SILESIAN UNIVERSITY OF TECHNOLOGY

FACULTY OF MECHANICAL ENGINEERING

DEPARTMENT OF COMPUTATIONAL MECHANICS AND ENGINEERING

Doctoral Dissertation

M. Sc. Eng. Przemysław Sebastjan

OPTIMIZATION OF AUTOMOTIVE SUSPENSION COMPONENTS WITH CONSIDERATION OF THEIR UNSTABLE BEHAVIOR

Supervisor:

Wacław Kuś, BEng, PhD, DSc

Gliwice, 2023

List of Contents

List of symbols and abbreviations....4
1. Introduction....7
1.1. Prologue....7
1.2. Dissertation motivation, aims, range, and thesis....7
1.3. Plan of the optimization methodology implementation....9
1.4. Dissertation structure....9
2. The design, exploitation, and numerical modeling of shock absorbers....11
2.1. Automotive shock absorber work principle....11
2.1.1. The design and operation of the shock absorber....11
2.1.2. The stability of the shock absorber in compression....15
2.2. Shock absorber discrete analysis using the finite element method....20
2.2.1. Displacement formulation of the Finite Element Method....20
2.2.2. Shape functions....24
2.2.3. Finite element method for nonlinear problems....26
3. Optimization and metamodeling methods....30
3.1. Optimization....30
3.1.1. Gradient-based optimization....33
3.1.2. Evolutionary optimization....36
3.1.3. Topology optimization....41
3.2. Metamodeling....45
3.2.1. Sampling plans....45
3.2.2. Response Surface Method (RSM)....49
3.2.3. The Kriging....50
3.2.4. The Artificial Neural Networks....53
4. Hybrid optimization....58
4.1. Optimization problem formulation....58
4.2. Interpolation schemes....65
4.3. Manufacturing constraints....69
5. Hybrid optimization parameters tuning....72

5.1. Metamodel of the FEM simulation 74
5.2. Hybrid optimization algorithm parameters definition 78
5.3. Hybrid optimization algorithm parameters tuning 79
6. Shock absorber optimization methodology – implementation examples 87
6.1. Optimization of the passive shock absorber 87
6.1.1. The passive shock absorber model 87
6.1.2. Topology optimization - spatial LSM 87
6.1.3. Topology optimization - projection method 90
6.1.1. Verification of results 92
6.2. Optimization of the semi-active shock absorber 95
6.2.1. The semi-active shock absorber model 95
6.2.2. Topology optimization - spatial LSM 96
6.2.3. Topology optimization - projection method 97
6.2.4. Verification of results 100
6.3. Results summary 103
7. Summary 105
7.1. Conclusions 105
7.2. Industrial implementation of optimization methodology 110
7.3. Future tasks 111
References 112
Abstract 121
Streszczenie 122

List of symbols and abbreviations

m_s – sprung mass

m_n – unsprung mass

m – the mass of the optimized part

c_s – damping coefficient of the shock absorber

c_t – damping coefficient of the tire

k_s – stiffness of the shock absorber (or spring)

k_t – stiffness of the tire

F_{BPD} – force at the beginning of plastic deformation

F_{max} – the force that may be transmitted by the system without losing stability

F_{limit} – the total axial force required by the car manufacturer that specifies the limit below which the loss of stability must not occur

Ω – arbitrary body with a certain area

Γ – the boundary of the body Ω

σ – stress

σ_0 – yield stress

ε - strain

u – nodal displacement

b – body (volumetric) force

t – surface traction

f – nodal force

E – Young modulus

ν – Poisson coefficient

δ_{ij} – Kronecker Delta

C_{ijkl} – stiffness tensor component

ε_{ij} – strain tensor component

σ_{ij} – stress tensor component

N – shape function

S – differential operator

$\mathbf{I}$ – identity matrix

$\mathbf{D}$ – elasticity matrix

$\mathbf{K}$ – stiffness matrix

$\mathbf{K_T}$ – tangent stiffness matrix

ξ, η, ζ – local system coordinates of the finite element

$\mathbf{\Psi}(\mathbf{u})$ – vector of force residuals

x_i – i-th coordinate of the vector $\boldsymbol{x}$, $\boldsymbol{i} \in [1, \ldots, k]$

$x^{(i)}$– i-th training point or sample, $\boldsymbol{i} \in [1, \ldots, n]$

$g(\mathbf{x})$ – inequality constraint

$h(\mathbf{x})$ – equality constraint

λ_L, μ_L – Lagrange multipliers

$S(\mathbf{x})$ – penalty associated with the design vector $\boldsymbol{x}$

v_p - the difference between the calculated constraint function value and the required constraint value

P_b – penalty base

P_m – penalty multiplier

P_e – penalty exponent

$\nabla f(\mathbf{x})$ – gradient of the function $f(\mathbf{x})$

$\mathbf{H}(\mathbf{x})$ – Hessian matrix

$\mathbf{d}^{(i)}$ – vector of local search direction in i-th iteration

$n_{GA_{iter}}$ – number of total objective function calls *(genetic algorithm)*

n_{GA_ind} – number of individuals *(genetic algorithm)*

n_{GA_gener} – number of generations *(genetic algorithm)*

n_{GA_ε} – convergence limit *(genetic algorithm)*

p_{GA_co} – crossover rate *(genetic algorithm)*

p_{GA_mut} – mutation rate *(genetic algorithm)*

$n_{GA_{isl}}$ – number of autonomous populations (*multi-island genetic algorithm)*

n_{GA_migint} – interval of migration *(multi-island genetic algorithm)*

p_{GA_migr} – the probability of migration *(multi-island genetic algorithm)*

R_{pop} – population ratio (*genetic algorithm*)

λ_{ES} – number of mutants within a single generation (*evolution strategy*)

ϕ – auxiliary field (*level-set method*)

t_{LSM} – threshold value (*level-set method*)

s_i – i-th scaling factor associated with the height of the RBF (*level-set method*)

O_{pm}, C_{pm} – offset and constant values used to scale RBF (*projection method*)

FE_{size} – characteristic length of the finite element

$\hat{y}(\mathbf{x})$ – approximation of function $\boldsymbol{y(x)}$

$\boldsymbol{\beta}$ – vector of polynomial coefficients (*response surface method*)

θ, p – parameters scaling the Gaussian basis function (*Kriging method*)

$\boldsymbol{\Psi_{krig}}$ – correlation matrix (*Kriging method*)

$\widetilde{\boldsymbol{\Psi}}$ – augmented correlation matrix (*Kriging method*)

μ, σ_{std} – parameters of Gaussian process – mean and standard deviation (*Kriging method*)

φ – activation function (*artificial neural networks*)

w_j – weight associated with the j-th neural connection (*artificial neural networks*)

E_{MSE} – loss function based on mean squared error (*artificial neural network*)

η_1, η_2 – learning rate and momentum parameters (*artificial neural network*)

R^2 – coefficient of determination (*metamodeling*)

1. Introduction

The following dissertation is a culmination of the four-year implementation doctorate project done in cooperation with one of the leading manufacturers of automotive shock absorbers. All of the examples shown within the dissertation are associated with the products created by the company, with real-world requirements regarding the design, its strength and structural behavior.

1.1. Prologue

Hydraulic shock absorber technology is growing rapidly since the beginning of the XXth century when the idea of the damper with a viscous fluid was first introduced. The role of the shock absorbers grew constantly through the years, ranging from a simple damping unit to a suspension strut, which not only counteracts the oscillations of the vehicle body and wheels but also assures the required rigidity, comfort, and handling properties required by the driver. Modern trends in the design of vehicles force suspension manufacturers to seek for mass-optimized solutions, with the lightweight target being one of the widely addressed topics in scientific literature. This can be achieved in several ways [125], i.e. by means of advanced manufacturing technologies [87], lightweight materials [51], or structural optimization [112]. The ongoing transformation from vehicles internal combustion engines (ICE) to electric vehicles (EV) increases the demand for weight reduction, as firstly, the mass of the batteries exceeds the mass of the ICE, and secondly, one of the EVs weaknesses - the range - is dependent on the total vehicle mass [23], [24]. Although most of the works are devoted to the primary function of the shock absorber, i.e. its damping [19] or the noise, vibration, and harshness (NVH) performance [121], the structural strength of the shock absorber and its components is also of interest [57], as it allows not only for the weight reduction but also for mitigating the performance issues associated with the inertia of unsprung mass and decrease in the material or manufacturing cost.

1.2. Dissertation motivation, aims, range, and thesis

The motivation for this dissertation is the need to meet the automotive market demands for lightweight shock absorber designs that are able to fulfill intricate structural requirements. Among those requirements, the constraint that is not often considered during the design phase (and design optimization) is the suspension stability, i.e. its ability to resist certain load levels without buckling or failure of particular components. Such a problem is associated with the fact, that a suspension

system (like the shock absorber) may not be able to withstand the desired loads, even if each of the constituent components subjected isolatedly to that load is sufficiently strong. In automotive chassis designs, the shock absorbers usually play the additional role of wheel travel limiters, therefore they are subjected to extreme load levels associated with either rebound or compression events. The effect of the latter type of loading on the shock absorbers is usually addressed with linear buckling analysis incorporating the finite element method [28], [91], from which the eigenvalues associated with specific buckling modes are obtained. Unfortunately, the application of that theory yields overly conservative results, leading to a nonoptimal weight of the part. That is why accounting for the actual load limits during the design optimization is crucial, even if it requires significantly higher computational cost compared to the linear analyses [66]. The design optimization itself must also incorporate the manufacturing requirements which are specific to a certain application. In this dissertation, the shock absorber forged brackets are of interest, therefore, on the optimization level, the manufacturability of those components must be assured. Summing up, the doctoral project requires the following sub-steps to be taken to achieve the goal:

1. Choosing the appropriate method of numerical analysis of the shock absorber subjected to extreme compression loads causing the loss of stability at the system level, incorporating effects of large deformations, material plasticity, and contact interactions
2. Definition of design topology and shape modification method that allows to arbitrarily modify the shock absorber design without violating the manufacturing constraints, and overcoming the drawbacks of the state-of-the-art optimization methods
3. Creation of an efficient optimization algorithm to couple simulations from step 1 with updated designs generated by the method from step 2 without any prior knowledge regarding the resultant shape of the component
4. Tune the optimization algorithm parameters to allow for maximum reduction of the shock absorber mass within the available computational timeframe
5. Verification of the proposed optimization method on the actual industrial examples of shock absorber constrained mass minimization

To avoid excessive complexity of the doctoral project and the dissertation, all of the work concerned in this dissertation is limited to telescopic automotive shock absorber designs with conventional (metallic) materials, conventional chassis technologies (double wishbone, multilink suspension, etc.), and with loads and boundary conditions reflecting the physical and virtual validation of the isolated shock absorbers (even though their actual vehicle-based working conditions may be different).

Based on the literature studies, there is no off-the-shelf solution that can be easily adapted to the specific problem of structural optimization with stability constraints considered in this dissertation. Therefore, the following thesis is formulated:

It is possible to formulate an optimization method and algorithm that allows for mass minimization of the shock absorber taking into consideration the stability of the whole system under extreme compression loads and its manufacturing requirements.

1.3. Plan of the optimization methodology implementation

The planned implementation of this doctoral project is the methodology of structural optimization that allows for the minimization of shock absorber mass, taking into account the system (chassis) stability under extreme compression loading and the manufacturability of the obtained components topology. The motivation for such implementation is the increase in the innovation and the competitiveness of the company (the graduate student employer), shorter prototyping phase, and solution of the drawbacks associated with optimization methods used currently within the company.

To assure that the presented methodology can be used within the company without additional investments, all of the required software is either already available (Abaqus [129], Isight [130], MiniTab [131]) or is freely available online, like the programming language Python [132] and site packages associated with it, such as pyTorch [133] or scikit-learn [134].

1.4. Dissertation structure

The dissertation consists of 7 chapters and it is written in a coherent way, as described briefly in the paragraphs below. Only the fundamental theoretical background needed to understand the work is included in this dissertation.

Chapter 1 (this section) provides the motivation for this doctoral project, aims, limitations, and goals.

Chapter 2 consists of the description of shock absorber design and work principles, problems associated with its operation, and stability considerations. Furtherly, the description of the finite element method is given with a focus on nonlinear aspects of simulations used within the dissertation for the assessment of shock absorbers structural behavior.

Chapter 3 provides an introduction to the field of optimization, with the fundamental description of algorithms utilized within the work done in this doctoral

project. The second part of this chapter is devoted to the field of metamodeling (surrogate modeling) that is (together with optimization theory) used excessively in subsequent chapters.

Chapter 4 gives a detailed description of the optimization method that was used to meet the objectives of this doctoral project, with references to methods described in Chapter 3. The complete description of the topology optimization method and the hybrid optimization algorithm are given with reference to the considered shock absorber FEM analysis, as introduced in Chapter 2.

Chapter 5 deals with the process of tuning the optimization algorithm parameters using surrogate modeling described in Chapter 3. This chapter presents the steps taken to increase the efficiency of the proposed optimization method that was described in Chapter 4 (and partially in Chapter 2).

Chapter 6 shows examples of the proposed methodology implementation for two different designs of automotive shock absorbers. The detailed verification of the structural behavior of the optimized shock absorbers is described, with emphasis on the stability and failure modes of the complete system.

Chapter 7 provides a summary of all the work done within the doctoral project with a short discussion about future tasks and opportunities.

2. The design, exploitation, and numerical modeling of shock absorbers

Shock absorbers are present in every road vehicle, from small passenger cars to trucks. They are used not only in chassis design, even though this is their primary environment, but also in seats, cabins, or steering systems. Depending on the application, their design may differ, the same as the load levels, design requirements, or manufacturing technologies. Still, modern shock absorbers damp the vibrations and oscillations through the usage of viscous fluid, i.e. hydraulic oil, even though different designs or valve technologies are incorporated.

2.1. Automotive shock absorber work principle

The following subsections provide an introduction to the shock absorbers designs, their function in the vehicle suspension, force flow inside the complete unit (shock absorber or modular assembly), and bring insight regarding sources of unstable behavior.

2.1.1. The design and operation of the shock absorber

The role of the shock absorber is quite broad, as the shock absorber itself may be an assembly of different systems, or its design may vary accordingly to the allowable packaging in the chassis. Still, the main set of requirements may be summarized in the following way [42]:

- The shock absorber should damp any oscillations or vibrations coming from the road that are transmitted to the vehicle body
- The shock absorber should reduce or eliminate the wheel oscillations and vibrations that are induced by the road conditions to assure constant contact between the tire and the road surface

The first point is connected to the feeling of the driver or passengers once the vehicle is subjected to input from the road. Different damping characteristics may be required for different vehicle types, e.g. comfort is a target for a family car while being not necessarily the concern for the sports car, where performance is the ultimate goal. The second point, on the other hand, treats the vehicle performance, i.e. its ability to accelerate, steer, or brake. Usually, this point is dominant in racing applications. In most automotive vehicles, a compromise must be found between comfort and performance.

Depending on the type of shock absorber, there are other requirements associated with its operation. The points above are mostly associated with the damping function (hence shock absorbers are commonly referred to as *dampers*). However, a vast part of the shock absorber designs is an assembly of a damper with the suspension spring. With the suspension spring being integrated into the shock absorber design, the unit forms a response for both – velocity and displacement excitation, as per (2.1) and (2.2), as shown schematically in Figure 2.1, together with the wheel (rim and tire).

