set included 65 trade-off solutions found after 1000 generations and with a population size of 300. The problem formulation can be interpreted as finding the optimal connecting structure between a heat drain and a surface that is to be cooled. Mass as well as the contact temperature to the heat drain are to be minimized.



**Figure 5.10:** 24 solutions to problem 4. From top left to bottom right the temperature at the upper left corner of the structure increases while the mass of the structure decreases.



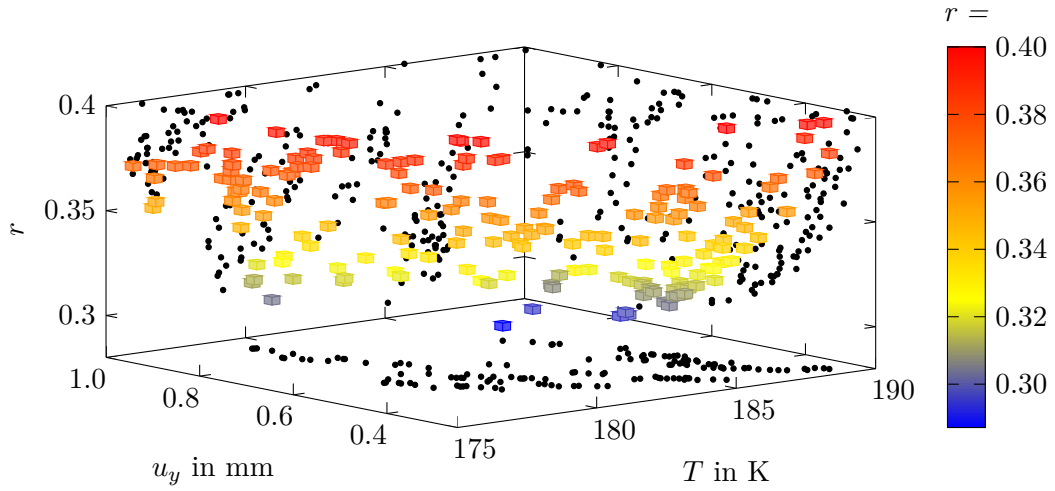
All found structures are thin lines that resemble printed circuit board tracks with a width of two elements. The lightest structure is a simple connection between the two surfaces. The structure with the lowest contact temperature is showing two distinct geometrical features. First, the whole heat drain is covered in material to maximize the heat transfer rate. The second feature is a meandering structure of the connection. All structures shown are path-shaped so the meandering leads to a longer path and thus the thermal resistance of the structure rises. Over the course of the Pareto-set of solutions the meandering decreases in both amplitude and frequency while minimizing the mass. Single solutions show spots at which the width of the connection is more than two elements. This is probably due to an inadequately set GEF parameter  $\varepsilon$ .

The structures seem so circuit board-like probably due to the rectangular structures. No predominant diagonal structures were found in any of the Pareto-optimal solutions. This could be an indicator for anisotropy caused by either the mesh or the genotype-phenotype mapping. By simply regarding the geometrical features one might get the impression that better performing solutions could have been found because the meandering could have been carried on further. The problem is a classical example for solutions that converge towards a micro structure if meshes are continuously refined. The coarse parameter mesh imposes another geometrical constraint by setting a minimum size for any geometrical feature. In the present case this has a positive effect for application because it forces solutions to stay machinable.

### Mechanical and Thermal Objectives

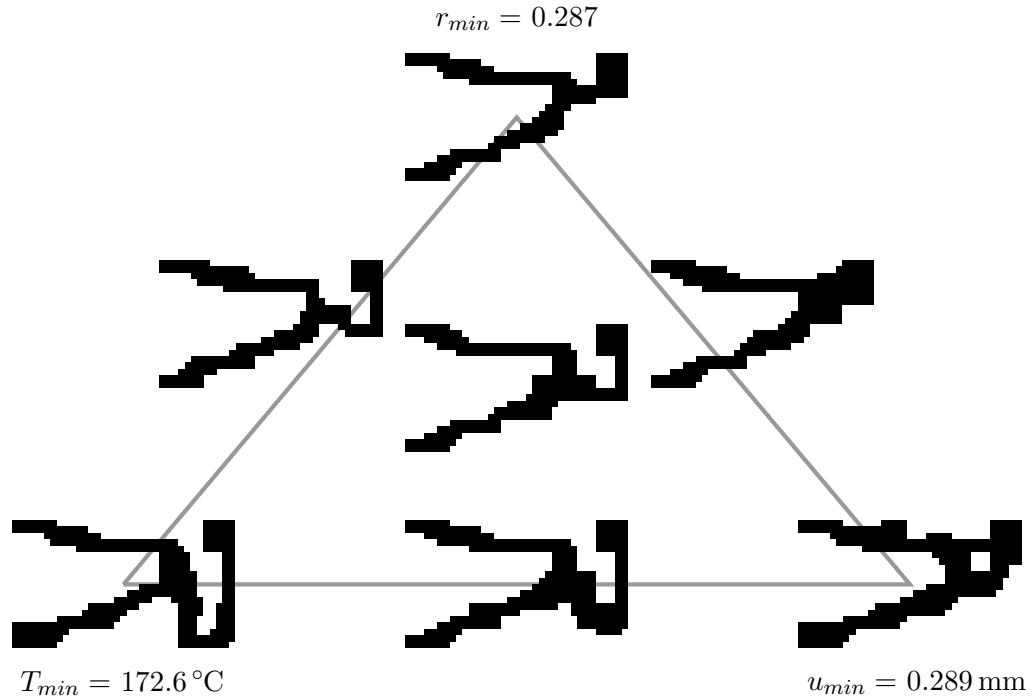
Problem 5 is the first problem tackled in this thesis that is optimized with respect to three objective functions. The three dimensional Pareto-front is plotted in Figure 5.11. 178 solutions were found after 1000 generations with a population size of 300. The scattered solutions in objective space are projected to the boundary planes of the plot in order to illustrate the shape of the front. The scattered solutions are color-coded to emphasize their relative mass.

Seven corresponding solutions are shown in Figure 5.12. The presented solutions were chosen manually from the three dimensional scatter plot in order to represent the single-objective optima, trade-offs between the single-objective optima and a trade-off between all the objectives. Analogous to the results for problem 4, one can see that, in order to minimize the contact temperature, meandering structures and full contact with the right edge are favorable. The structure itself distinguishes significantly from the results presented in Figure 5.10 because multiple constraints are to be taken into account. The stiffest solution resembles the solutions presented in Figure 5.8, but no similar solution was found, even though no constraints would have prevented it. In comparison to Figure 5.8 another constraint is imposed forbidding displacements of less than 1.0 mm. Even considering this additional constraint the solution of minimum weight found for problem 5 outperforms the solution of minimum weight found for problem 3.