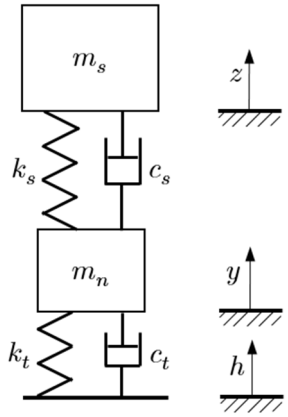

Figure 2.1. The quarter vehicle chassis model [93]

The shock absorber force response is obtained utilizing the differential equilibrium equations of the following form:

$$m_s \frac{d^2 z}{dt^2} = -c_s \left(\frac{dz}{dt} - \frac{dy}{dt}\right) - k_s(z - y) \tag{2.1}$$

$$m_n \frac{d^2 y}{dt^2} = -c_s \left(\frac{dy}{dt} - \frac{dz}{dt}\right) - k_s(y - z) - c_t \left(\frac{dy}{dt} - \frac{dh}{dt}\right) - k_t(y - h) \tag{2.2}$$

where m_s is the sprung mass, m_n is the unsprung mass, c_s and c_t are damping coefficients of the shock absorber and tire respectively, k_s and k_t are stiffness coefficients of the shock absorber and tire respectively. The h, y, and z represent the road input (excitation), unsprung mass, and sprung mass displacements respectively. The displacements of the vehicle body m_s and wheel center m_n is coupled through the springs and dampers. Those equations are usually transformed and solved into the matrix form in the time domain [37], as in real conditions, coefficients c_s, c_t, k_s, k_t are not constant, and they are functions of displacement, velocity, or frequency [111]. Another, obvious method for obtaining the shock absorber force response is to perform a physical measurement, however, this method is usually not available in the prototype phase (which is dominated by the virtual prototypes and computer simulations).

The automotive springs usually take a form of a helical coil design or an air spring. Even though those are completely different designs, the working principle and the roles are the same. On top of that, shock absorbers serve as travel-limiting devices,

preventing the wheel to hit the car body or other suspension parts. This makes the design of a shock absorber much more complicated, and it is associated with the increased level of forces that must be transferred through the shock absorber. A scheme representing different designs of a shock absorber is shown in Figure 2.2. Even though different types of shock absorbers are shown, they share almost the same type of structural components, i.e. the upper attachment (top mount), the rod, the tubes (depending on the application, from one to three), and the lower attachment which is some form of the bracket, either a loop or a stamped or forged component.

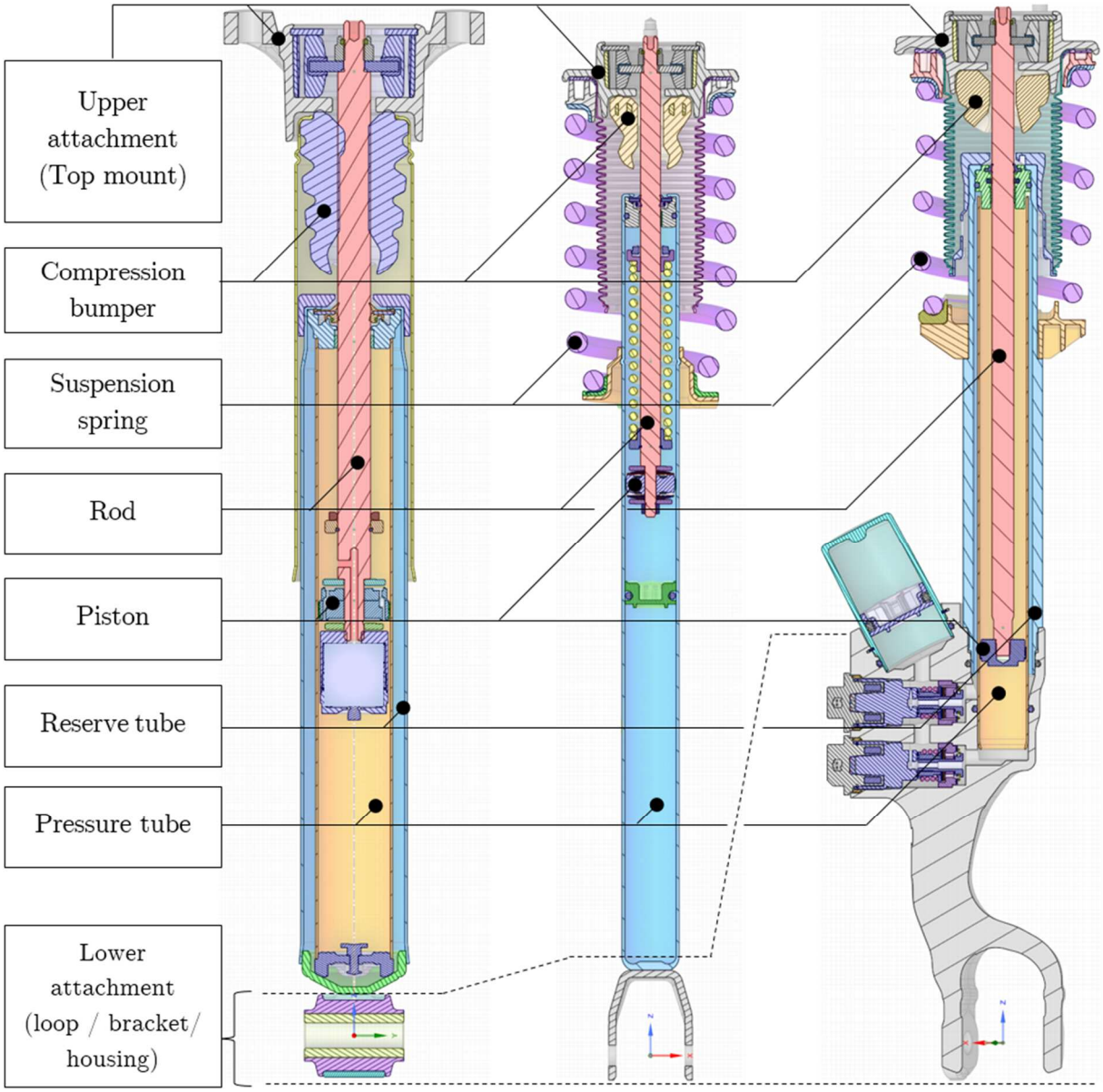

Figure 2.2. Structural components of the automotive shock absorber

All of the aforementioned components play a significant role in the load transfer, even though they participate in varying degrees and are responsible for different actions. The roles of the components may be listed in the following way:

- **The upper attachment** (usually a top mount or a loop) is a bridge between the vehicle body and the shock absorber. It isolates the rod vibrations from the body, prevents excessive rotation of the rod, and fixes the position of the shock absorber during its work
- **The compression bumper** is one of the travel limiting features (serving the same purpose as a rebound bumper, but for a different stroke direction). It prevents the wheel from colliding with the vehicle body or other suspension components. It is activated once a certain level of the compression stroke is reached, and there is no further possibility of stopping this movement through spring or damping
- **The suspension spring** supports the weight of the vehicle (vehicle corner weight). Depending on the chassis design, it may be subjected to forces several times higher than the weight of the quarter, due to the motion ratio between the wheel and shock absorber. Spring preload and rate play a significant role in absorbing the shock input from the road surface. It also prevents excessive body roll (which in general is a primary function of the stabilizer bar)
- The main role of **the rod** is to guide the piston, which is usually the main valve of the shock absorber system. In MacPherson struts, the rod must counteract the side forces and bending moments generated by this type of chassis. However, in other chassis designs, like multilink or double wishbone suspensions, the rod may be subjected to side forces due to one of the following reasons: stabilizer bar loads which are inclined versus the shock absorber axis, suspension spring piercing points misalignment or lateral deformation of the system subjected to excessive compressive loading, i.e. buckling
- **The piston** is, together with the rod guide, a contact point between the rod (the part connected to the vehicle body) and the structure of the shock absorber (connected with the wheel by a spindle or knuckle). In passive shock absorbers, pistons are used as a part of the valve system
- **The reserve tube** serves two major roles – first is to make room for the gas (pure nitrogen or air without oxygen and moisture) which compensates for the volume of the rod being displaced into the shock absorber body. The second role is the structural support of side and axial loads, especially those coming from the suspension spring and the bumpers
- **The pressure tube** counteracts the pressure of the oil being displaced or compressed by the rod and valves. Also, it guides the piston (also it guides the floating piston in a single tube design)
- **Lower attachment** (usually a loop, bracket, or housing) connects the shock absorber to the chassis (knuckle, spindle). In semi-active shock absorbers, it serves

as a valve mounting point. The role of the lower bracket is also to assure the stability and lateral stiffness of the shock absorber during excessive operating loads

2.1.2. The stability of the shock absorber in compression

One of the primary roles of the shock absorber is to transmit the whole collection of axial forces from the road surface to the vehicle body (and vice versa). Those axial forces may act only in two directions: either in the rebound stroke, where the wheel is pushed out of the vehicle body, or in the compression stroke (referred to as the *jounce* stroke), in which the wheel is pushed towards the vehicle body.

The first case, i.e. rebound stroke, is rather limited in force, as the total magnitude of the load is generated as a result of energy accumulated in the compressed suspension spring, jounce bumper, and the gas from inside the shock absorber. This total energy may be foreseen during the shock absorber development phase, and the rebound bumper characteristic may be adjusted to minimize the peak force during the short period of wheel sudden deceleration.

On the other hand, the compression stroke is barely limited in the maximum force that may be generated on the shock absorber, as there always may be an additional factor increasing the total load magnitude, e.g. higher vehicle speed or mass, bigger obstacle, lower tire profile, etc. Therefore, the peak compression forces are significantly higher compared to the values achieved for the rebound strokes. On top of that, any additional load applied to the chassis is usually transferred mainly by the jounce bumper to the shock absorber, as this component tends to have an asymptotic load-deflection characteristic, which causes it to transmit a significant portion of load once it is subjected to additional, minor compressive displacement while being in a pre-compressed state. An example of the force overview inside the shock absorber during the 'drive through pothole' event is presented in Figure 2.3, clearly showing the unequal load distribution between components. The red curve represents the total axial force measured at the shock absorber, while the green, blue, and violet curves represent the spring load, damping load, and the jounce bumper load respectively. For the considered example, at the beginning of the event, i.e. before the pothole strike, there is no force transferred by the jounce bumper, as the vehicle is suspended by the spring force and minor oscillations are countered by the damping force. During the peak loading, at the strike, when the suspension is fully compressed, 86% of the total load is transferred by the jounce bumper, 13% by the spring load, and only 1% by the damping load. The increase in the spring load is almost negligible, compared to the delta in jounce bumper load before and during the event. The total increase in the axial force transferred by the shock absorber before and during the strike is over 700%.

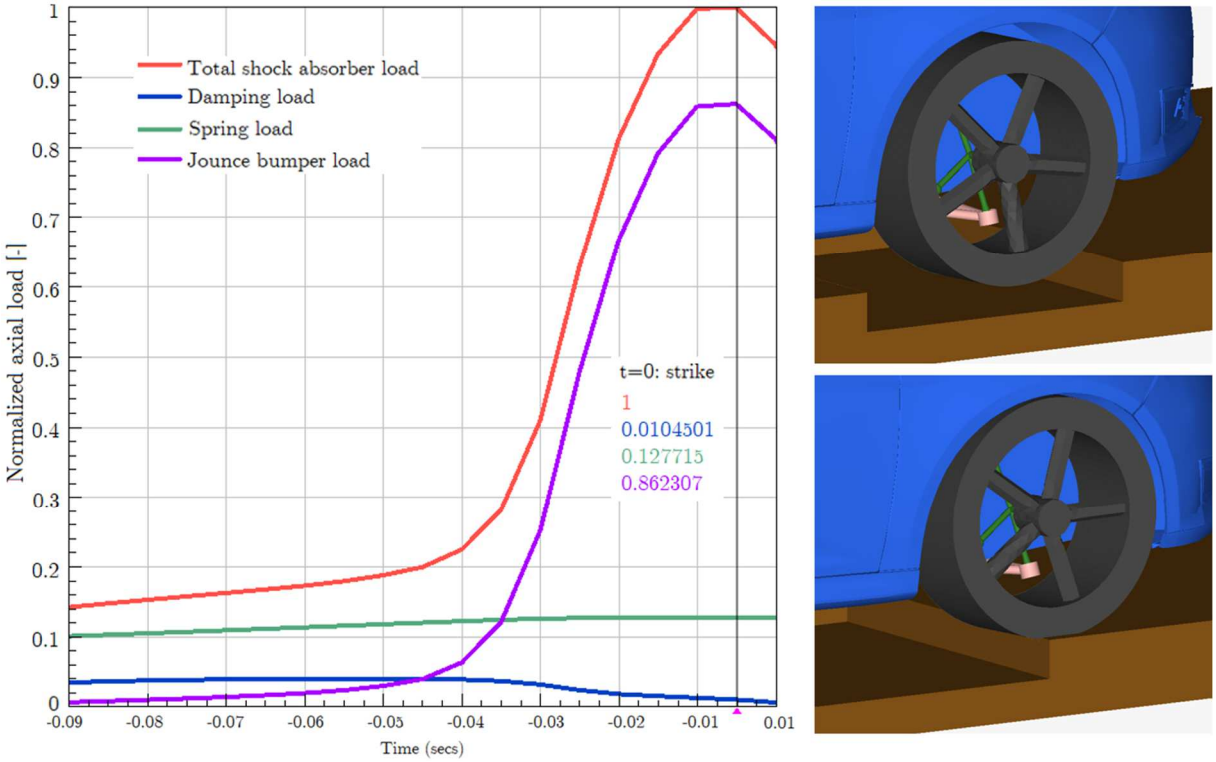

Figure 2.3. Force distribution inside the shock absorber during the 'drive through pothole' event

Even though the strength of the material is usually higher for compression rather than tensile stress, the dangerous phenomenon of lateral deflection and/or buckling may occur, as the result of the compression forces. Especially, the lateral deflection during the jounce phase is something that should be avoided, as once the whole structure starts to bend or deflect laterally, there are additional bending moments and side loads generated in the system, as the load application points - like the spring seat or jounce bumper cap - are shifted away from the original shock absorber axis. Such a case is shown schematically in Figure 2.4.

The translation of the load application point is visible clearly on the deformed model of the shock absorber. This deflection together with the axial force creates a bending moment due to load application point and structure supports misalignment. The sources of the lateral deformation may include:

- Bending moments generated due to suspension **bushing deflection** and/or rotation
- Suspension **spring piercing points** misalignment
- **Unsymmetric cross-section** of the structural components, which is generally observed for the lower brackets and housings

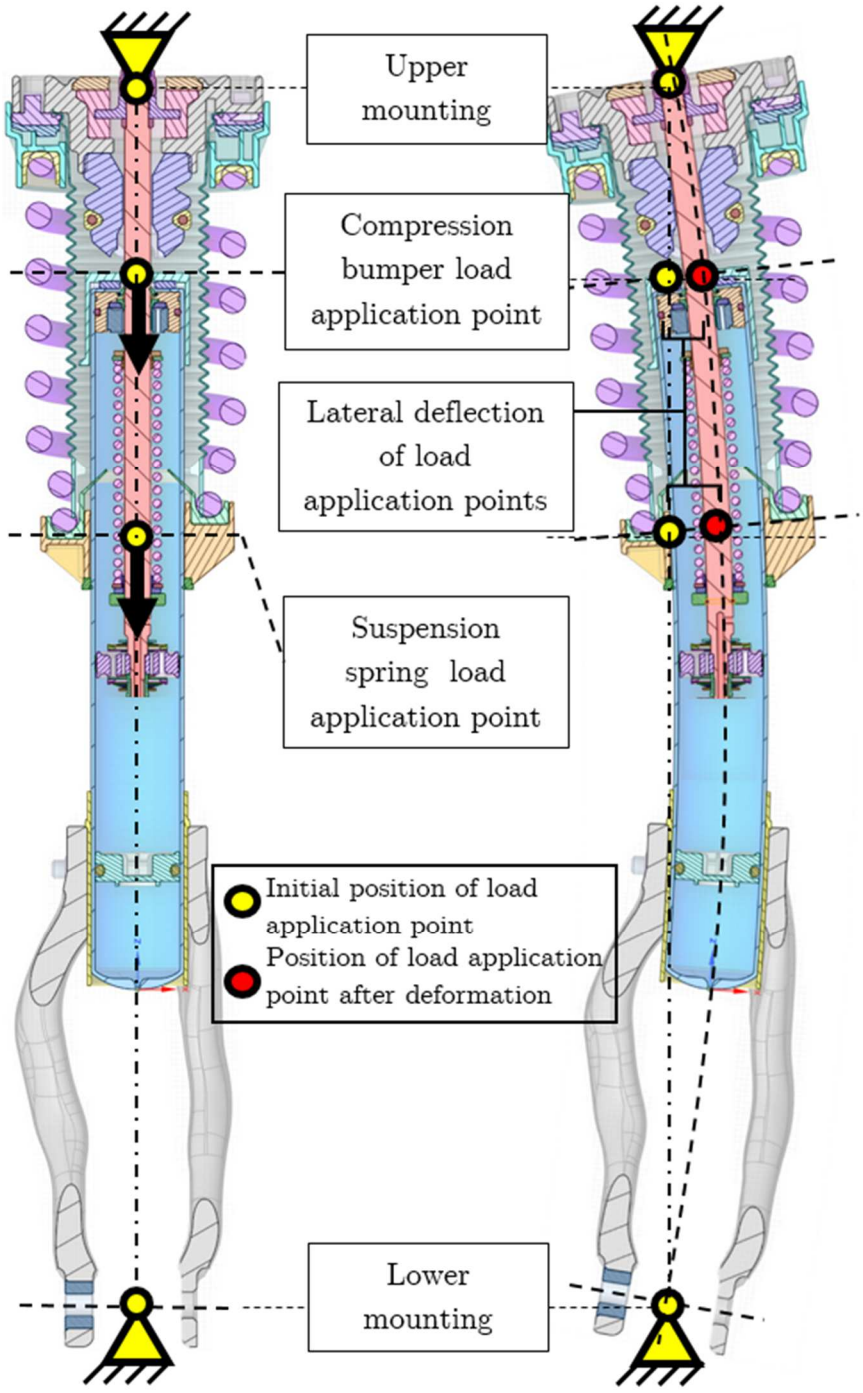

Figure 2.4. The shock absorber: a) undeformed model, b) model subjected to lateral deformation due to axial forces

The bushing behavior is usually adjusted by the vehicle manufacturer, to match the noise, vibrations, and harshness (NVH) requirements, as well as the anticipated levels of elastic deformation due to chassis operation.

Suspension spring piercing points are usually production flaws (despite MacPherson struts, in which the piercing points misalignment is induced intentionally for the purpose of minimizing the contact and friction forces inside the shock absorber), so they are not influenced by the shock absorber or vehicle manufacturer.

The process of designing the shape of the brackets, or housing, is usually the responsibility of the shock absorber manufacturer. As this part may play a significant role in inducing (or countering) the transverse deformation, it is of great importance to design such parts with caution. Generally, it is required to verify the influence of the lower brackets on the strength of the shock absorber with the system level analysis (shock absorber level) rather than as an isolated component analysis. The reason for that is the compliance (both axial and lateral) of the lower attachment that affects the stress distribution over all other structural components - especially the rod and the tubes – which then may act as potential failure modes of the system.

The failure mode is considered as a component that fails first under the given chassis loads. Failure may be related to either plastic deformation, crack, or separation of the part. As the shock absorbers are generally slender, they rarely achieve material failure due to purely compressive stresses, but rather due to transverse deformation of the whole system instead. Therefore it is so important to predict the system behavior and take into account its stability. For many slender constructions under compressive forces, the transition between the linear state of deformation to the collapse is almost instantaneous, as shown in Figure 2.5, in which the shock absorber is loaded incrementally until reaching the point of its stable behavior, leading to excessive (plastic) deformation and degradation of further load bearing capacity.

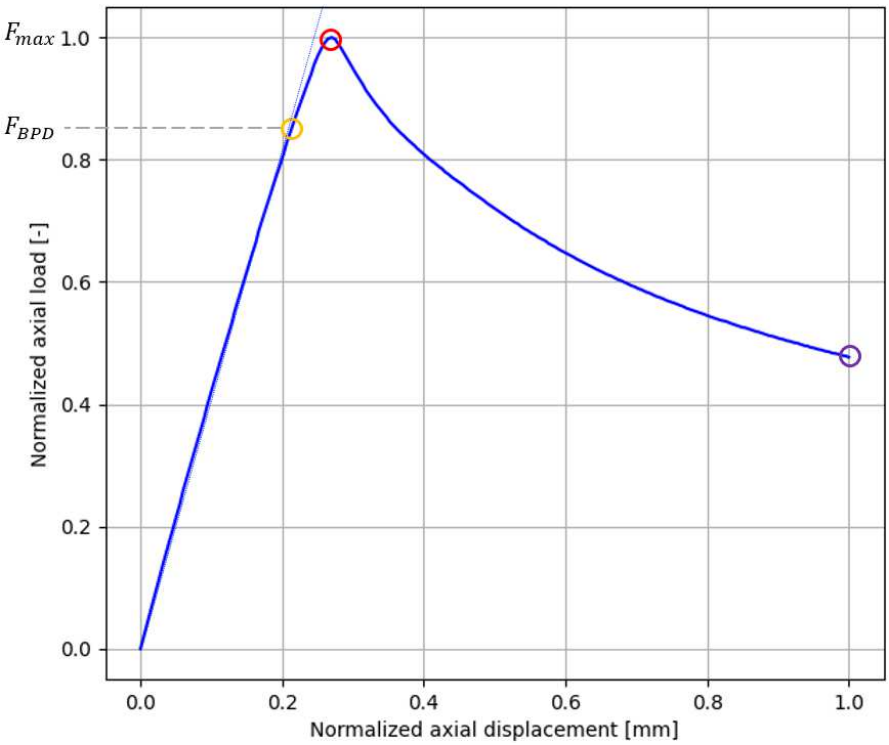

Figure 2.5. Compressive load-deflection curve of the shock absorber

There are three points marked on the presented load-deflection graph. The yellow point represents the F_{BPD} which is a force at the beginning of plastic

deformation, the red point represents the maximum force F_{max} that may be transmitted by the system without losing stability, and the violet point is the arbitrary post-buckling deformation mode of the whole system ($F_{deformed}$). They are all represented by different levels of plastic strain and deformation, as shown in Figure 2.6.

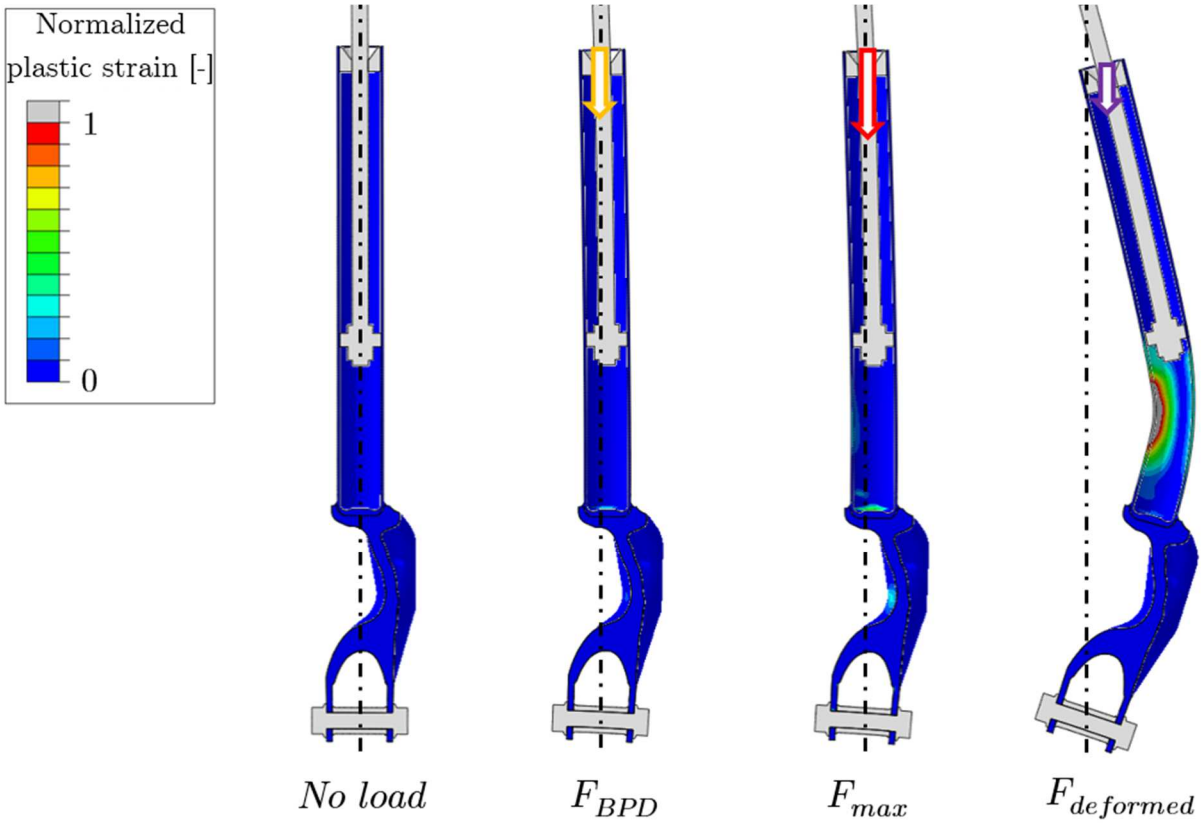

Figure 2.6. Shock absorber with different levels of deformation and normalized equivalent plastic strain (1 – engineering strain at the ultimate tensile strength)

The shock absorber is no longer fulfilling its role as the wheel travel limiter once the F_{max} is reached, as the subsequent deformation is achieved without increasing the operational load (like the inertia of the vehicle body). Therefore predicting the F_{max} is a crucial part of design development.

2.2. Shock absorber discrete analysis using the finite element method

To verify the structural behavior of the shock absorber (or any arbitrary body) subjected to a combination of assembly and axial compression loads, computer simulations may be utilized. This chapter provides a brief introduction to the finite element method (FEM), which is one of the most commonly used tools to analyze the intricate problems of structural mechanics in most fields of practical engineering, such as civil, automotive, space, or aerospace engineering.

2.2.1. Displacement formulation of the Finite Element Method

The beginnings of the FEM span to the first half of the 20$^{\text{th}}$ century, with pioneering works such as [22], [48]. Rapid development occurred during the second half of that century, with works such as [126]–[128]. The classical analysis of continuous mechanics problems requires the determination of the stress, strain, and displacement fields that satisfy the differential equilibrium equations, and constitutive and geometric relationships at all points in the considered area, including the boundary of the body. Therefore, analytical models are often not available for practical engineering problems. The finite element method is assuming the approximate displacement field, in each of the discretized small domains, called finite elements. The determination of the stress and strain fields is calculated for the finite number of points only and approximated (interpolated) across the whole body. This way the FEM can be adapted to numerous applications, taking into consideration more sophisticated formulations like nonlinear geometrical behavior, anisotropy with inelastic characteristics, or interaction between bodies – contact, both in problems of static and dynamic systems [56], [85].

The derivation of FEM equations for the discrete system is a consequence of the analysis of the principle of virtual work for the continuous body. This body Ω with the boundary Γ is presented in Figure 2.7. The body forces (or volumetric forces) b are applied over the whole volume of the body. The surface forces t_0 are applied on the part of the boundary Γ_t. The prescribed displacements u_0 are specified for the part of the boundary Γ_u. Such a deformable body, which is treated as a part of the continuous medium, stays in equilibrium under the presence of the aforementioned loads. The following equations apply: $\Gamma = \Gamma_t \cup \Gamma_u$ and $\Gamma_t \cup \Gamma_u = \emptyset$

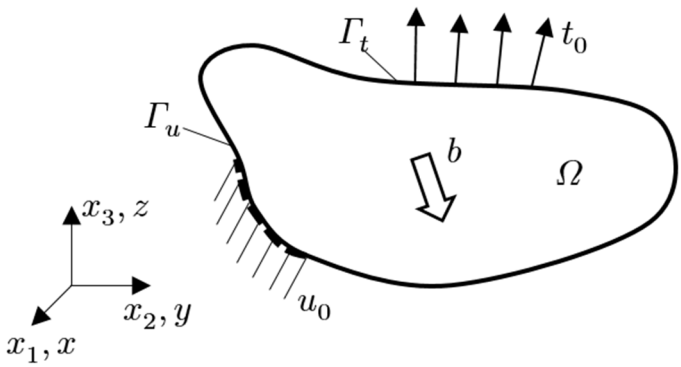

Figure 2.7. Deformable body Ω

The following equilibrium equations may be formulated for the body considered:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0 \text{ within the body } \Omega \tag{2.3}$$

where σ_{ij} is a stress tensor component, such that:

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \tag{2.4}$$

Taking into account the moment equilibrium in a unit body, the stress tensor (2.4) can be represented by a vector (2.5):

$$\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{xz}, \tau_{yz}] \tag{2.5}$$

The boundary conditions for the considered body are given as per (2.6) and (2.7):

$$u_i = u_i^0 \text{ on the boundary } \Gamma_u \tag{2.6}$$

$$\sigma_{ij} n_j = t_i^0 \text{ on the boundary } \Gamma_t \tag{2.7}$$

where n_j is a versor normal to the boundary Γ, and t_i is the surface force vector.

The constitutive equations (Hooke law) that determine the relationship between stress and strain of the body may be formulated (2.8):

$$\sigma_{ij} = \frac{E}{(1+\nu)} \left[\varepsilon_{ij} + \frac{\nu}{(1-2\nu)} \delta_{ij} \varepsilon_{kk} \right], \tag{2.8}$$

or in general form (2.9):

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} \tag{2.9}$$

where E is the Young modulus, ν is the Poisson coefficient, δ_{ij} is the Kronecker Delta, C_{ijkl} is the stiffness tensor, and ε_{ij} is a strain tensor component, such that:

$$\varepsilon = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} = \begin{bmatrix} \varepsilon_x & \frac{\gamma_{xy}}{2} & \frac{\gamma_{xz}}{2} \\ \frac{\gamma_{yx}}{2} & \varepsilon_y & \frac{\gamma_{yz}}{2} \\ \frac{\gamma_{zx}}{2} & \frac{\varepsilon_{zy}}{2} & \varepsilon_z \end{bmatrix} \tag{2.10}$$

Similarly to the stress tensor, the strain tensor can be represented by a vector (2.11):

$$\boldsymbol{\varepsilon} = [\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}] \tag{2.11}$$

The strains shown in (2.11) are partial derivatives of the displacements (2.12):

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.12}$$

The above equations can be used to formulate the weak form of the displacement formulation of the linear elasticity equations:

$$\int_{\Omega} \delta\varepsilon_{ij}\sigma_{ij}\, d\Omega = \int_{\Omega} \delta u_i b_i\, d\Omega + \int_{\Gamma} \delta u_i t_i\, d\Gamma \tag{2.13}$$

where δu_i are the arbitrary displacements and the arbitrary strains have the form as per (2.14):

$$\delta\varepsilon_{ij} = \frac{1}{2}\left(\frac{\delta\partial u_i}{\partial x_j} + \frac{\delta\partial u_j}{\partial x_i}\right) \tag{2.14}$$

Equation (2.13) may be rewritten in the matrix form:

$$\int_{\Omega} \delta\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\, d\Omega = \int_{\Omega} \delta\mathbf{u}^{\mathrm{T}}\mathbf{b}\, d\Omega + \int_{\Gamma} \delta\mathbf{u}^{\mathrm{T}}\mathbf{t}\, d\Gamma \tag{2.15}$$

where the left side of the equation is the work stored in the deformed body as the strain energy, while the right side represents the work generated by the body and traction forces done on the virtual displacements.

The virtual work done by the system may be calculated on the element-basis, instead of dealing with the continuum. Such a procedure requires the continuum to be replaced with a set of discrete finite elements (therefore the process is called discretization). The deformable body in discrete form is presented in Figure 2.8, where triangular elements were used to divide the body area into a set of finite elements.