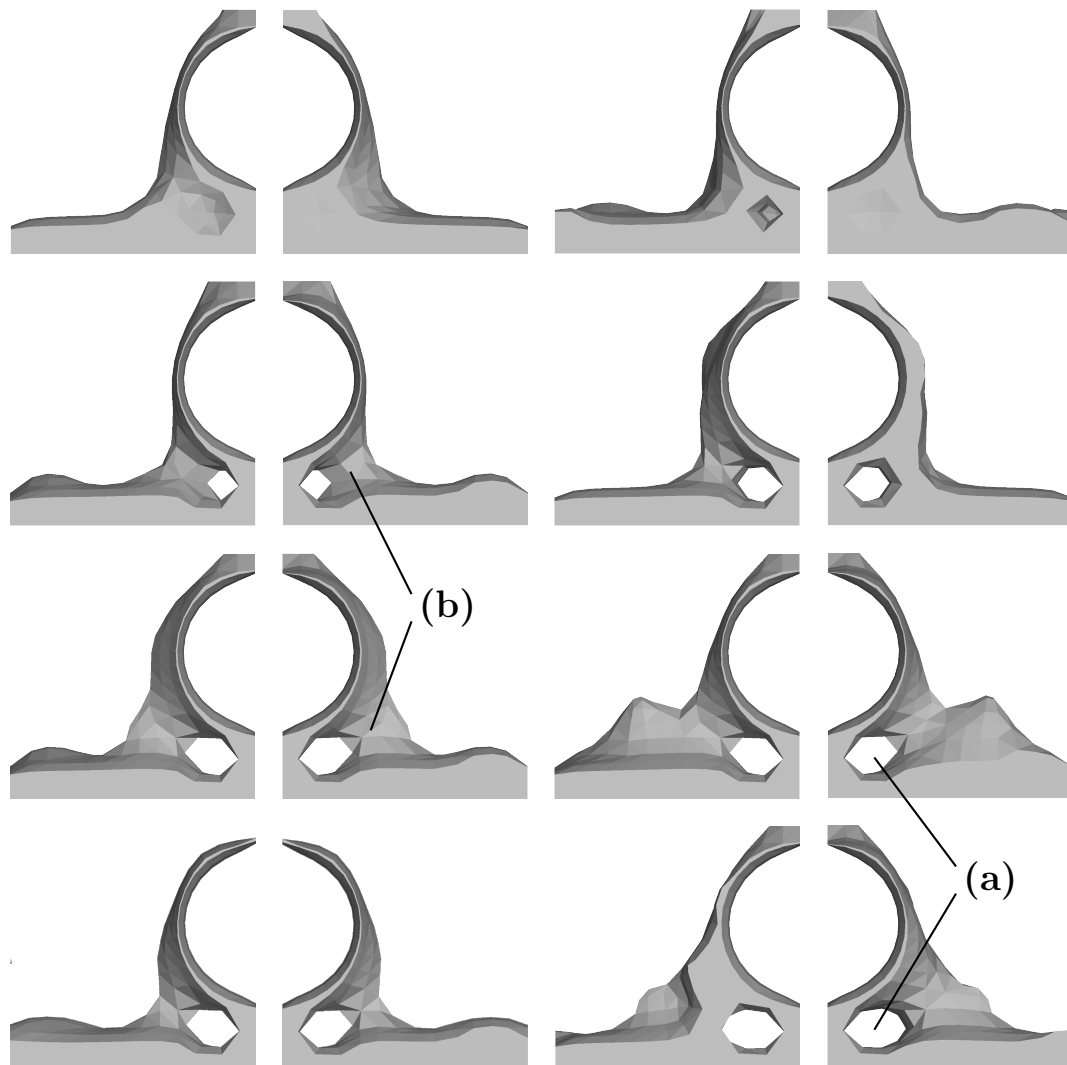
**Figure 5.11:** Pareto-surface of solutions found for problem 5 as thick markers. The markers are color-coded to show the relative mass. The plane is projected to three two dimensional plots shown as black dots in the boundary planes.

This means that both runs of NSGA-II may have in some way failed to fully occupy the allowed objective space and to find optimum structures. Nevertheless, it was possible to easily obtain information about what geometrical features influence the objectives. The trade-off solutions presented in Figure 5.12 illustrate that the geometrical features influence changes when the importance of another objective rises.



**Figure 5.12:** Seven solutions to problem 5 in a triangular arrangement. The corners of the triangle show structures of minimum objectives. Between two corners Pareto-optimal trade-offs between the two objectives are shown. The central structure represents a trade-off between all three objectives..

In Figure 5.13 smoothed solutions to problem 7 are shown from both sides to illustrate their three dimensional structure. 1205 solutions were found over the course of 300 generations with 300 population members. From the shown Pareto-set 81 solutions were extracted that are non-dominated considering only the stiffness and contact temperature of the structure. The three dimensional Pareto-front is thereby reduced to a problem of two objectives. This approach was found superior to a simple two objectives problem with a mass constraint because during the optimization there is a driving force towards lightweight structures rather than just a discarding of too heavy solutions. From that set of 81 solutions eight solutions were picked to be displayed.

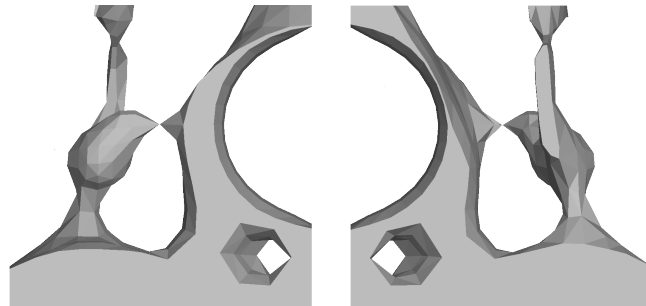


**Figure 5.13:** Eight smoothed solutions to problem 7. Shown are a front view on the left and a back view on the right, respectively. From top left to bottom right the compliance increases and the maximum contact temperature decreases. Interesting geometrical features (a) and (b) can be observed throughout the Pareto-set.

Of course, all solutions have in common the regions that were fixed elements in the parameter mesh. Due to the relative mass objective all solutions showed less material at the side opposite to the shaft hole. The stiffest solution shows material added between the

bottom plate and the hole to reduce the displacement in  $y$ -direction. When the structures get less stiff and towards a lower contact temperature, material is removed below the shaft hole. The shaft hole is supported by single rods of supporting material. In Figure 5.13 it can be seen that the void below the shaft hole (a) enlarges with decreasing contact temperature and that additional supporting material (b) is added farther away from the symmetric axis. Apart from these geometrical features some random features can be observed: Single elements are added to the bottom plate without obvious reason. Some support structures show counter-intuitively much material added to it while others showed non-symmetric features. All in all, the results indicate that not all found solutions are optimal and that the computational effort probably was too little. The used optimizer parameters did not fulfill the rule of thumb given in Equation 3.3. Nevertheless, interesting features could be observed that influence the solutions performance in objective space.

However, selecting fitting solutions from the truncated two dimensional Pareto-front required some engineering knowledge and visual inspection of the Pareto-front because some obviously unfeasible solutions were output. An example for a nonsense solution that shows random outgrowth of material is given in Figure 5.14. In this instance the smoothing algorithm seems to have failed at places where two solid elements were only connected to each other via a single node. This kind of obviously non-optimal result can probably be avoided by a larger population and a greater overall number of generations.

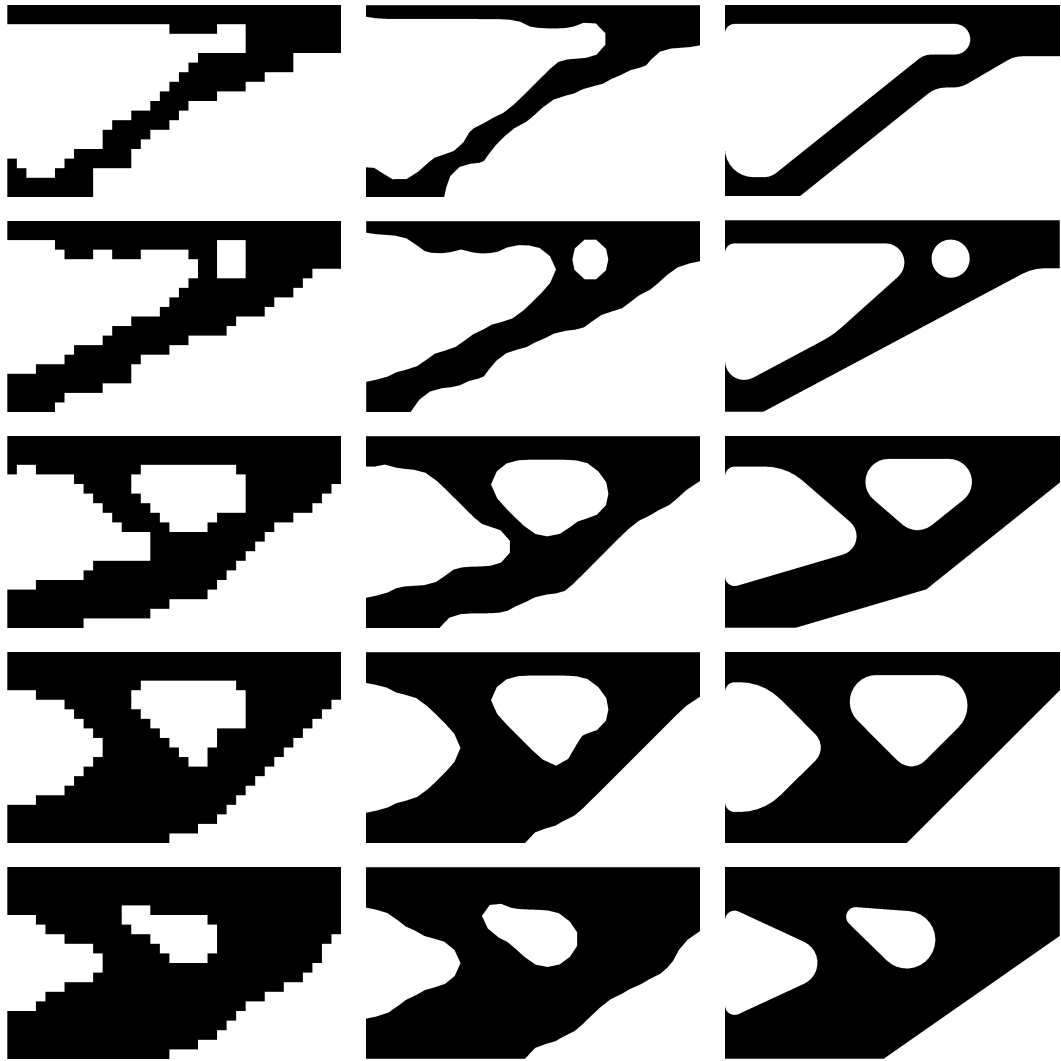


**Figure 5.14:** Unfeasible solution with random material outgrowth. Both sides of the smoothed structure are shown.

### 5.3.1 Digital Image Correlation

The mechanical problem 3 was chosen to be subject to a subsequent derivation of a set of engineering designs. From the Pareto-set depicted in Figure 5.8 five results were selected according to their position in objective space. The chosen structures are shown in Figure 5.15.

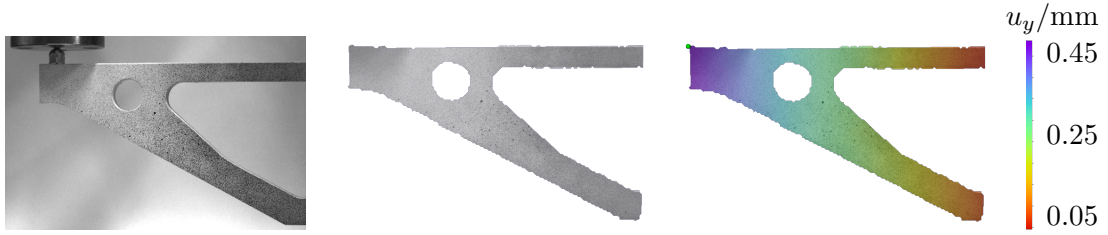
The pixel-like distributions of material and void were smoothed using the previously described smoothing technique. Both of these distributions are shown in Figure 5.15 and both were used to design the area that was to be optimized (gray area in Figure 4.7). During the design process of the specimens for the digital image correlation attention



**Figure 5.15:** Five solutions to problem 3. From top to bottom the relative mass increases from 0.3 to 0.7. On the left are the optimization results, in the middle are the smoothed results and on the right hand side the derived designs are shown.

was paid to keeping the original shape of the results of NSGA-II while simultaneously ignoring obviously random geometries and features that arose from the finite element discretization. Additionally, manufacturing limitations were considered e.g. minimum inside radii and an overall simple design consisting of a manageable number of geometrical features to keep the machining effort on an acceptable level. The design propositions given in Figure 5.15 on the right hand side were found acceptable trade-offs. The relative mass of the solutions changed, but the order was kept from lightest to heaviest structure.

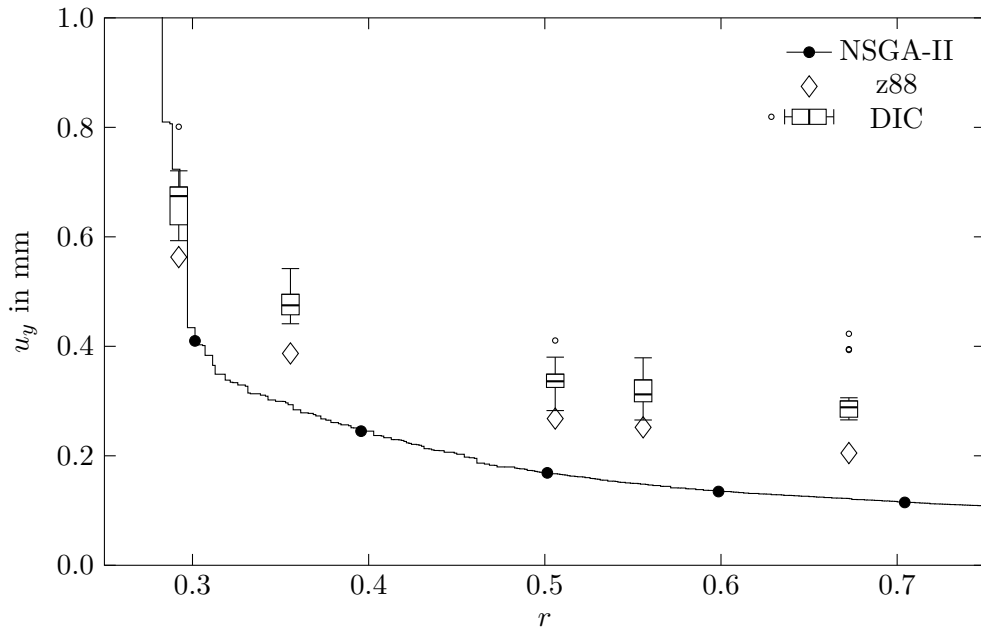
The specimens were prepared, mounted, and loaded as described in Section 4.2. The machined specimens are shown in the appendix. Images were captured of unloaded and loaded samples in order to gain information about the local displacement of the evaluation point in  $y$ -direction. Images and results obtained of a single image correlation are shown in Figure 5.16. An image of the loaded part is shown as well as the structures con-



**Figure 5.16:** Digital image correlation process. From left to right: Initial image, found contour of the deformed part and displacement field with evaluation point.

tours found during the correlation. On the right hand side the obtained  $y$ -displacement field is shown over the deformed sample. The evaluation point is indicated by a green dot. The correlation of the images works semi-automatically. A suitable starting point for the correlation has to be set by hand. Prominent parts of the speckle pattern in the vicinity of the evaluation point were chosen for this purpose. The default parameters of the correlation algorithm were chosen. The results of the image correlation are shown in Figure 5.17 together with the Pareto-front obtained by NSGA-II and predictions about the displacement of the specimens by z88AURORA.

The displacement values are presented in a box plot in order to take account for the statistical variation over the different obtained values. The boxes show the second and third quartile and the median. The whiskers show maximum and minimum values within 1.5 times the interquartile range. Outliers are given as single dots. This way



**Figure 5.17:** Results of the digital image correlation experiments as a box plot. The boxes show the median and the first and third quartile. The whiskers show minimum and maximum values within 1.5 times the interquartile range [51]. Outliers are shown as small circles. Also plotted are finite element analysis predictions and the underlying solutions on the Pareto-front found by NSGA-II.

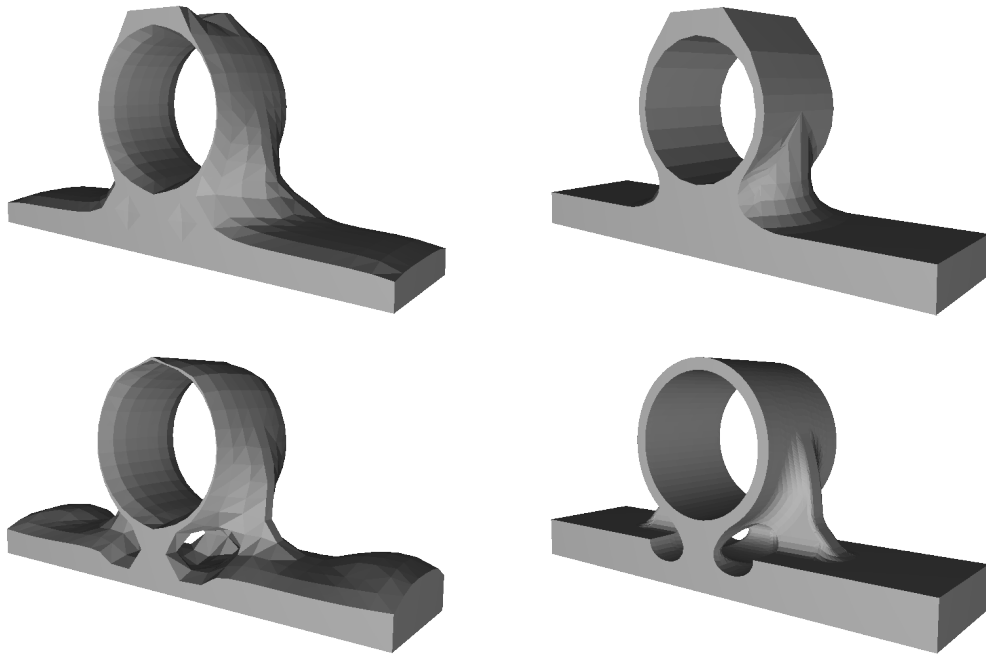
of representing the data considers the small amount of sample points and the lack of knowledge about the underlying statistics [51].