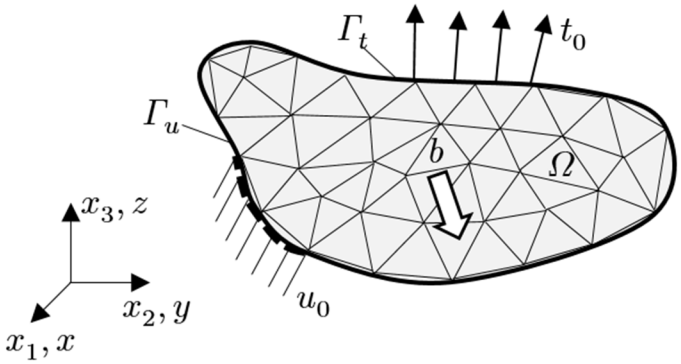

Figure 2.8. Deformable body Ω in discrete form

The displacement and the arbitrary displacement fields are interpolated inside the finite elements, using the shape functions N. The role of the shape function is to determine the unknown displacements inside the finite element, based on the calculated displacements in the element nodes.

$$u(x) = \mathbf{N}\mathbf{u} \tag{2.16}$$

$$\delta u(x) = \mathbf{N}\delta\mathbf{u} \tag{2.17}$$

Equation (2.17) may be used to formulate the arbitrary (virtual) strains in (2.15) using the following form:

$$\delta\varepsilon = \mathbf{S}\delta u(x) = (\mathbf{S}\mathbf{N})\delta u = \mathbf{B}\delta u \tag{2.18}$$

where S is the differential operators' matrix.

The above equation may be used to reformulate the internal virtual work (2.13) of a system in the form of (2.19):

$$\int_{\Omega} \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\, d\Omega = \mathbf{f} \tag{2.19}$$

where $\mathbf{f}$ is the external virtual work:

$$\mathbf{f} = \int_{\Omega} \mathbf{N}^{\mathrm{T}}\mathbf{b}\, d\Omega + \int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{t}\, d\Gamma \tag{2.20}$$

For a simple linear stress-strain relationship, the matrix $\boldsymbol{\sigma}$ may be replaced using (2.21):

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} \tag{2.21}$$

where $\mathbf{D}$ is the elasticity matrix. Also, the strains may be rewritten as (2.22):

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{u} \tag{2.22}$$

The equation (2.19) may be again rewritten, using the equations (2.21) and (2.22) in a new form (2.23):

$$\int_{\Omega} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\, d\Omega \mathbf{u} = \mathbf{f} \tag{2.23}$$

Defining the stiffness matrix $\mathbf{K}$ as (2.24):

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\, d\Omega \tag{2.24}$$

equation (2.22) may be rewritten to the final form of (2.25):

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.25}$$

There are numerous ways to solve the following equations to get the unknown nodal displacements $\mathbf{u}$ taking into account the boundary conditions, such as direct methods: Gauss elimination, LU decomposition or Cholesky factorization, and iterative methods: Gauss-Seidel method or conjugate gradient method [7]. Once the system (2.25) is solved, the calculation of stress inside the body is performed using equations (2.21) and (2.22).

2.2.2. Shape functions

The displacements in any point of the analyzed body are calculated using the shape functions, as shown in (2.16). They enable interpolation of the displacement inside any finite element based only on the information about the displacement vector in the element nodes. The type of the shape function is dependent on the element geometry, i.e. its shape and the number and position of nodes.

There are several requirements that must be met for a given set of shape functions. They must allow for a description of any strain state, as well as rigid body motion. Also, they must ensure continuous displacement interpolation at the common nodes of the finite elements. There are also two properties with regard to the values of the shape functions, i.e. the sum of all shape functions in any point inside the finite element must be equal to one (2.26):

$$\sum_{i=1}^{n} N_i(x, y, z) = 1 \tag{2.26}$$

and at each of the element nodes, the values of the shape functions must be either zero or one (2.27):

$$N_i(x_j, y_j, z_j) = \begin{cases} 1, & where \;\; i = j \\ 0, & where \; i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{2.27}$$

with n representing the number of nodes in the finite element.

Although the order of the polynomial used for a shape function is not directly restricted, in practical applications, the 1st and 2nd order polynomials are sufficient for most purposes. When it comes to the higher order polynomials, the increased computational cost associated with the increased number of degrees of freedom may not be compensated by the increase in accuracy, compared to the 2nd order polynomial interpolation.

For isoparametric finite elements, in which both the coordinates and displacements are using the same order of interpolation, the shape functions are defined in the local coordinate system of undeformed finite element, as per Figure 2.9.

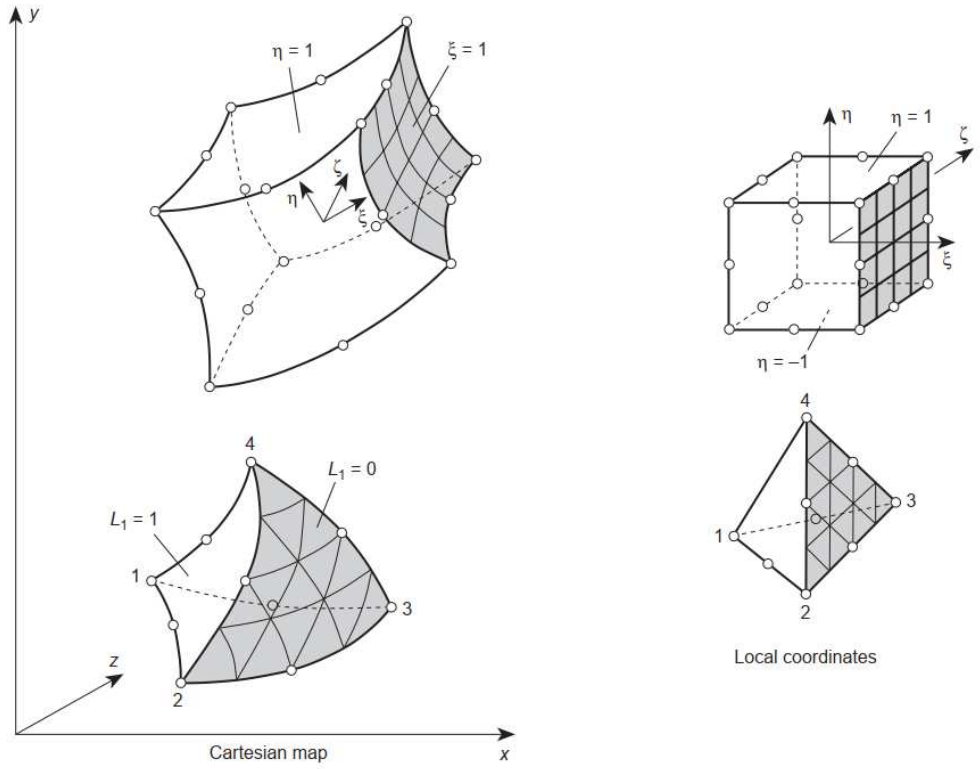

Figure 2.9. Second-order finite element: a) deformed, in a global coordinate system, b) mapped to the local coordinate system [126]

The following formulation of shape functions using 2nd order polynomials is used for the hexahedral elements:

Corner nodes (2.28):

$$N_i = \frac{1}{8}(1+\xi_0)(1+\eta_0)(1+\zeta_0)(\xi_0+\eta_0+\zeta_0-2) \tag{2.28}$$

Exemplary mid-side node (2.29):

$$N_i = \frac{1}{4}(1-\xi^2)(1+\eta_0)(1+\zeta_0) \tag{2.29}$$

for $\xi_i = 0$, $\eta_i = \pm 1$, $\zeta_i = \pm 1$, and where $\xi_0 = \xi\xi_i$, $\eta_0 = \eta\eta_i$, $\zeta_0 = \zeta\zeta_i$.

The shape functions that are used for tetrahedral elements are often expressed using volume coordinates, as per Figure 2.10.

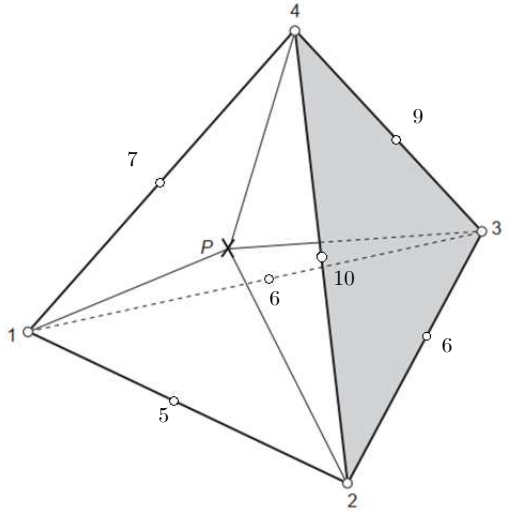

Figure 2.10. Volume coordinates [126]

The volume coordinates are constructed in such a way, that they are representing the ratio of the volume of the sub-tetrahedron to the volume of the total tetrahedron. The sub-tetrahedron is a tetrahedron, in which one of the vertices was replaced by point P. The exemplary coordinate L_1, which represents the volume ratio between the sub-tetrahedron (having point P instead of vertex 1) to the volume of the total tetrahedron is formulated as in equation (2.30).

$$L_1 = \frac{V_{P234}}{V_{1234}} \tag{2.30}$$

The shape functions for the corner nodes of the tetrahedron (using notation from Figure 2.10) are as per equation (2.31):

$$N_1 = (2L_1 - 1)L_1, \quad etc. \tag{2.31}$$

while for mid-nodes as per (2.32):

$$N_5 = 4L_1L_2, \quad etc. \tag{2.32}$$

2.2.3. Finite element method for nonlinear problems

Three sources of nonlinearity may be encountered in the FEM simulation, including a nonlinear material model, large strains, and contact between bodies. One of the most common sources of material nonlinearity is elastic-plastic material behavior. What is more, the presence of plastic deformation usually involves strains for which the small-strain theory is not sufficient [120].

To include the large strains, the strain formulation from (2.12) is replaced with the formula (2.33), which includes the product of partial derivatives of strains:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right) \tag{2.33}$$

Equation (2.33) is known as the Green-Lagrange strain tensor definition, which may be referred to as the sum of the small strain tensor and the quadratic term of the strain. Such a tensor is commonly used to describe finite deformation, together with the second Piola-Kirchhoff stress tensor. Exemplary elastic-plastic material models, for which the high strains are anticipated, are shown schematically in Figure 2.11:

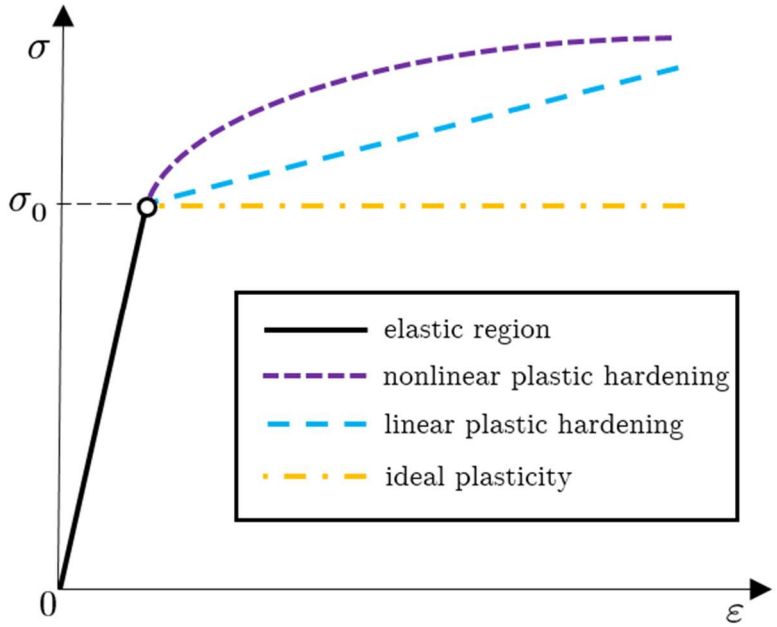

Figure 2.11. Different elastic-plastic material models incorporating plastic deformation

In order to include the nonlinear material constitutive model in the finite element method, the system of equation (2.25) needs to be defined as a function of strain, or more generally, as a function of displacements, from which the strains are derived. As the matrix **D** (2.21) is now dependent on the strains (displacement derivatives), the general form of the canonical equation for a non-linear problem in the finite element method is shown in (2.34):

$$\mathbf{K}(\mathbf{u})\mathbf{u} = \mathbf{f} \tag{2.34}$$

The stiffness matrix $\mathbf{K}(\mathbf{u})$ that is dependent on the current deformation state is referred to as the tangent matrix $\mathbf{K_T}$ with the form (2.35), which is an extension of (2.24):

$$\mathbf{K}_{\boldsymbol{T}} = \int_{\Omega} \mathbf{B}^{\mathbf{T}}\mathbf{D}_{\mathbf{T}}\,\mathbf{B}\,d\Omega \tag{2.35}$$

where $\mathbf{D_T}$ is a tangent elasticity matrix, dependent on the level of strain. Using such notation, the equation (2.34) may be rewritten in incremental form as (2.36):

$$\mathbf{K_T}\mathrm{d}\mathbf{u} = \mathrm{d}\mathbf{f} \tag{2.36}$$

As the forces, strains, and stresses are now considered in the iterative form, the dedicated solution method must be introduced. Commonly, the iterative approach is combined with the incremental increase in the applied loads, especially for highly non-linear systems, like plastic deformation, or second-order deformations. Systems of equations as in (2.36) are commonly solved by the Newton-Raphson or the Modified Newton-Raphson method, both of which being the iterative procedures [120]. The first one is known to have a quadratic rate of convergence, on the cost of the computationally expensive calculation of the $\mathbf{K_T}$ matrix in each iteration, while the second uses the initial $\mathbf{K_T}$ and update it only after finding the equilibrium for the given load increment. The equilibrium equation for the nonlinear problem is given in the following matrix form (2.37):

$$\mathbf{\Psi}(\mathbf{u}) = \mathbf{f} - \mathbf{K}(\mathbf{u}) = \mathbf{0} \tag{2.37}$$

where the $\mathbf{\Psi}(\mathbf{u})$ is a vector of force residuals. The Newton-Raphson method approximates the equation (2.37) using the incremental approach (2.38):

$$\mathbf{\Psi}(\mathbf{u^{i+1}}) \approx \mathbf{\Psi}(\mathbf{u^i}) + \left(\frac{\partial \mathbf{\Psi}}{\partial \boldsymbol{u}}\right)^{\boldsymbol{i}} d\mathbf{u^i} = \mathbf{0} \tag{2.38}$$

where $d\mathbf{u}^i$ is the increment of displacement in the i-th iteration, which updates the vector of nodal displacements as per (2.39). Similarly, the increment of force applied in the i-th iteration is added to the nodal force vector, as per (2.40):

$$\mathbf{u} = d\mathbf{u}^1 + d\mathbf{u}^2 + \cdots + d\mathbf{u}^i + \cdots + d\mathbf{u}^n \tag{2.39}$$

$$\mathbf{f} = d\mathbf{f}^1 + d\mathbf{f}^2 + \cdots + d\mathbf{f}^i + \cdots + d\mathbf{f}^n \tag{2.40}$$

where $\mathbf{u}$ and $\mathbf{f}$ are displacement and external force vectors respectively, and n is the total number of iterations in the Newton-Raphson method. As the $(\frac{\partial \mathbf{\Psi}}{\partial \boldsymbol{u}})$ in (2.38) represents the (negative) tangent stiffness matrix $\mathbf{K_T}$, the equation (2.38) may be rewritten to (2.41):

$$\mathbf{\Psi}(\mathbf{u^i}) = \boldsymbol{K}_{\boldsymbol{T}}^{\boldsymbol{i}} d\mathbf{u^i} \tag{2.41}$$

The Newton-Raphson method and its modified version are depicted in Figure 2.12. The difference between those methods, i.e. $\mathbf{K_T}$ update frequency, is visible in the convergence paths, as the variable or constant tangent moduli in the one-dimensional example shown in Figure 2.12.