It can be seen that the displacements predicted during a run of NSGA-II are significantly smaller than those found for the parts designed basing on NSGA-II's results. The results of the finite element analysis show that it was possible to obtain a set of trade-off solutions to the mechanical design problem. The median values of the displacement obtained by DIC follow a similar course but are again significantly greater. The deviation of the measured displacements seem roughly equal for the three medium-heavy specimen and was greatest for the lightest and smallest for the heaviest specimen. Less than 6 % of the measured values were outside the range of 1.5 times the interquartile range. The deviation of the measured displacements may be a superposition of many errors. The compliance of the mounting support was measured significantly less than 0.02 mm under full loading. The loading force was captured automatically and was found to have an overall error of less than 5.0 N which is less than 2 %, due to the manual operation of the hydraulic press. Further sources of error may be a flawed speckle pattern or non-optimal settings of focus, aperture and exposure duration of the cameras or the calibration of the image correlation. The theoretical error of the image correlation algorithm lies in the sub-pixel domain [49].

The result plotted in Figure 5.17 strongly indicate that the set of derived designs are in fact trade-off solutions. The actual objective values meanwhile do strongly deviate from those predicted by NSGA-II. This might be due to the adapted design, but it seems more likely to be due to the insufficient spatial discretization that had to be used in order to keep the computational effort tolerable. One can not explicitly state that the found design solutions are optima. Designs derived from Pareto-optimal solutions found by NSGA-II do not dominate each other. Considering that, one might assume that Pareto-optimal points on the pareto front can be used to design Pareto-optimal parts. The results obtained by the finite element analysis seem qualitatively adequate as objective functions even though, quantitatively, they are imprecise.

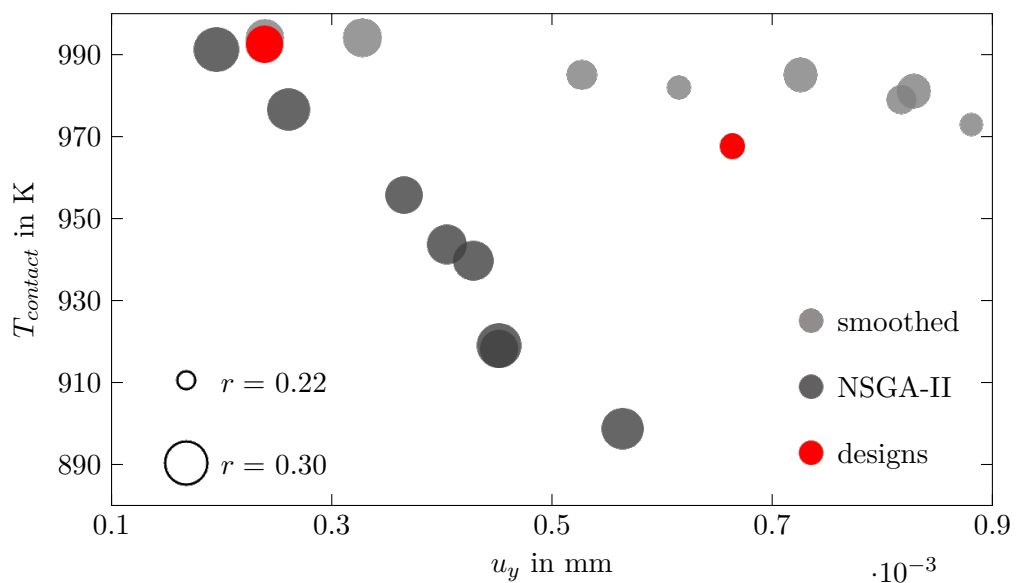
### 5.3.2 Simulations

In order to determine the practical value of the solutions presented in Figure 5.13 further simulations were carried out. The smoothed solutions as well as two derived trade-off designs were regarded. The trade-off designs are shown in Figure 5.18 together with the smoothed solution from which each one was derived. The designs were made using the technique presented in the previous chapter. The results found by NSGA-II show a topology not machinable by classical means. This circumstance is passed on into the derived design. The bottom plate seems significantly thicker for the derived designs. This is an optical effect due to the smoothing and the hence rounded down edges.



**Figure 5.18:** Derived designs of two selected solutions to problem 7. On the left hand side are the smoothed solutions and on the right hand side are the redesigned solutions.

The smoothed solutions were meshed using the TETGEN<sup>8</sup> mesh generator included in z88AURORA and subsequently thermal and mechanical calculations were conducted. The results were compared to results obtained using COMSOL. Mesh sizes were chosen appropriately by comparing different discretization levels. The results obtained by



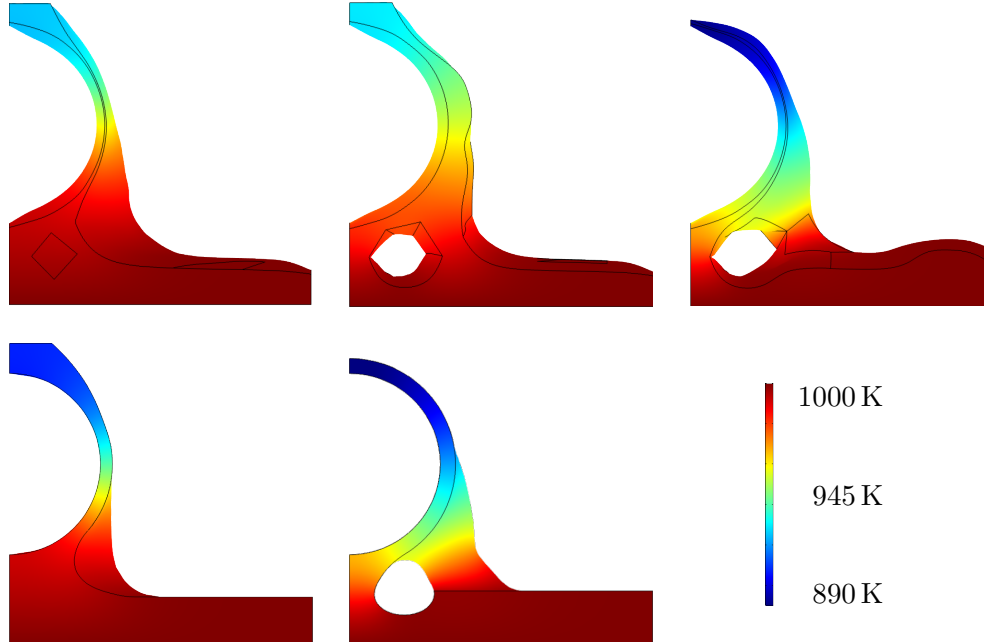
**Figure 5.19:** Solutions to problem 7 in objective space. Shown are the structures of Figure 5.13 found by NSGA-II on a coarse mesh, the smoothed structures and the derived designs as shown in Figure 5.18. Circle radii correspond to the relative mass of the structure.

<sup>8</sup><http://wias-berlin.de/software/tetgen/>



z88AURORA are very close to those found by COMSOL and are plotted in Figure 5.19 together with the results obtained by NSGA-II. The relative mass is illustrated by the radii of the circles.

It can be seen that the objective values obtained on a sufficiently fine mesh differ strongly from those obtained on the rather coarse meshes used in the optimizer. Displacements and contact temperature were estimated too low. This has already been observed and measured in the previous section. Different from these observations, for problem 7 the regarded set of trade-off solutions was not non-dominated when further evaluations were conducted. This may be due to the influence of the smoothing tool. As already mentioned, even though being the second largest to be optimized in this thesis, the finite element mesh used is considerably coarse. In this instance the smoothing shows stronger influence to the later structure. This can also be observed in Figure 5.19, where the smoothed solutions show significantly decreased relative masses. In Figure 5.20 three temperature distributions are shown on three exemplary smoothed solutions that were obtained in COMSOL. The decreasing contact temperature can be observed by the color coding. Further, it can be observed that the supporting material on which the bearing rests effectively reduces thermal conduction. The thermal resistance rises and much lower contact temperature is achieved. The reduction of material below the loading force of course leads to a bigger displacement.



**Figure 5.20:** Temperature distribution in side views of bearing mount design propositions. Top: Three smoothed solutions obtained by NSGA-II. From left to right the stiffness and the contact temperature decrease. Bottom: Two derived designs. The left design is stiffer but shows a higher contact temperature than the right design.

The two derived designs presented in Figure 5.18 were also subject to further simulations. Temperature distributions found in COMSOL are also presented in Figure 5.20.

The performance of the derived design propositions in objective space is depicted in Figure 5.19 next to the proposed Pareto-front obtained by NSGA-II and the performance of the smoothed solutions. The two designs are not dominated by each other and therefore trade-off solutions. It can be observed that the derived designs perform worse than the Pareto-front would suggest. This is in accordance with previous findings and probably also an effect of the too coarse mesh. Nevertheless, in terms of domination, the two designs perform better than the smoothed solutions. This is especially true for the less stiff solution. This may be due to inaccuracies during the redesigning process. All in all, it was possible to find trade-off engineering designs and corresponding geometrical features that influence the performance in objective space.

## 6 Discussion and Outlook

In this chapter the methodology and findings described earlier are to be brought into context and are to be discussed. First, the results of the optimizations are to be reviewed and, from that point of view, implications for evolutionary algorithms and multi-objective optimization in engineering are derived. Finally, an outlook is given towards which evaluations and which features might provide further insight into the capability of the presented method.

### 6.1 Discussion of the Results

The choice of a fitting optimization approach was based on considerations about the number of individual solutions that are found, their even distribution in objective space and their proximity to an optimum structure. The decision to choose an evolutionary algorithm and especially NSGA-II was based on the assumption that accepting solutions which are probably only near-optimum may be a rewarding compromise, especially when probabilistic operators are involved. Even for comparably small structures it was never possible to find the same structure during more than one out of ten independent runs of the optimizer. Nevertheless, found structures differed only in a comparably small number of flipped bits. It was found that a large amount of evenly distributed trade-off solutions can provide better insights into the optimization problem. Evolutionary algorithms are considered the best performing approaches for multi-objective optimization when these qualities of the Pareto-set are desired [52]. Interesting insights that were gathered throughout this thesis concerned geometrical features and their specific influence on the design's performance in objective space. Subsequently, reducing the deliberately large set of trade-off solutions to a clearly arranged set of diverse clustered structures was found helpful for analyzing the Pareto-optimal structures.

A rough estimate about the approaches computational extensiveness and the required computing time revealed that even if one would accept near-optimum solutions, a significant reduction of the computational effort is to be pursued. The described measures to reduce the computational effort showed success and the results presented in Section 5 show that the described optimizer is reliably capable of finding sets of generation-wise improving trade-off structures for problems of diverse physics. The application of a multi-step genotype-phenotype mapping procedure ensures that the structures meet certain criteria crucial for engineering application. This is achieved by a heavy reduction of the permissible parameter space.

Numerical instabilities that are common for topology optimization were also encountered during this thesis. Checkerboarding was suppressed right from the start as combinatorial approaches to generate structures are usually prone to this kind of behavior [5, 7]. The feasibility filtering and the GEF both efficiently suppress the formation of such patterns. Non-uniqueness of the solutions was not explicitly dealt with because in complex design problems it is uncommon [32]. Getting trapped in local optima was avoided intrinsically by the use of evolutionary algorithms. Nevertheless, statements about the quality of an optimum could only be made in a comparing manner. Results indicate that often better minima were found compared to classical approaches.

A numerical issue that was observed frequently and whose occurrence has great influence on the resulting structure is the mesh dependency. Very coarse meshes had to be used in order to keep the computational effort on a reasonable level, so minimum dimensions of any geometrical feature are limited. This is especially an issue when regarding unconstrained problems of thermal conduction or very light weight structures because these can strongly benefit from fine structures. However, the resolution of the parameter mesh also acts as an additional geometrical constraint preventing convergence towards micro structures and thus allowing for finely discretized finite element meshes to be used. By using cubically shaped elements and a structured meshing algorithm the resulting meshes are also prone to anisotropic behavior because the filter renders diagonal structures broader. An adequate choice of the parameter mesh is in any case fundamental for the successful application of the presented method. The definition of the genotype-phenotype mapping easily allows for arbitrary element types and mesh structures to be used.

As for all topology optimization approaches that rely on element-wise density distributions, it is not possible to define surface dependent boundary conditions without further ado. This is usually less of a problem for solely mechanical design problems but is more relevant when considering e.g. thermal problems that are subject to convective heat transfer. If surface dependent effects were to be considered the genotype-phenotype mapping presented in this thesis would fail because ragged surfaces would be beneficial rather than disadvantageous. Also, non-linearities had to be considered that would take more computational effort to solve. However, the strict 0-1 distribution simplifies the distinction between material and void and the definition of surfaces, which may be beneficial for surface dependent boundary conditions.

While the derived designs for problem 3 could successfully be machined and tested, the results of an optimization of problem 7 were still far from manufacturing and application. This is mainly due to problems that any complex topology optimization holds. Solutions to three dimensional problems can contain complex topologies and advanced additive manufacturing processes may have to be considered. Also, in the scope of this thesis only linear models were used. For a universally applicable and reliable development convection and temperature dependent material behavior have to be considered.

## 6.2 Evolutionary Algorithms in Topology Optimization

Apart from the considerations of the preceding section, another important trait of NSGA-II, as an evolutionary algorithm, is the comparably low number of parameters that have to be set initially. The user only has to appropriately set the population size and the number of overall generations. For the population size one might rely on the rule of thumb that for every binary parameter there should at least be one individual in the population. Larger populations usually lead to more diversity and a higher probability of finding global optima [38]. Complex structures, e. g. meandering structures, were found to benefit from further increased population sizes. Unfortunately, information about the type of optimum structure is not always available a priori.

A poorly set abortion criterion is usually not critical because further generations can always be added or conducted later. Working with evolutionary algorithms, in contrast to converging optimization approaches (in the mathematical sense), more computing time holds the possibility of finding completely new optima without the need to change the problem definition or the optimizer's parameters. However, the decision to accept the current results as the obtained solution or to add more generations is up to the user and depends on the user's experience and knowledge.

It was found that constraint handling works rather well even when relatively simple constraint handling mechanisms are implemented. It was always possible to find appropriate solutions even to problems with more than one constraining function. This is not common amongst all optimization algorithms and an important trait of evolutionary algorithms. Considering this and the relatively easy way to set parameters, the problem definition is not always trivial. The correct choice of objective functions, constraining functions and their implementation can be problematic and requires understanding of the algorithm as well as of the problem's physics and numerics. Too harsh constraint handling will lead to a disadvantageous loss of diversity in the parental population. Therefore, more gentle implementations are to be preferred even if they may require increased population sizes and more generations.

Considering the demands for universally versatile topology optimization approaches, given in [33], the presented approach, like evolutionary algorithms in general, performs well in terms of starting guess independence, constraint handling and applicability to arbitrary physics. The presented approach requires some insights into the algorithm and the problem's physics and numerics but is, apart from that, relatively straight-forward to use. The approach was also found somewhat prone to mesh dependent effects. The critical disadvantage that evolutionary algorithms hold is the extensive computational effort to solve even moderately sized problems. The largest mesh of 2000 elements that was optimized in the scope of this thesis is considerably bigger than meshes that were optimized using evolutionary multi-objective optimizers in published literature in this way before. Nevertheless, the mesh size was too coarse for the finite element analysis to generate reliable system responses. Considering the fact that the SIMP approach can

easily handle mesh sizes that are larger by multiple magnitudes, the presented approach does not pass the crucial demand for efficiency on large scale problems.

### 6.3 Multi-Objective Optimization in Engineering

Multi-objective optimization in engineering is an issue that has been tackled by various approaches developed before. Common techniques to consider multiple objectives include adding up the stiffness of two independent loading cases [31] or combining mathematical compliance minimization methods and empirical, biomimetic approaches that aim to find an even stress distribution<sup>9</sup>. While such approaches efficiently find single Pareto-optimal structures they are usually less capable of providing the insights that a fully occupied Pareto-front can offer. As described in Section 5.3, one can gain information about what geometrical feature influences the designs performance. This could be used to create a parameterized CAD design that can be subject to further shape optimization and sizing.

It was found that two dimensional Pareto-fronts were relatively easy to handle and to gain information from. When a three dimensional Pareto-surface is to be analyzed one has to rely on advanced plotting techniques and tracking the change of the structure in dependence of its position in objective space becomes more extensive. A reduction of the Pareto-front to two objectives was found helpful. Sometimes constraints had to be reinforced by not only penalizing violating solutions but also to minimize them. Such objectives may be suppressed in further visualization steps. For more than three objective functions the output information may be even more overwhelming and a projection of the Pareto-front to elements of lower dimensionality is inevitable. Sometimes manual selection of appropriate designs is required, especially when obviously non-optimal solutions are found due to too few generations.

Because the presented method fails to provide near-design solutions and precise objective estimates, its possible area of application would be in a very early stage of the designing process. It may give an engineer helpful information about the feasible solution space and may be a good basis for further more detailed structural optimization.

### 6.4 Outlook

In this thesis a method for multi-objective optimization using evolutionary algorithms is presented. The focus lay on reducing the computational effort in order to be able to solve medium scale problems in reasonable time. Yet, the efficiency of the procedure leaves much to be desired. Further features may be added in order to account for the specifics of the regarded problems. Amongst such feature are specialized operators to generate structures. A mass-conserving crossover operator is presented in [6]. Other approaches

---

<sup>9</sup><http://z88.de/z88arion/>

may include element-numbering independent crossover. For improved practicability it may be helpful to be able to automatically generate the finer mesh from the parameter mesh and to assign the corresponding boundary conditions. This would also allow for multi-stage GEF as it is proposed in [46]. Simultaneous optimization of the parameter bit string and the GEF parameter  $\varepsilon$  would reduce the influence of its correct setting on the feature size of the found structures and would allow for finer and bulkier solutions to be found more easily. The binary representation of the element density allows for simplification of the finite element program used to gain system responses. For uniform elements with a binary distribution of the material data only two different element stiffness matrices had to be determined before. This would save additional time compared to determining a stiffness matrix for every element.