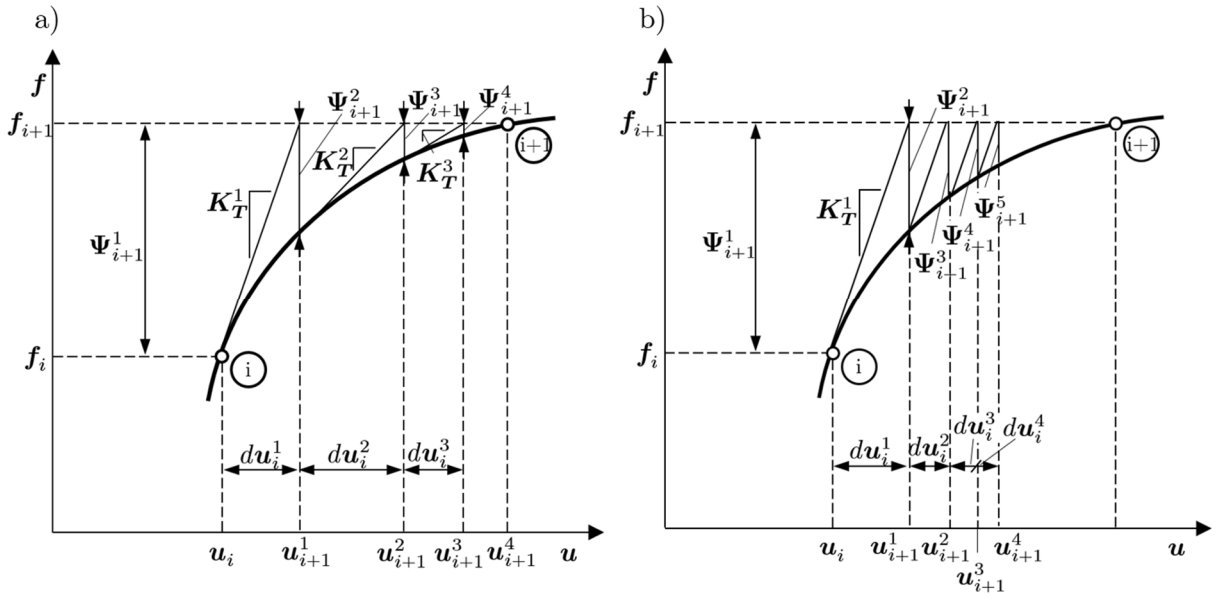

Figure 2.12. Iterative methods: a) Newton-Raphson (NR) method, b) Modified Newton (MN) method [127]

The algorithm of the aforementioned Newton-Raphson method is the following:

1. Calculation of the tangent stiffness matrix $\mathbf{K^i_T}(\mathbf{u}^\mathrm{i})$
2. Solution of the system of equations and calculate $d\mathbf{u}^{\boldsymbol{i+1}}$:
$$\mathbf{K^i_T}(\mathbf{u}^\mathrm{i})d\mathbf{u}^{\mathrm{i+1}} = d\mathbf{\Psi^i} \tag{2.42}$$
3. Calculation of the total displacements:
$$\mathbf{u}^{i+1} = \mathbf{u}^{\boldsymbol{i}} + d\mathbf{u}^i \tag{2.43}$$
4. Update of the tangent stiffness matrix $\boldsymbol{K}^{i+1}_{\boldsymbol{T}}(\boldsymbol{u}^{i+1})$
5. Calculation of the $d\mathbf{\Psi}^{\boldsymbol{i+1}}$

This algorithm is repeated as long, as the $d\mathbf{\Psi}$ is higher than a specified tolerance. If the Modified Newton method is used instead of the Newton-Raphson method, the 4$^{\mathrm{th}}$ step is omitted, as the initial tangent stiffness matrix is used throughout the step (or increment).

3. Optimization and metamodeling methods

Once the behavior of the (shock absorber) mechanical system may be predicted by means of the finite element simulation (or any other technique in general), the iterative modification of the system layout may be introduced to achieve certain goals, such as minimization of the mass of the shock absorber taking into consideration its unstable behavior. The subsections of this chapter provide an introduction to the techniques of mathematical optimization and methods of shape and topology optimization. Also, they present the concept of metamodeling and methods of constructing and training the surrogates, that can be used to mimic the simulated behavior of the shock absorber within a small fraction of a second needed to perform a single FEM-based analysis.

3.1. Optimization

Optimization is a field of mathematics that aims to seek *the best* possible solution under given circumstances, which is either a minimum or maximum value of a certain function or functional [34], [63], [76], [101]. It is hard to point to any particular date which can be considered as the beginning of mathematical optimization, but some authors find Euclid (3^{rd} century BC) to be the first mathematician to consider the optimization problem - seeking the shortest distance between two geometrical features. Then, after almost two millennia, Isaac Newton and Gottfried Wilhelm Leibniz formulated the basis of the calculus of variation, which started a new era in modern mathematics. The first optimization method that was developed, and is still used until this day, was the Gradient Descent method, which is attributed to Augustin-Louis Cauchy (1847). The rapid development of optimization techniques started in the second half of the 20^{th} century, as a result of computational methods development and computing power growth.

The optimization aims to seek the minimum or maximum value of a certain function or functional (3.1). However, any arbitrary function may be multimodal, i.e. it can contain numerous local minima (or maxima) and only one global minimum (or maximum) value within a given domain. To distinguish the latter minima (or maxima) types, both of those types are shown in Figure 3.1, which shows a multimodal function of one variable. Local extrema are marked with green circles, while global are marked with red stars:

$$y = sin(x) + sin\left((\tfrac{10}{3}x)x\right) \text{ for } x \in (0,6)$$

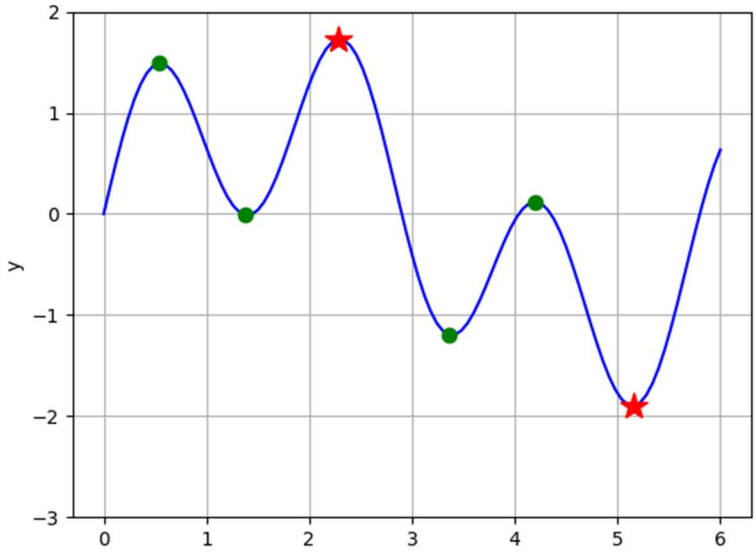

Figure 3.1. The multimodal function of one variable

For simplification of the notation, only minimization problems are of interest in this dissertation. There are numerous optimization techniques, that aim to find the best solution, however, most of those usually end up finding only local minima. This is the case mostly for deterministic optimization, especially gradient-based algorithms. This is why alternative optimization methods were developed, to avoid premature convergence to the local extrema, instead of exploring the available domain to find the global, true extrema. Generally, the optimization methods can be divided in the way shown in Figure 3.2. The mathematical optimization algorithms may be divided in many ways, so the proposed scheme is not the only possible one, as commonly the primary distinction is done between local and global search algorithms. However, the scheme proposed below shows the nature of the algorithms with some most common examples of a given algorithm type.

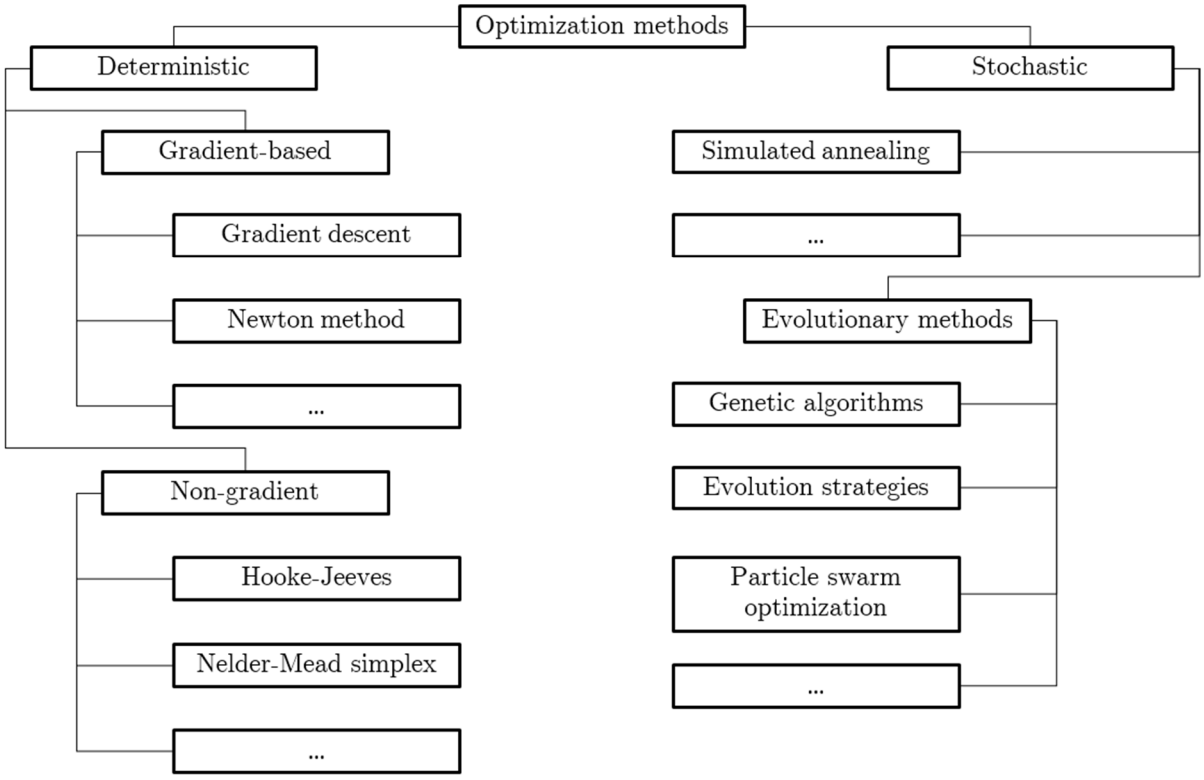

Figure 3.2. Examples of optimization methods

Optimization problem formulation contains the objective function (or functional) and sets of equality or nonequality constraints, as in (3.2)-(3.3):

$$\min_{\mathbf{x}} y = f(\mathbf{x}) \tag{3.1}$$

such that:

$$g(\mathbf{x}) \leq 0 \tag{3.2}$$

$$h(\mathbf{x}) = 0 \tag{3.3}$$

where $f(\mathbf{x})$, $g(\mathbf{x})$, and $h(\mathbf{x})$ are functions of the design variables vector $\mathbf{x}$.

The considerations of the constraints set up yet another division of the optimization algorithms. To handle to $g(\mathbf{x})$ and $h(\mathbf{x})$ constraints problem, they must be incorporated into the original objective function $f(\mathbf{x})$. This may be achieved in two alternative ways:

- The Lagrange method
- The penalty method

The first method directly reformulates the optimization problem of the form (3.1)-(3.3) into the new one (3.4):

$$\min_{\mathbf{x},\lambda_L,\mu_L} y = L(\mathbf{x}, \lambda_L, \mu_L) \tag{3.4}$$

where $L(\mathbf{x}, \lambda_L, \mu_L)$ is called a Lagrange function (or *Lagrangian*) of the form (3.5):

$$L(\mathbf{x}, \lambda_L, \mu_L) = f(\mathbf{x}) + \ \lambda_L h(\mathbf{x}) + \mu_L g(\mathbf{x}) \tag{3.5}$$

while λ_L and $\mu_L$are Lagrange multipliers that are now new design variables.

The second method, i.e. the penalty method, is based on a simpler concept. The objective function (3.1.) is modified in such a way, that once the set of constraints (3.2)-(3.3) is violated, the penalty is added to the modified objective function, as in (3.6), where the external penalty method is presented [63]:

$$\underset{\boldsymbol{x}}{min}\ y = f(\mathbf{x}) + S(\mathbf{x}) \tag{3.6}$$

where the $S(\boldsymbol{x})$ is the calculated penalty (3.7):

$$S(\mathbf{x}) = h(\mathbf{x}) + \max\ (0, g(\mathbf{x})) \tag{3.7}$$

Almost every arbitrary form of the $S(\mathbf{x})$ may be used in this method, as there is no general rule on how to choose the form of the penalty function. If there is no continuity requirement, the penalty may even take the form of a step function, where there is a constant number added on top of the objective function regardless of the level of constraint violation. However, the usage of such a step penalty function may be undesirable in certain situations, such as in metamodeling.

Assuming that the violation $\boldsymbol{v_p}$ represents the difference between the calculated constraint function value and the required constraint value, i.e. $v_p = \max\big(0, g(\mathbf{x})\big)$, the following general penalty function (3.8) may be incorporated:

$$S(\mathbf{x}) = P_b + P_m * v_p^{P_e} \tag{3.8}$$

where P_b is the penalty base (step), P_m is the penalty multiplier that scales the violation v_p which is raised to the power P_e (penalty exponent).

3.1.1. Gradient-based optimization

Thanks to the aforementioned development of the calculus of variations, the gradient-based methods were born and are until this day widely used in optimization problems [109]. The general idea behind the gradient methods is to use the information about the variation of the objective function in the vicinity of the analyzed point $\mathbf{x}$. This is done by differentiation of the objective function over the set of design variables (3.9):

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f(\mathbf{x})}{\partial x_1} \quad \frac{\partial f(\mathbf{x})}{\partial x_2} \quad \ldots \quad \frac{\partial f(\mathbf{x})}{\partial x_n}\right]^T \tag{3.9}$$

The graphical representation of the gradient is the vector, which is perpendicular to the level set of the function hypersurface going through that point. A gradient vector field together with the function $f(x, y) = x^2 - y^2$ is shown in Figure 3.3. The length of the vector (arrow) represents the steepness of the function in the gradient calculation point. The arrow points to the direction of maximum change in the function value. It is clearly visible, that going in the direction of a negative gradient minimizes the value of the analyzed function, which shows the nature of gradient-based optimization.