Also, one might want to take a deeper look into the general foundation of the multi-objective optimization strategy. An extensive study of the suitability of the diverse population based optimizers may reveal even better suited meta-heuristics for multi-objective optimization. Also for many-objective problems specialized algorithms exist [22, 30]. Different definitions of the structure or combinations of several optimization concepts could also be subject of a comparative study. Maybe less promising approaches, e.g. single-objective methods, can be tuned to outperform the inefficient population based methods.

Apart from improvements of the procedure itself further studies are to be conducted in order to determine and demonstrate the limits and capabilities of the presented method. Superiority in finding global minima over the SIMP approach was indicated by the results of this thesis but no statement of general validity can be made yet. For this purpose, the reliability of the results had to be ensured. A very promising approach would be to use significantly finer finite element meshes while keeping the same coarse parameter mesh. Large scale parallelization could also boost the efficiency and open the way for bigger structures to be optimized. This ,however, would not prove industrial applicability because access to large-scale computer clusters is not common for most companies yet. To verify the predictions about potential use in industrial engineering, the method has to be embedded into an existing design process and further tests have to be conducted considering industrial demands.

## 7 Summary

The scope of this thesis was to develop a multi-objective topology optimization method. An engineering part is optimized with respect to several objective functions. Such an optimization results in a set of Pareto-optimal solutions that are all trade-off solutions to the problem. None of these solutions is better than another with respect to all objectives.

The evolutionary algorithm NSGA-II was found the best fitting for this task and thus implemented. An estimate of the required computational effort showed that even moderately sized problem could only be solved with great effort. The optimization tool was therefore extended with an adequate genotype-phenotype mapping and some minor features. The extensions accelerate the search for optimum structures and ensure applicability of the structures in engineering. Recommendations for adequate parameter setting were derived from literature.

To evaluate the performance of the optimizer seven design problems were defined and solved. Mechanical and thermal objectives were considered and to some extent harsh constraints were imposed. The optimizer proved to be able to always find large and evenly distributed solutions sets. The shape of the solutions and influence of geometrical features were analyzed and discussed. Some design problems were considerably larger than those presented in literature and could still be solved in a satisfactory manner. The results indicate that the presented method may find better results than classical approaches might do. Nevertheless, the computational effort is still too great to really outperform them.

From the obtained solutions five design propositions for a beam were derived and manufactured. Deformation of the parts was measured by digital image correlation. Results showed that it is possible to derive trade-off solutions but also that their performance stays behind the predictions. This is due to the insufficient discretization of the design space that is needed to keep the computational effort on a reasonable level. Similar observations could be made for other derived designs and simulations.

The great computational effort and quantitative error make this method not suitable as a stand-alone tool. Its versatility with respect to physics and constraint may add to understanding a design problem in an early stage of the design process.



## 8 Zusammenfassung

Ziel der vorliegenden Arbeit war es, eine multikriterielle Topologieoptimierungsmethode zu entwickeln. Dabei wird ein Bauteil nach mehreren gleichrangigen Zielfunktionen optimiert. Das Ergebnis einer solchen Optimierung ist eine Menge an Pareto-optimalen Lösungen, die alle verschiedene Kompromisslösungen darstellen. Keine dieser Lösungen ist in jeder Hinsicht besser als eine andere.

Der evolutionäre Algorithmus NSGA-II wurde als für am besten geeignet befunden und implementiert. Da eine Abschätzung des Rechenaufwandes ergab, dass selbst mittelgroße Probleme nur mit großem Aufwand zu lösen sein würden, wurde das Optimierungstool um ein problemspezifisches Genotyp-Phänotypmapping und mehrere kleinere Features erweitert. Diese beschleunigen einerseits die Suche nach optimalen Strukturen und andererseits stellen sie sicher, dass diese Strukturen einen gewissen Wert für die ingenieurtechnische Anwendung haben. Es konnten darüber hinaus Empfehlungen für eine korrekte Parameterwahl aus der Literatur abgeleitet werden.

Um die Performance der Optimierungsmethode zu testen, wurden sieben Probleme definiert und gelöst. Die Probleme wurden nach mechanischen und thermischen Zielfunktionen optimiert und waren teilweise deutlichen Beschränkungen unterworfen. Es konnte gezeigt werden, dass der evolutionäre Algorithmus immer in der Lage ist, große und gleichmäßig besetzte Lösungsmengen zu finden. Die Gestalt und der Einfluss einzelner geometrischer Substrukturen der Lösungen wurden analysiert und diskutiert. Die untersuchten Probleme waren teilweise deutlich größer als die in der Literatur dargestellten und es konnten trotzdem zufriedenstellende Ergebnisse gefunden werden. Die Ergebnisse legen den Schluss nahe, dass die vorgestellte Methode in der Lage ist, deutlich bessere Lösungen zu finden als klassische Ansätze. Der Rechenaufwand ist jedoch noch immer deutlich zu groß, um diesen ernsthaft Konkurrenz machen zu können.

Aus den gefundenen Lösungen wurden fünf Pareto-optimale Entwürfe für einen Balken abgeleitet und hergestellt. Mit Hilfe von digitaler Bildkorrelation wurden die Verformung der Teile gemessen. Die Ergebnisse zeigten, dass es mit dem vorgestellten Tool möglich ist, Kompromisslösungen zu erhalten. Die tatsächlichen Werte der Zielfunktionen reichen jedoch nicht an die vorhergesagten heran. Das liegt an der ungenügenden Diskretisierung des Bauraums, die nötig ist, um den Rechenaufwand auf einem tolerierbaren Niveau zu halten. Ähnliches wurde bei weiteren abgeleiteten Entwürfen und Simulationen beobachtet.

Der große Rechenaufwand bei gleichzeitigen quantitativen Fehlern führt dazu, dass die Methode kaum als alleinstehende Anwendung tauglich ist. Die Vielseitigkeit im Bezug auf

physikalische Modelle und Beschränkungen könnte jedoch in einem frühen Stadium des Konstruktionsprozesses zu zusätzlichem Kenntniserwerb über das Problem beitragen.

## Bibliography

- [1] S. S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Hoboken, NJ, 4th edition, 2009.
- [2] R. J. Yang and A. I. Chahande. Automotive Applications of Topology Optimization. *Structural and Multidisciplinary Optimization*, 9(3):245–249, 1995.
- [3] H. A. Eschenauer and N. Olhoff. Topology Optimization of Continuum Structures: A Review. *Applied Mechanics Reviews*, 54(4):331–390, 2001.
- [4] K. Deb, K. Sindhya, and J. Hakanen. Multi-Objective Optimization. In R. N. Sengupta, A. Gupta, and J. Dutta, editors, *Decision Sciences*. CRC Press, Boca Raton, 2016.
- [5] K. Deb and T. Goel. A Hybrid Multi-Objective Evolutionary Approach to Engineering Shape Design. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 385–399. Springer, 2001.
- [6] J. F. Aguilar Madeira, H. Rodrigues, and H. Pina. Multi-Objective Optimization of Structures Topology by Genetic Algorithms. *Advances in Engineering Software*, 36(1):21–28, 2005.
- [7] T. Kunakote and S. Bureerat. Multi-Objective Topology Optimization using Evolutionary Algorithms. *Engineering Optimization*, 43(5):541–557, 2011.
- [8] S. I. V. Peña, S. B. Rionda, and A. H. Aguirre. Multiobjective Shape Optimization Using Estimation Distribution Algorithms and Correlated Information. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 664–676. Springer, 2005.
- [9] S. Bureerat and S. Srisomporn. Optimum Plate-Fin Heat Sinks by using a Multi-Objective Evolutionary Algorithm. *Engineering Optimization*, 42(4):305–323, 2010.
- [10] S. Kanyakam and S. Bureerat. Multiobjective Optimization of a Pin-Fin Heat Sink Using Evolutionary Algorithms. *Journal of Electronic Packaging*, 134(2):021008, 2012.
- [11] S. Kanyakam and S. Bureerat. Multiobjective Evolutionary Optimization of Splayed Pin-Fin Heat Sink. *Engineering Applications of Computational Fluid Mechanics*, 5(4):553–565, 2014.