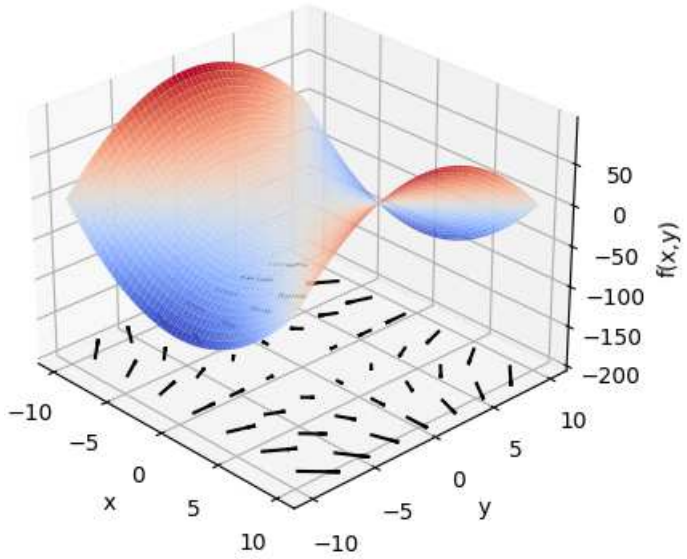

Figure 3.3. Gradient field (arrows) of the nonmonotonic function

One interesting feature can be observed in Figure 3.3 – the gradient is zero at the point [0,0], so it could be expected to find the local function minima there. However, this is the saddle point, in which the function first derivative (gradient) is indeed zero, but it does not meet the local extrema definition, as points in the vicinity of this point show both higher (x) and lower (y) function values. Therefore, it is useful to rely not only on the function gradient but also on the rate of its change, i.e. function second derivative. To do so, the Hessian matrix is introduced (3.10):

$$\mathbf{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(\boldsymbol{x})}{\partial x_n^2} \end{bmatrix} \tag{3.10}$$

For any *critical point*, i.e. a point in which all of the first derivatives of the function are zero, the following test can be conducted using the Hessian matrix to verify the presence of minima or maxima:

- If the Hessian matrix $\mathbf{H}(\mathbf{x})$ is positive definite in the critical point, the function f has a local minimum in this point

- If the Hessian matrix $\mathbf{H}(\mathbf{x})$ is negative definite in the critical point, the function f has a local maximum in this point
- If the Hessian matrix $\mathbf{H}(\mathbf{x})$ is neither positive nor negative semidefinite in the critical point, the function f has no extrema in this point

Introduction of the information coming from the Hessian matrix $\mathbf{H}(\mathbf{x})$ to the optimization procedure is known as the *Newton method*. In general, as all of the gradient methods are iterative, the difference between first-order methods (purely gradient-based) and second-order methods (Hessian-enhanced) is visible in the determination of the direction for the one-dimensional search procedure. Equations (3.11) and (3.12) shows the direction $\mathbf{d}$ calculation procedure for 1st and 2nd-order method respectively:

$$\mathbf{d}^{(i)} = -\nabla f(\mathbf{x}^{(i)}) \tag{3.11}$$

$$\mathbf{d}^{(i)} = -H^{-1}(\mathbf{x}^{(i)})\, \nabla f(\mathbf{x}^{(i)}) \tag{3.12}$$

During the optimization iteration, the new design point is determined based on the search direction $\mathbf{d}$, step length h, and the previous design point, according to the formula (3.13):

$$\mathbf{x}^{(i+1)} = \mathbf{x}^{(i)} + \ h^{(i)}\mathbf{d}^{(i)} \tag{3.13}$$

until the optimum is found or stop conditions are met.

The *Newton* method requires the calculation of the Hessian matrix, which may be a costly process once the underlying function is unknown, so gradients must be approximated using finite differences, or the number of variables is excessive. Therefore, there is a group of *quasi-Newton* methods, in which the Hessian matrix is approximated iteratively over the optimization procedure. In such a case, the equation (3.12) is substituted with (3.14):

$$\mathbf{d}^{(i)} = -\mathbf{V}^{(i+1)}\nabla f(\mathbf{x}^{(i)}) \tag{3.14}$$

One of the most prevalent quasi-Newton methods is the *BFGS algorithm*, named after its creators names: Broyden, Fletcher, Goldfarb, and Shanno [15], [29], [33], [99]. It assumes the following approximation of the Hessian matrix $\mathbf{V}^{(i)} \approx \mathbf{H}^{-1}(\mathbf{x}^{(i)})$, as in (3.15):

$$\mathbf{V}^{(i+1)} = \mathbf{V}^{(i)} + \mathbf{A}^{(i)} + \mathbf{B}^{(i)} \tag{3.15}$$

with:

$$\mathbf{A}^{(i)} = \frac{\mathbf{a}^{(i)}(\mathbf{a}^{(i)})^T}{(\mathbf{a}^{(i)})^T\mathbf{r}^{(i)}} \tag{3.16}$$

$$\mathbf{B}^{(i)} = -\frac{\mathbf{V}^{(i)}\mathbf{s}^{(i)}(\mathrm{s}^{(i)})^T\mathbf{V}^{(i)}}{(\mathbf{s}^{(i)})^T\mathbf{V}^{(i)}\mathbf{r}^{(i)}} \tag{3.17}$$

where $\mathbf{r}^{(i)} = \nabla f(\mathbf{x}^{(i)}) - \nabla f(\mathbf{x}^{(i-1)})$ and $\boldsymbol{a}^{(i)} = \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}$.

In the first iteration of the BFGS algorithm, the Hessian matrix is approximated with the identity matrix and updated iteratively in the subsequent optimization steps. This quasi-Newton approach leads to a significantly reduced number of operations compared to the original Newton method, therefore it is broadly used in practical optimization.

3.1.2. Evolutionary optimization

The gradient-based methods shown in the previous chapter have certain limitations, like they tend to lead to local extrema (instead of global) or they require costly finite-differences calculations to approximate gradients once the underlying objective function is of unknown form. One of the methods to address those flaws is the incorporation of biology-inspired optimization algorithms [6], [69], which do not share the same drawbacks as the gradient-based methods. Among the group of evolutionary methods, genetic algorithms (GA) and evolution strategies (EVOL) are one of the most widely used, even with the ongoing development of particle swarm optimizers or artificial immune systems [18]. An additional advantage of genetic algorithms is the ease of parallelizing their execution, which results in lower optimization wallclock time [59], [61].

The creator of the genetic algorithms is John Henry Holland, who published his work in the field of computational intelligence originally in 1975 [45]. The basic idea behind the genetic algorithm is the replication of the adaptation process observed in nature, where the chance of survival is highest for individuals that are best adapted to the environment. As observed in the real world, the adaptation process takes a stochastic form, during which generations of individuals are changed semi-randomly, i.e. two descendants from the same parents do not share the same features (two siblings are not identical) in general. However, the set of inherited features is somehow related to the features of the parents. The following naming convention is introduced to better describe the evolution process:

- A **chromosome** is most commonly a string containing coded information of all of the optimization variables. It consists of an ordered collection of genes, each representing a single design variable.
- **Gene** is a single element of the chromosome

- The **population** is a set of chromosomes that forms a single generation. The amount of individuals (unique chromosomes) in the population is known upfront and it is one of the key parameters of the genetic algorithm. The population in each iteration of the genetic algorithm is referred to as a **generation**.
- The **fitness function** represents how well the given individual (represented by the chromosome encoding its properties) is fitted. The higher the fitness function value, the better for the individual, as its chance to survive and yield offspring increases.

The process of generating a new population by means of genetic operators (elitism, selection, crossover, and mutation) is controlled by the evolutionary optimization parameters, which directly influence the convergence rate (or even convergence at all) of the algorithm. **Elitism** assures, that the best-fitted individual among the generation is kept unchanged, i.e. it is replicated, to carry its desired features for the subsequent generation(s). **Crossover** is responsible for generating a new individual from the parents chromosomes, with a certain strategy. The parents that will be crossed are priorly picked using one of the **selection** strategies. The selection is responsible for minimizing the possibility that poorly fitted individuals will pass their genes to new generations, but rather they will extinguish. The last genetic operator is the **mutation**, which specifies the probability of the change of a single gene within the chromosome. It induces additional randomness in the generation of new offspring. The schematic representation of the working principles of the genetic algorithms is shown in Figure 3.4:

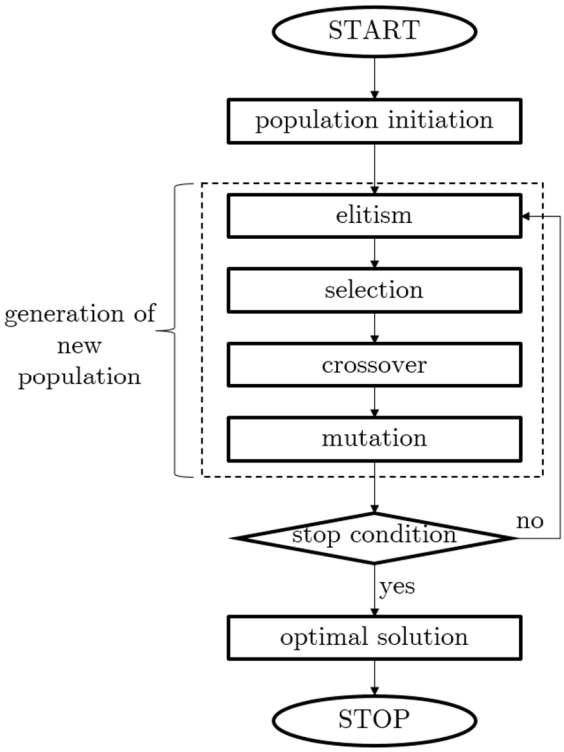

Figure 3.4. General working principle of genetic algorithms [63]

The selection of the individuals for a crossover may be done in several ways, each having an influence on the convergence of the genetic algorithm [32]:

- **Roulette wheel selection** – the individuals are occupying the roulette wheel angular width according to (proportionally to) their objective functions (fitness functions). The wheel is then moved, and the individual whose circular sector is overlapping with the roulette indicator is chosen as the parent for the next generation. The better the fitness of the individual, the higher probability of being chosen as a parent.
- **Ranked selection** is a similar selection method to the roulette wheel, but rather than dividing the wheel proportionally to the fitness function, it is divided according to the rank of the individual.
- **Tournament selection** – a group of k individuals is picked from the population and the most fitted individual from the group is picked as a parent for the next generation

Once the individuals that will serve as parents are selected, they are crossed together according to one of the following crossover scenarios:

- **Single-point crossover** – a point (position) among the chromosome is randomly selected. Genes that are positioned before that point inherit their values from parent A, while genes that are positioned after that point inherit their values from parent B
- **Multi-point crossover** – similar to a single-point crossover, but two points are randomly selected. An offspring inherits genes after parent A for genes positioned before the first and after the second point. The genes between points A and B are inherited from the parent B
- **Homogenous crossover** – each gene has an equal probability to inherit its value from parent A or B
- **AND / XOR crossover** – a gene is formed as a result of logical operator AND or XOR performed on the adequate genes of the parents

The examples of aforementioned crossover scenarios are shown in Table 3.1. Even in such a simple example, all of the generated offspring are different. The apostrophe symbol represents the point(s) of the crossover.

Table 3.1. Crossover types and examples

parent chromosome	**Parent A**	**Parent B**
	0101 0100	1111 0000
single-point crossover	010'~~1 0100~~ & ~~111~~'1 0000 -> 0101 0000	
multi-point crossover	01'~~01 0~~'100 & ~~11~~'11 0'~~000~~ -> 0111 0100	
homogenous crossover	~~0~~101 ~~0~~100 & 1~~11~~1 0~~00~~0 -> 1101 0000	
AND crossover	0101 0000	
XOR crossover	1010 0100	

The only process shown in Figure 3.4 that is still missing is the *stop condition*. There are two stop conditions that are most commonly used in genetic algorithms:

- **Maximum number of function calls** n_{GA_iter} which is calculated as the number of individuals n_{GA_ind} multiplied by the number of generations n_{GA_gener} in the genetic algorithm
- **Lack of objective function improvement** over n_{GA_gener} within the convergence limit n_{GA_ε}

Usually, the second stop condition is neglected, as due to randomness in the offspring generation process, there is always a chance to make an improvement, even after a significant number of generations without success.

All of the aforementioned operators describe the fundamental type of the genetic algorithm. However, one of the widely used modifications is especially popular – the multi-island genetic algorithm (MIGA). The core of the algorithm is analog to the one proposed by J. Holland. However, the MIGA incorporates several, distributed autonomous populations, which are represented as islands [16], [60], [62], [118]. The standard genetic operations are held out within each of the islands separately. After each n_{GA_migint} generations, the migration between the island takes place. The individuals are picked with a certain probability p_{GA_migr} and moved to another island, increasing the population diversity and introducing more randomness in the optimization process.

Genetic algorithms are examples of global optimization techniques. However, there is another biology-inspired optimization method that also aims to find the global optima, which is the evolution strategy. It was independently introduced by American scientist Lawrence J. Fogel [30] and by German scientists Rechenberg [84] and Schwefel [11], [94]. In the basic form of the evolution strategy, the only genetic operator is the mutation, which drives the optimization process.

During each iteration of the algorithm (as in Figure 3.5), a random vector, with normally distributed quantities, is added to the vector of design variables of the parent

individual. The standard deviation, which is responsible for the level of randomness in the random vector generation, is adjusted during the optimization, according to the number of successful mutations in the previous generations, making the evolution process self-adaptive. In contrast to the aforementioned genetic algorithm, the evolution strategy does not incorporate a binary representation of genes. Therefore, each gene (or design variable) is represented using a real value.

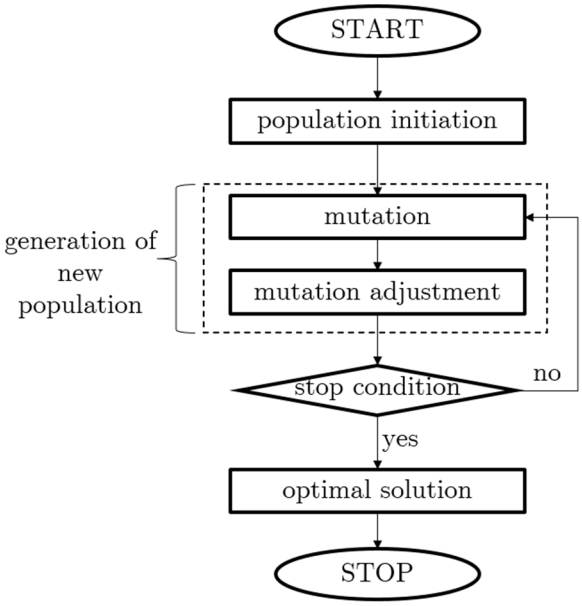

Figure 3.5. General working principle of evolution strategy

There are many variants of the evolution strategies, from which (1+λ_{ES})-ES is commonly used for its parallelization capabilities. Instead of having a population of 1 parent and 1 offspring, as in the fundamental implementation of the ES, the parent yields λ_{ES} number of mutants, each subjected to a different mutation vector (however with the same standard deviation). Those λ_{ES} individuals can be then computed in parallel (instead of consecutive calculations in (1+1)-ES), and the best of those mutants is used as a parent for the next generation.

The field of evolutionary optimization is under constant development, with recent works such as [55] in which the natural disasters that are happening during the lifespan of civilizations are imitated to replace certain individuals with randomly generated specimens with intact (previously unprocessed) chromosomes. Another novel type of GA [72] is based on alternating usage of genetic operators, i.e. crossover and mutation, which are done over individuals each $2n$ and $2n + 1$ generations respectively. Genetic algorithms are also incorporated in the optimization of automotive suspensions, as in [124] which proves the GA efficiency even with a relatively low number of generations.

3.1.3. Topology optimization

Topology optimization is a different field of mathematical optimization, as it is directly linked to the design process of solid structures. It serves as a dedicated mathematical tool that enables the determination of the optimal material distribution (layout) within a certain domain under given goals and requirements (objective function and constraints respectively). The growth of topology optimization techniques is connected to advancements in available computing power and computational mechanics, from which the finite element method is the most widely implemented. Topology optimization is the most general form among optimization problems, as shown in Figure 3.6, where a simple overview of structural design optimization is presented. Topology optimization may handle all types of design changes, including the ones which are imposed in the sizing optimization (here shown as explicit design parametrization) or shape optimization, which varies only the boundary of the body. On the other hand, neither sizing nor shape optimization is not able to create new boundaries, i.e. induce the general form of a hole in the structure.

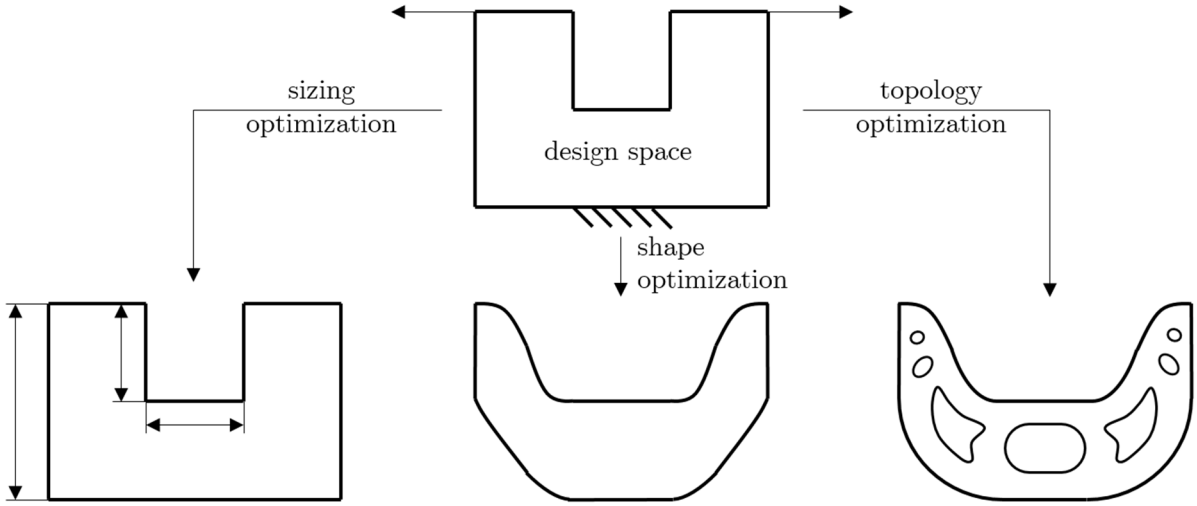

Figure 3.6. Types of structural design optimization [1]

The early works on FEM-based topology optimization are dated to 1988 when the homogenization method was introduced [9]. The aim of this method was to make the optimization process well-posed, which is not the case for the discrete 0-1 approach, in which each of the finite elements is either removed from or it is maintained in the structure being optimized. Homogenization allowed to make design variables continuous, as the material properties of the structure (within each finite element) were approximated using the homogenization technique. That started a new era, in which the implicit design description was used for optimization purposes, instead of explicit design parametrization, which is not general.

Shortly after the homogenization method, the new density-driven topology optimization strategy was proposed [8], in which the final layout of the structure was a function of elemental densities only. This material density was used for two reasons: firstly, it was used to calculate the mass of the whole design (summing elemental weights), and secondly, to interpolate the mechanical properties of the isotropic material [10]. In the most popular SIMP (Solid Isotropic Material with Penalization) method or the RAMP (Rational Approximation of Material Properties) method [105], the optimal distribution of elemental densities is sought, such that the optimized structure satisfy the initial constraints. This type of approach was called the "soft-kill" penalization, as it suffered from certain limitations, with intermediate "grey" densities being the most problematic one. During the optimization procedure (either by means of the optimality criteria method [78], or mathematical programming [106]), the resultant body could include elements with a density that does not correspond to purely void – lack of material, or to totally solid. This creates situations, in which the result may be hard to interpret physically, as in real representation the part is seen as a black-and-white design without any intermediate zones. Besides being a very computationally efficient method, the SIMP-based optimization technique suffers from other drawbacks, like solution dependency on the degree of material penalization, initial density, and especially numerical problems in intermediate-density elements subjected to high levels of strain [113]. The latter feature makes this method prone to a lack of convergence for tasks including structural stability, where nonlinear effects are taken into account (such as the ones considered within this dissertation). Additionally, in buckling or stability analysis, the material penalization scheme may strongly influence the level of convergence, or even lead to a lack of convergence at all [65]. Therefore, the density-based method like SIMP may not be that efficient in the cases of structural stability, even though multiple different constraints are incorporated in the method, like the stress constraints [46], [100].

Another family of the FEM-based topology optimization, hence not related to the adjoint, density-based formulation mentioned above, is the ESO (Evolutionary Structural Optimization) method, proposed in 1992 [122], [123]. The ESO method iteratively removes obsolete elements from the given design space. The contribution of the element to the objective function or constraint function is calculated based on the filtered sensitivities. The variant of ESO that allows the material (elements) to be added to the existing structure is called BESO (Bi-Directional Evolutionary Structural Optimization)[80]. Both methods are based on the heuristic assumption, that elements with the lowest criterion metric – stress, strain energy, or similar – are not efficiently used within the given space, therefore they must be excluded from the design. It is

done on a finite element basis, as the basic implementation of BESO operates on discrete values ("hard-kill" method).

None of these methods could efficiently represent the boundary of the optimized structure, due to the limitations of the elemental approach. Therefore, a new approach was proposed in 2000, called the LSM (Level-Set Method) [4], [98], [116]. It utilizes the auxiliary field ϕ that varies throughout 2D or 3D design space D. In each point of the design space, the auxiliary field value is calculated and compared with the assumed threshold value t_{LSM}. The LSM defines the design using the following relations (3.18) [25]:

$$\begin{cases} \phi(\mathbf{X}) > t_{LSM} & \rightarrow \quad \mathbf{X} \in \Omega \ (material) \\ \phi(\mathbf{X}) = t_{LSM} & \rightarrow \quad \mathbf{X} \in \Gamma \ (boundary) \\ \phi(\mathbf{X}) < t_{LSM} & \rightarrow \quad \mathbf{X} \in (D \backslash \Omega) \ (void) \end{cases} \tag{3.21}$$

The value of the auxiliary field is calculated as a sum of the basis functions of the form (3.22), where the $\mathbf{X}$ denotes the spatial coordinates and s_i are the optimization variables scaling the basis functions:

$$\phi(\mathbf{X}, \mathbf{s}) = \sum_{i=1}^{n} \phi_i(\mathbf{X}, s_i) \tag{3.22}$$

where the basis function $\phi_i(\mathbf{X}, s_i)$ can be represented as (3.33):

$$\phi_i(\mathbf{X}, s_i) = s_i N_i(\mathbf{X}) \tag{3.23}$$

Finally, when implementing e.g. Radial Basis Functions (RBF) for the Level-Set-Function (LSF) [117], the following form is obtained (3.24):

$$\phi_i(\mathbf{X}, s_i) = s_i * e^{-\frac{\|x - x_i\|^2}{d_i^2}} \tag{3.24}$$

where $\|x - x_i\|$ represents the Euclidean distance between the considered point and the center of the basis function, while d_i represents an additional scaling factor, but related to the width of the RBF, rather than its height (as s_i).

The centers of the basis functions (knots) may be located in the nodes of the given fixed FEM mesh grid or may be disjoint from the body discretization. Basis functions may be formulated over any arbitrary points, however, the regular grid is most commonly utilized in the literature. The iterative manipulation of the body boundaries during the optimization may be realized either directly by the optimization algorithm [89] or by utilizing shape or topological sensitivities with Hamilton-Jacobi equations [3] that track the shape and topology changes. The definition of the design using the LSM is shown in Figure 3.7, where the level set function defined by two radial basis functions is overlapped with the threshold plane, forming a curved boundary and the solid body inside.

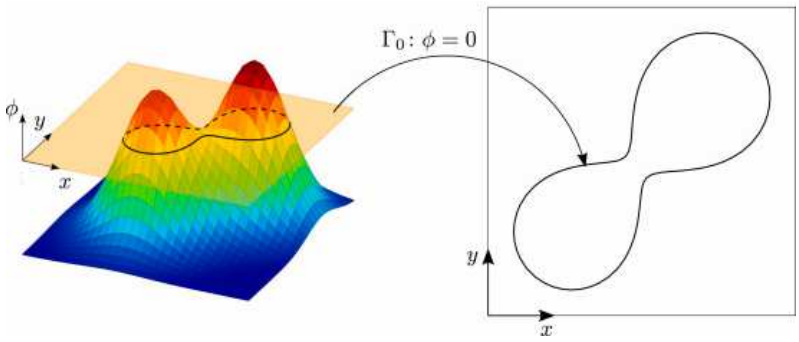

Figure 3.7. Graphical representation of the Level Set Method [73]

The number of radial basis functions used to create the auxiliary field is directly connected to the resolution of the resultant shape, as shown in Figure 3.8.

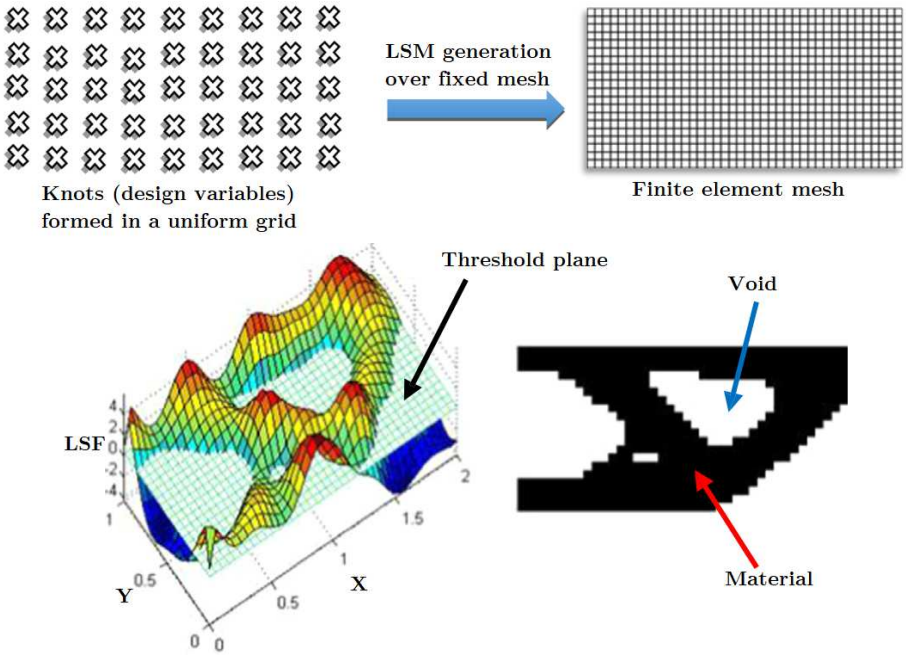

Figure 3.8. The cantilever beam topology optimization using 45 RBFs [38]

On the cost of higher optimization complexity (increased amount of design variables), the resultant topology may become more sophisticated, allowing for maximum exploitation of available design space.

3.2. Metamodeling

The usage of metamodels (also called surrogate models) enables the engineers to augment the analysis results obtained through expensive simulations, like the nonlinear finite element analysis described in the previous chapters [36]. Those models may mimic the underlying nature of the problem, once they are exposed to the samples (data points) obtained in a specific manner. Then, metamodels may be used either to replace the high-cost methods or to provide additional information based on the learned patterns. The main advantage of metamodels is execution time reduced by several orders of magnitude compared to the computer simulation. Chapter 3.2.1 describes the method of obtaining the training data, or rather designing a data sampling plan to create the metamodels most efficiently, i.e. achieve good fits within the limited number of expensive FEM analysis runs and possibly the best representation of the design space. Three different types of metamodels are described in the subsequent chapters 3.2.2-3.2.4, i.e. the Response Surface Method (RSM) [14], the Kriging [58], and the Artificial Neural Networks (ANNs) [49], representing metamodels widely used according to recent research papers [41], [102]. All of the descriptions and conclusions are made for normalized data, i.e. for inputs and outputs that are reduced to the range of [0,1]. This procedure allows to avoid the distortion of the polynomial coefficients, Kriging theta values, or neural network weights due to uneven orders of magnitude, making the training process more robust and resultant fitted models easier to analyze.

3.2.1. Sampling plans

In metamodeling, it is assumed that the underlying problem that is trying to be mimicked by a surrogate is of a black box type. The design variables are passed to such a black box, and the generated (calculated) responses are recorded, as in (3.25.):

$$\mathbf{x}^{(i)} \rightarrow \mathrm{y}^{(i)} = f(\mathbf{x}^{(i)}) \tag{3.25}$$

where $\mathbf{x}$ is the design variables vector, y is the model response, and i is the sample number. For k design variables, $f(\mathbf{x})$ is an output for a point within the design space D, such that $\mathbf{x} \in \mathrm{D} \subset \mathbb{R}^k$. The collection of all vectors $\mathbf{x}$ is called a sampling plan, represented as $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \dots, \mathbf{x_i}, \dots, \mathbf{x_n}\}$.

Such a black box process does not provide any general information about the model, or behavior of the underlying problem across the whole design space. Therefore, the design space must be sampled, i.e. numerous design points (design variables vectors) must be spread across the allowable space, from which the model response will be recorded and used for the surrogate training. The overlay of those points influences

greatly the quality of the constructed metamodel. There are numerous possible methods for the generation of the sampling plan for metamodel training and validation purposes, with full- or fractional-factorial designs of experiments (DOE, also referred to as grid search), (Optimal) Latin hypercubes (OLHS/LHS), Sobol sequences, orthogonal arrays, or Monte Carlo methods being commonly incorporated in the scientific papers [31]. Some of those methods are shown in Figure 3.9 below:

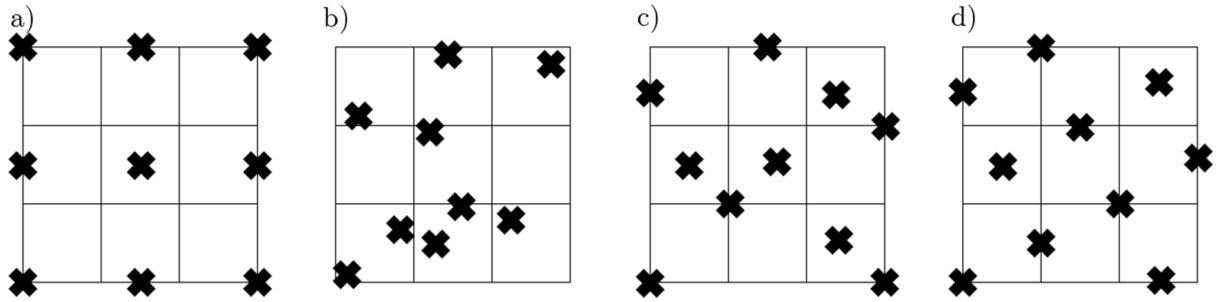

Figure 3.9. Sampling plans for 2 dimensions: a) full-factorial DOE, b) Monte-Carlo, c) LHS, d) OLHS [95]

Even though all of the above plans are applicable for metamodel training, each of them exhibits sets of advantages and disadvantages. The main problem associated with the factorial DOE approach is that the number of samples grows exponentially with the increase in the number of design variables, as:

$$n = 2^k \ or \ n = 3^k \tag{3.26}$$

where the base is adjusted depending on the number of levels in the DOE. For a simple problem with 10 variables, the sampling plan would require 2^{10}=1 024 or 3^{10}=59 049 samples, depending on the desired number of levels per factor. Even for such small problems, those numbers often seem exaggerated. This feature of the factorial sampling plans is referred to as a *curse of dimensionality* [53]. Therefore, for the time-consuming simulations, the factorial plans may be too computationally expensive. On top of that, even though a significant number of design points is generated, the distribution of samples projected on each of the design variable axes is very narrow, as shown in Figure 3.10, so there is no possibility of capturing higher-order dependency between the input and output variables (or harmonic behavior).