- 
- [12] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method: Volume 1: The Basis*. Butterworth-Heinemann, Oxford, 5th edition, 2002.
- [13] R. T. Fenner. *Finite Element Methods for Engineers*. Imperial College Press, London, 2nd edition, 2013.
- [14] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method: Volume 2: Solid Mechanics*. Butterworth-Heinemann, Oxford, 5th edition, 2002.
- [15] S. S. Rao. *The Finite Element Method in Engineering*. Elsevier, Amsterdam, 5th edition, 2011.
- [16] D.-Z. Du, P. M. Pardalos, and W. Wu. *Mathematical Theory of Optimization*. Springer, Boston, MA, 2001.
- [17] P. Pedregal. *Introduction to Optimization*. Springer, New York, 2011.
- [18] S. Baluja. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical report, Carnegie-Mellon University, Dept. of Computer Science, Pittsburgh, PA, 1994.
- [19] F. J. Ayala, editor. *Essential Readings in Evolutionary Biology*. Johns Hopkins Univ. Press, Baltimore, MD, 2014.
- [20] M. Mitchell. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA, 2001.
- [21] S. Bureerat and K. Sriworamas. Population-Based Incremental Learning for Multiobjective Optimisation. *Soft Computing in Industrial Applications*, 39:223–232, 2007.
- [22] H. Jain and K. Deb. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014.
- [23] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.
- [24] J. Branke, editor. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer, Berlin, 2008.
- [25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

- [26] N. Srinivas and K. Deb. Multi-Objective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [27] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- [28] A. Konak, D. W. Coit, and A. E. Smith. Multi-Objective Optimization using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.
- [29] C. A. Coello Coello and M. S. Lechuga. MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 1051–1056, 2002.
- [30] K. Deb and H. Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.
- [31] M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods, and Applications*. Springer, Berlin and Heidelberg, 2nd edition, 2004.
- [32] O. Sigmund and J. Petersson. Numerical Instabilities in Topology Optimization: A Survey on Procedures Sealing with Checkerboards, Mesh-dependencies and Local Minima. *Structural and Multidisciplinary Optimization*, 16(1):68–75, 1998.
- [33] O. Sigmund and K. Maute. Topology Optimization Approaches. *Structural and Multidisciplinary Optimization*, 6(48):1031–1055, 2013.
- [34] W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.
- [35] G. Taubin. A Signal Processing Approach to Fair Surface Design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 351–358, New York, NY, 1995. ACM.
- [36] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM, 1999.
- [37] I. Farmaga, P. Shmigelskyi, P. Spiewak, and L. Ciupinski. Evaluation of Computational Complexity of Finite Element Analysis. In *The Experience of Designing and Application of CAD Systems in Microelectronics*, Piscataway, NJ, 2011.

- [38] J. T. Alander. On Optimal Population Size of Genetic Algorithms. In *Comp Euro '92. 'Computer Systems and Software Engineering'*, pages 65–70, Los Alamitos, CA, 1992.
- [39] D. Harel and Y. Feldman. *Algorithmics: The Spirit of Computing*. Springer, Berlin, 3rd edition, 2012.
- [40] F. Mendoza, J. L. Bernal-Agustin, and J. A. Dominguez-Navarro. NSGA and SPEA Applied to Multiobjective Design of Power Distribution Systems. *IEEE Transactions on Power Systems*, 21(4):1938–1945, 2006.
- [41] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 29(2):147–160, 1950.
- [42] I. Harvey. Artificial Evolution: A Continuing SAGA. *Lecture Notes in Computer Science*, 2217:94–109, 2001.
- [43] A. E. Eiben and S. K. Smit. Evolutionary Algorithm Parameters and Methods to Tune Them. *Autonomous Search*, pages 15–36, 2012.
- [44] S. Bureerat, A. Boonapan, T. Kunakote, and J. Limtragool. Design of Compliance Mechanisms using Topology Optimisation. In *19th Conference of Mechanical Engineering Network of Thailand*, pages 421–427, Phuket, 2005.
- [45] S. Bureerat and J. Limtragool. Performance Enhancement of Evolutionary Search for Structural Topology Optimisation. *Finite Elements in Analysis and Design*, 42(6):547–566, 2006.
- [46] S. Bureerat and J. Limtragool. Structural Topology Optimisation using Simulated Annealing with Multiresolution Design Variables. *Finite Elements in Analysis and Design*, 44(12-13):738–747, 2008.
- [47] S. Even, G. Even, and R. M. Karp, editors. *Graph Algorithms*. Cambridge Univ. Press, Cambridge, 2nd edition, 2012.
- [48] N. Olhoff, E. Ronholt, and J. Scheel. Topology Optimization of Three-Dimensional Structures using Optimum Microstructures. *Structural Optimization*, 16(1):1–18, 1998.
- [49] B. Pan, K. Qian, H. Xie, and A. Asundi. Two-Dimensional Digital Image Correlation for In-Plane Displacement and Strain Measurement: A Review. *Measurement Science and Technology*, 20(6):062001, 2009.
- [50] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary Many-Objective Optimization: A Short Review. In *IEEE Congress on Evolutionary Computation*, pages 2419–2426, Piscataway, NJ, 2008.

- [51] M. Frigge, D. C. Hoaglin, and Bo. Iglewicz. Some Implementations of the Boxplot. *The American Statistician*, 43(1):50, 1989.
- [52] R. T. Marler and J. S. Arora. Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization*, 26(6):369–395, 2004.

# Appendix

## NSGAI1.ctrl

```
#OBJECTIVE FUNCTIONS AND CONSTRAINTS
OBJECTIVE          = TEMPERATURE          # objective 1
iparam1            = 1492                  # node number
max                = 200                   # constraint
OBJECTIVE          = DISPLACEMENT        # objective 2
iparam1            = 1492                  # node number
iparam2            = 2                     # z-direction
max                = 5                     # constraint
OBJECTIVE          = MASS                 # objective 3
max                = 0.75                  # constraint
min                = 0.25                  # constraint
OBJECTIVE          = MAX_STRESS           # additional constraint
max                = 200                   # constraint

#MATERIAL PARAMETERS
E                  = 70000                  # elastic modulus
nu                 = 0.33                  # poisson's ratio
lambda             = 0.2                   # thermal conduction

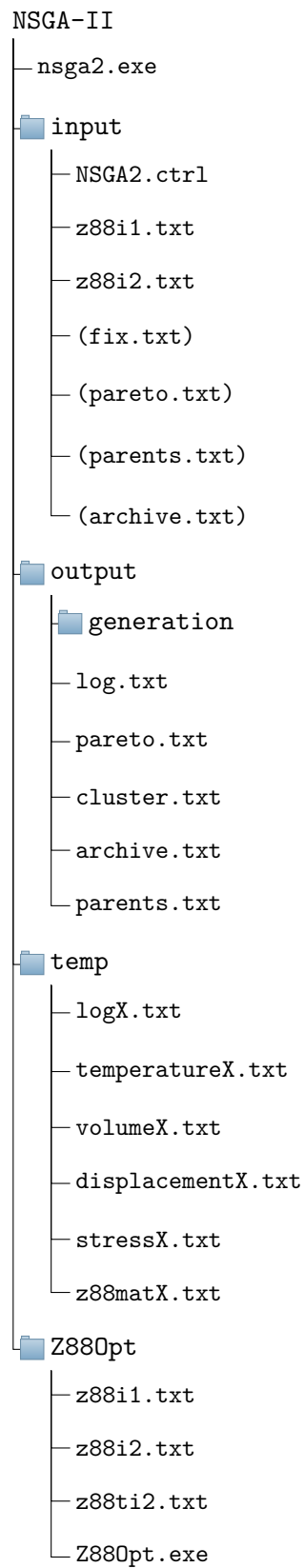
#ALGORITHM PARAMETERS
number_of_objectives = 3                   # minimize first 3 objectives
population_size      = 300
number_of_generations = 300
cross_over_probability = 0.95
mutation_probability = 0.01
rel_cluster_distance = 0.1
gef_param            = 0.1
filter               = ON
```



## Implemented Objectives

Key	iparam1	iparam2	pathparam	Description
EMPTY	-	-	-	returns 0.0
MASS	-	-	-	relative mass $r$
DISPLACEMENT	node	0,1,2 = $x,y,z$ 3 = euclid.	-	displacement at specified node in specified direction
MAX_STRESS	-	-	-	Maximum VAN MISES element stress
COMPLIANCE	-	-	-	Simplified compliance measure $e^*$
TEMPERATURE	node	-	-	Temperature at specified node
MAX_TEMPERATURE	-	-	rel. path to list of nodes	Maximum temperature in a set of nodes


## Folder and File Structure



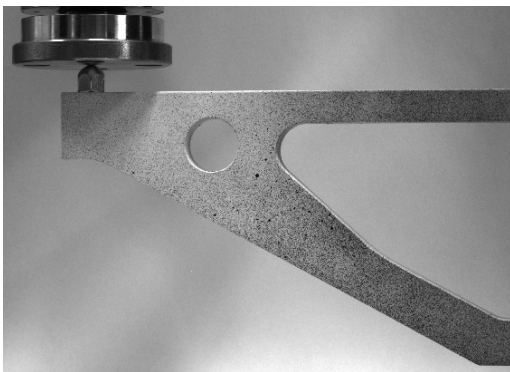
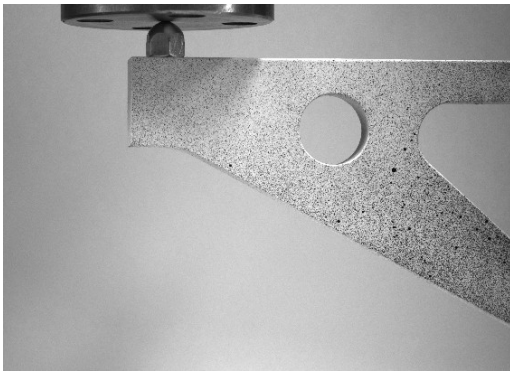
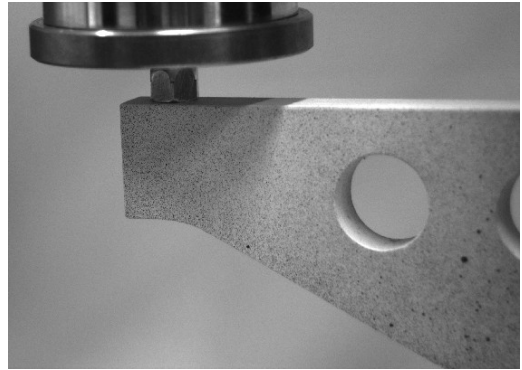
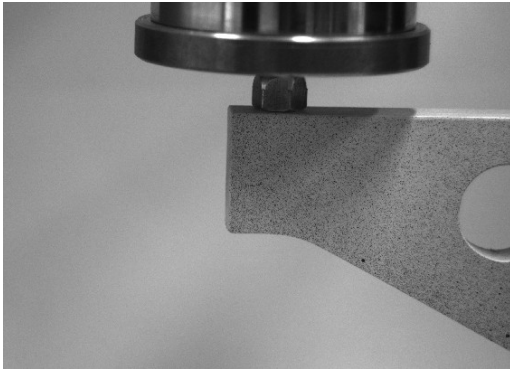
## Design Problems

Problem No.	Shape	Physics	Material		Mesh Size	Binary Param.	Objectives	Constraints
			Data					
1	Beam	mechanical	$E = 200 \text{ GPa}$		300	108	MASS	$r \in [0.5; 0.7]$
			$\nu = 0.30$				DISPLACEMENT	$\sigma < 200 \text{ MPa}$
2	Bridge	mechanical	$E = 200 \text{ GPa}$		400	90	MASS	$\sigma < 200 \text{ MPa}$
			$\nu = 0.30$				COMPLIANCE	
3	Beam	mechanical	$E = 70 \text{ GPa}$		700	243	MASS	$r \in [0.25; 0.75]$
			$\nu = 0.27$				DISPLACEMENT	$\sigma < 200 \text{ MPa}$
4	Beam	thermal	$\lambda = 200 \text{ W m}^{-1} \text{ K}^{-1}$		700	243	MASS	$r < 0.7$
5	Beam	mechanical & thermal	$E = 70 \text{ GPa}$		700	243	MASS	$r < 0.4$
			$\nu = 0.27$				TEMPERATURE	$\sigma < 200 \text{ MPa}$
			$\lambda = 200 \text{ W m}^{-1} \text{ K}^{-1}$				DISPLACEMENT	$u_y < 1 \text{ mm}$
6	Cubical	mechanical	$E = 200 \text{ GPa}$		2000	427	MASS	$r < 0.8$
			$\nu = 0.30$				DISPLACEMENT	$\sigma < 200 \text{ MPa}$
7	Bearing Mount	mechanical & thermal	$E = 60 \text{ GPa}$		1404	342	MASS	$r < 0.4$
			$\nu = 0.30$				MAX_TEMP.	$\sigma < 80 \text{ MPa}$
			$\lambda = 15 \text{ W m}^{-1} \text{ K}^{-1}$				DISPLACEMENT	

## Data Sheet AlCuMgPb

 <b>Werkstoff-Datenblatt AlCuMgPb</b> aushärtbar										Gattung K 8 Automaten- legierungen		Blatt-Nr. K 8.3			
<b>Zusammensetzung in % der Masse (Gew.), Rest Al</b>										<b>Bezeichnungen:</b> DIN 1725 T. 1: AlCuMgPb Werkstoff-Nr.: 3.1645 Internationales Legierungsregister: 2007 ISO: - Frankreich: (2030/A-U4Pb) Großbritannien: - Italien: -					
Si	Fe	Cu	Mn	Mg	Cr	Zn	Ti	sonstige Elemente, Bemerkungen	Andere Elemente einz.   zus.						
0,8	0,8	3,3 -4,6	0,50 -1,0	0,40 -1,8	0,10	0,8	0,20	Ni 0,20; Bi 0,20 Pb 0,8-1,5; Sn 0,20	0,10	0,30					
<b>Mindestwerte der mechanischen Eigenschaften nach DIN</b>										<b>Anwendbar nach:</b> DIN 4113 nein AD-Merkbl. W. 6/1 nein German Lloyd nein Deutsche Bundes- bahnen (DS 952) nein Werkstoff- Leistungsblatt nein					
Halbzeugart		Zustand ?)		Dicke Dmr. Wanddicke mm		Zugfest. R <sub>m</sub> N/mm <sup>2</sup>		0,2-Grenze R <sub>p0,2</sub> N/mm <sup>2</sup>		Bruch- dehnung A <sub>5</sub> %		Härte HB		<b>Allgemeine Eigenschaften<sup>1)</sup></b>	
Stange DIN 1747 T. 1		F37 .51 F34 .51		50 50 80 80		370 340		250 220		7 7		100 90		<b>Beständigkeit gegen:</b> Meerwasser 5 Witterung 5	
Rohr DIN 1746 T. 1		F37 .51 F36 .51		≤ 6 6 bis 20		370 360		250 230		12 10		100 100		<b>Wärmumformbarkeit<sup>1)</sup></b> Strangpressen 4	
<sup>?)</sup> Zustands-Anhängezahl an Werkstoff-Nr. nach DIN 17007 Teil 4 (s. Tafel 1, Seite 1) <sup>?)</sup> Elektrisch leitende Anschlüsse durch Schutzgasschweißen mit AlSi-Zusatz herstellbar <sup>4)</sup> Verschleißfeste Oberflächen durch Hartanodisieren (Sonderverfahren), Galvanisieren, chemisches Abscheiden oder Spritzbeschichten möglich										<b>Kaltumformbarkeit<sup>1)</sup></b> Stauchen, Nieten, Biegen 4					
<b>Hauptanwendungen</b> Spanend bearbeitete Teile für Maschinenbau im weitesten Sinne										<b>Schweißbarkeit<sup>1)?)</sup></b> Schmelzschweißen ng Widerstands- Punktschweißen ng					
<b>Hinweis</b> Bei Halbzeug größerer Querschnittsabmessungen (über den genormten Durchmesser- bzw. Dickenbereich hinaus) kann durch nicht rotationssymmetrisches oder einseitiges Bearbeiten Verzug infolge Eigenspannungen auftreten. Spannungsabbau ohne deutliche Eigenschaftsänderung ist durch Glühen 200 °C/2 h möglich (darf nur einmal ausgeführt werden!)										<b>Lötbarkeit<sup>1)</sup></b> Hartlöten ng Weichlöten (Reib- löten) 4					
<b>Physikalische Eigenschaften</b>										<b>Spanbarkeit<sup>1)</sup></b> Zustand ausgehärtet 1					
Dichte (spez. Gew.) g/cm <sup>3</sup>		Erstarrungs- bereich °C		Elektrische Leitfähigkeit m/Ω mm <sup>2</sup>		Wärmeleit- fähigkeit W/m · K		Wärmeaus- dehnungszahl 10 <sup>-6</sup> · 1/K		Elastizitäts- modul N/mm <sup>2</sup>		<sup>1)</sup> Die Bewertungen geben eine Reihenfolge innerhalb der Aluminiumwerkstoffe mit von 1 nach 5 fallender Tendenz. nz = nicht zutreffend ng = nicht geeignet			
2,85		507-650		18-22		130-160		23		~70 000					

## Camera Views



## Manufactured Specimen



# Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich die Masterarbeit selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und noch nicht veröffentlicht.

---

Ort, Datum

---

Felix Schleifer