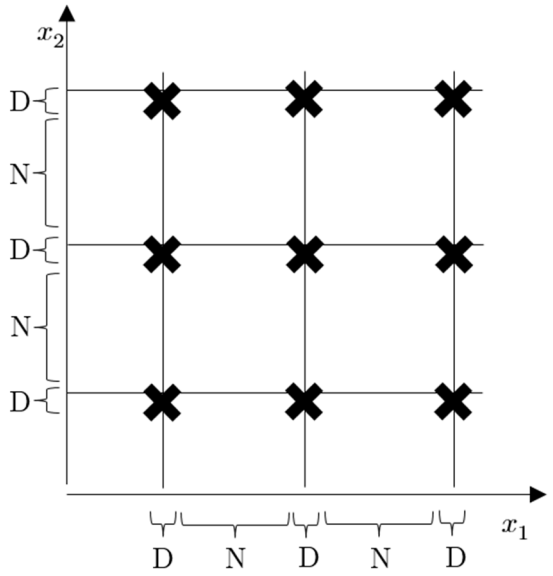

Figure 3.10. Sample points projected on x_1 and x_2 axes – areas, where multiple points are located are marked with D (dense), and areas without any sample points are marked with N (none)

It would seem reasonable to choose a random distribution of the design points, as it is quick and the resultant number of samples is not restricted to any dependency (like in the factorial design). However, a totally random process may generate areas of dense point distribution, or even overlapping points (not favorable for certain metamodeling techniques, like the Kriging, which will be described later) and areas without any points at all (lack of function information). Therefore, one of the most common techniques used for generating the sample plan is the Latin Hypercube Sampling (also referred to as Latin Hypersquare Sampling), in which the saturation of each of the design variables axis with design points is almost uniform, on the cost of unequal spacing between the design points within D. That is why, the Optimal Latin Hypercube Sampling was developed, to mitigate the aforementioned flaw, by the process of maximization of the minimum distance between any two design points, as described in [50]:

$$\min_{l\leq 1, j<n, i\neq i} d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \tag{3.27}$$

where $d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ is the distance measured for any two sample points $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$, calculated as:

$$d(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = d_{ij} = \left[\sum_{k=1}^{m} \left|\mathbf{x}^{(i)}{}_k - \mathbf{x}^{(j)}{}_k\right|^t\right]^{1/t} \tag{3.28}$$

and t depends on the used norm (1 - Manhattan, 2 - Euclidean).

The $\emptyset_p$ criterion, introduced in [74], which is based on (3.27), states that the sampling plan is optimal when the $\emptyset_p$ criterion is minimized (3.29). To verify that for a given LHS, all of the distances must be calculated. Once it is done, they must be sorted in ascending order $(d_1, d_2, \ldots, d_s)$. The next step verifies how many of the design points, within the sample plan, are separated by the same distance d_j. In this way, the index vector is created $(J_1, J_2, \ldots, J_s)$, which contain J_j elements. The sampling plan $\mathbf{X}$ is called the maximin plan (among all of the available plans) if it maximizes d_1, minimizes J_1, maximizes d_2, minimizes J_2, etc.

$$\emptyset_p = \left[\sum_{i=1}^{s} J_i d_i^{-p}\right]^{\frac{1}{p}} \tag{3.29}$$

The aforementioned plans, like Monte Carlo, LHS, or OLHS are suited for most types of surrogate models. However, there are groups of metamodels that need a very specific sampling plan to fit the model. One of such plans is the Box-Behnken (BB) [13] sampling designed especially for the Response Surface Method (RSM) surrogate. The plan minimizes the number of design points needed to fit the model, with the limitation of fitting the model up to the quadratic term (i.e. RSM with the squared term, product of two factors, linear term and constant). In cases where there is a special concern about the corners, i.e. the combination of the factors at the extreme values, RSM can be fitted using a more computationally expensive, Central Composite Circumscribed (CCC) plan. The CCC can be represented as the full factorial design with an additional star design, to capture the curvature of the output data. A graphical comparison of the Box-Behnken and CCC sampling plans for 3 factors (variables) is shown in Figure 3.11.

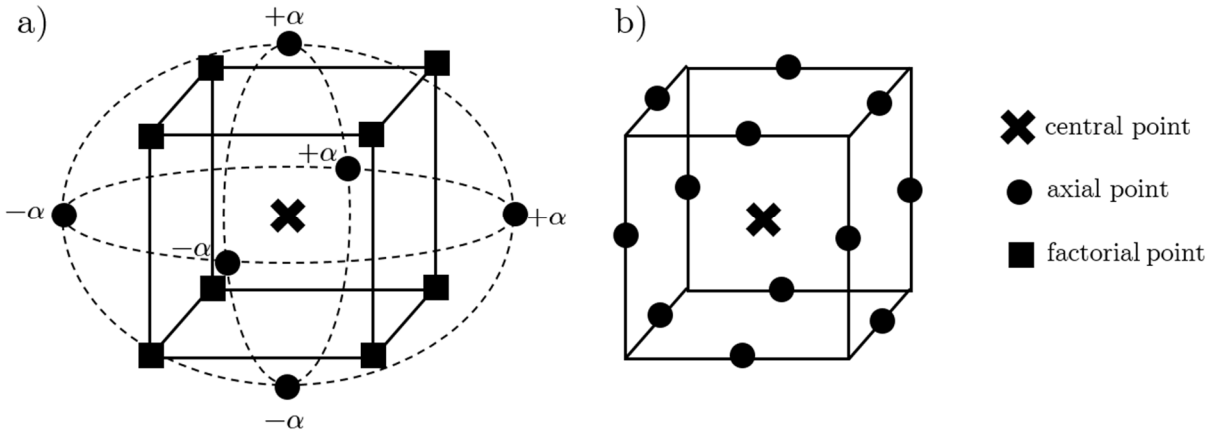

Figure 3.11. Sampling plans: a) central composite circumscribed (CCC), b) Box-Behnken (BB)

There are some drawbacks of the CCC plan, like the design points being (by default) positioned outside the design space, but this may be partially limited by adjusting the factor α. Unfortunately, some obstacles are hard to overcome, like the distribution of design points which does not apply to integer factors (e.g. number of individuals in the genetic algorithm, which cannot be a fraction). Those problems are mitigated in the BB sampling plan, where all of the variables (factors) are positioned in [-1,0,+1] levels, with -1 and +1 being the extreme values of a factor range.

In the process of constructing a surrogate, it is important to take into account that not only training vectors must be provided using a suitable technique, but also a set of validation data, which consists of data points that were not presented to the metamodel during the training. This way, a surrogate performance may be verified. If the quality metrics show a good fit, but poor prediction capabilities for unseen data, the model is said to be overfitted, as shown in Figure 3.12. Overfitting occurs e.g. when an overly intricate model is fitted to the data, making its predictions useful only in very close vicinity of the training points (sampled data), and useless in any other region.

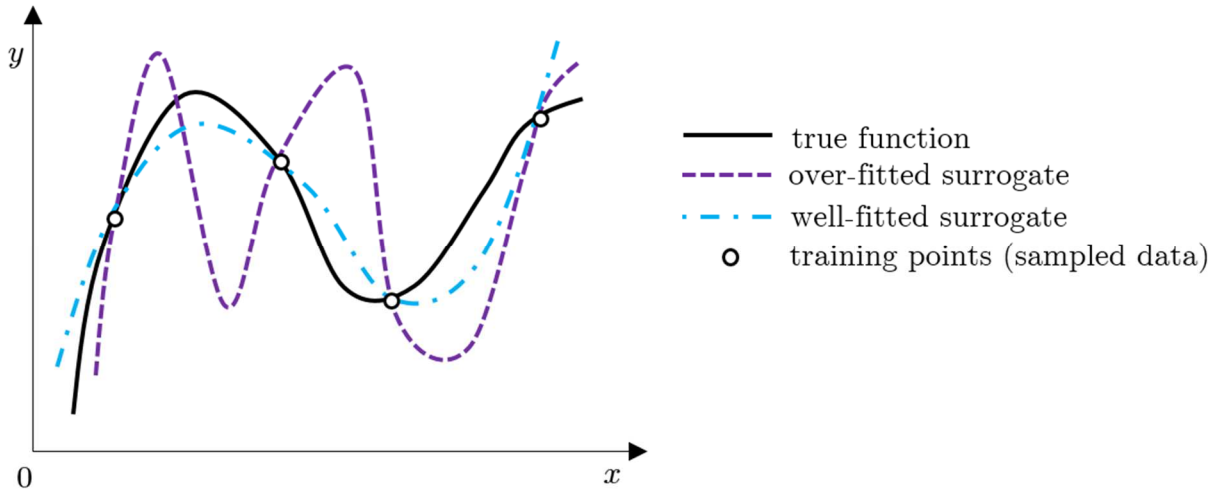

Figure 3.12. Surrogate predictions vs true function

3.2.2. Response Surface Method (RSM)

The Response Surface Method is one of the oldest techniques that is being used as the surrogate model. The concept was described in 1951 [14], and it consists of a polynomial equation that relates the output (predicted variable y) with the input variables by means of sums of the polynomial coefficients $\boldsymbol{\beta}$ multiplied by the design parameters vector $\mathbf{x}$, as in (3.30):

$$\hat{y} = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{ij(i<j)} \beta_{ij} x_i x_j \tag{3.30}$$

Equation (3.6.) represents the model with order up to the quadratic term only. However, it is possible to use 3rd or 4th-order polynomials, which results in a more flexible model, but on the cost of sampling the data on more levels. In such cases, Central Composite (circumscribed or inscribed) designs are preferred over the Box-Behnken ones, as CCC probes 5 values along the axis, while BB probes at 3 levels only, which is not sufficient to fit higher order polynomials.

One of the limitations of the RSM is the assumption made regarding the function (or process) being surrogated. If it is multimodal or highly nonlinear along the whole design space, then the RSM would provide a poor fit, as the generalization capabilities of the RSM are directly associated (limited) with its underlying mathematical model, i.e. polynomial equation. Nonetheless, if RSM is fitted properly, it provides insight about the modeled function, by the analysis of vector $\boldsymbol{\beta}$ containing fitted polynomial coefficients.

The response surface method has a unique feature (among other regression metamodeling techniques), that the derivatives of the $\hat{y}$ are available directly, therefore the RSM is suited for optimization purposes [70], as finding the function local minima/maxima is almost costless.

3.2.3. The Kriging

The Kriging method, even though much more sophisticated than the RSM, was developed at a similar time, as the original work by D. Krige was published in 1951 [58]. This interpolation method was initially developed for geostatic application, as the author master thesis. It was used for gold exploration, relying on soil sampling data measured over the area (two-dimensional problem). Nowadays, the Kriging method (referred to as *the Kriging*) is one of the most widely used interpolation methods, which finds use in multidimensional problems as well. As the Kriging is a very flexible model, it is eagerly used as a metamodel for strongly nonlinear processes, often represented by computationally expensive FEM-based simulations [81], [88].

Even though there are several types of Kriging, they share a similar form. The most common types of this model include ordinary, simple, universal, and blind Kriging [21]. The first two types have the most straightforward implementation and they provide good generality of the data. The prediction in the Kriging method is done using formula (3.31):

$$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + Z(\mathbf{x}) \tag{3.31}$$

where $\hat{y}$ is the predicted value, $f(\mathbf{x})$ represents the trend function, and $\mathrm{Z}(\mathbf{x})$ is the realization of a stochastic process, usually a normally distributed Gaussian random

process, with zero mean μ and standard deviation σ_{std}. The first component - $f(\mathbf{x})$ - is a global trend model. It is common to use constant $f(\mathbf{x}) = m$, in which m is the known mean, or $f(\mathbf{x})$ being the low-order polynomial function. On the other hand, the Z($\mathbf{x}$) may be understood as the part modeling local deviations, and it is fully characterized by its correlation function (3.32):

$$cor[y(\mathbf{x}^{(i)}), y(\mathbf{x}^{(l)})] = e^{-\sum_{j=1}^{k} \theta_j |x_j^{(i)} - x_j^{(l)}|^{p_j}} \tag{3.32}$$

The equation (3.32) represents the Gaussian basis function, in which the vectors $\boldsymbol{\theta}$ and $\boldsymbol{p}$ represent its shape (bell curve). The influence of changing the $\boldsymbol{\theta}$ and $\boldsymbol{p}$ parameters on the correlation function is shown in Figure 3.13:

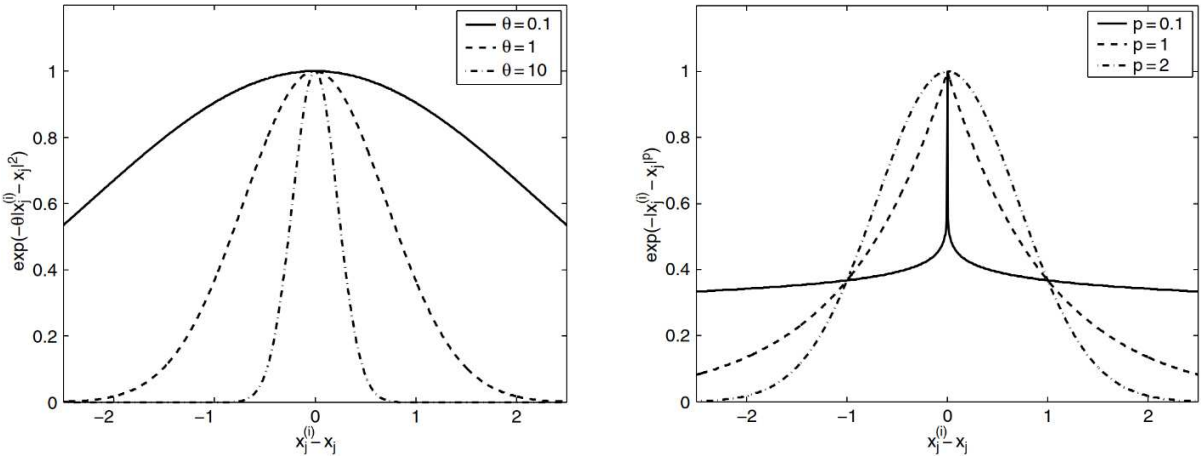

Figure 3.13. The correlation functions [53] with: a) varying $\boldsymbol{\theta}$, b) varying $\boldsymbol{p}$

When all of the design parameters (factors) are predicted to yield a similar influence on the output function, vector $\boldsymbol{\theta}$ can be replaced with a single value of θ. This yield an isotropic version of the Kriging, which is the simplest to fit, however in most engineering applications, the anisotropic Kriging with the vector of unknown θs is incorporated, providing more accurate predictions. The equation (3.32) may also be replaced by one of the other basis functions, including exponential, cubic spline, matern linear, and matern cubic [82]. One of the useful insights from the fitted $\boldsymbol{\theta}$ vector in the anisotropic Kriging is the preliminary analysis of the factors. Once the θs are fitted, the parameters mostly affecting the output function may be identified as the ones associated with the highest θ values, which may serve as the initial screening.

The correlation matrix $\boldsymbol{\Psi_{krig}}$ is generated for the whole dataset used to fit the model, as in (3.33):

$$\boldsymbol{\Psi_{krig}} = \begin{pmatrix} cor[y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(1)})] & \cdots & cor[y(\mathbf{x}^{(1)}), y(\mathbf{x}^{(n)})] \\ \vdots & \ddots & \vdots \\ cor[y(\mathbf{x}^{(n)}), y(\mathbf{x}^{(1)})] & \cdots & cor[y(\mathbf{x}^{(n)}), y(\mathbf{x}^{(n)})] \end{pmatrix} \tag{3.33}$$

The Kriging model fitting process is done by maximizing the likelihood L of the observed data $\boldsymbol{y}$, as shown in (3.34). This way the generalization error of the model is minimized.

$$L(y^{(1)} + y^{(2)} + \cdots + y^{(n)} \big| \mu, \sigma) = \frac{1}{(2\pi\sigma_{std}^2)^{n/2} \big| \boldsymbol{\Psi_{krig}} \big|^{1/2}} e^{-\left[\frac{(\boldsymbol{y}-\mathbf{1}\mu)^T \boldsymbol{\Psi}^{-1}(y-1\mu)}{2\sigma_{std}^2}\right]} \tag{3.34}$$

Equation (3.34) may be simplified utilizing natural logarithms to form (3.35):

$$\ln(L) = -\frac{n}{2} ln(2\pi) - \frac{n}{2} ln(\sigma^2) - \frac{1}{2} ln(\big| \boldsymbol{\Psi_{krig}} \big|) - \left[\frac{(\boldsymbol{y} - \mathbf{1}\mu)^T {\boldsymbol{\Psi_{krig}}}^{-1} (\boldsymbol{y} - \mathbf{1}\mu)}{2\sigma_{std}^2}\right] \tag{3.35}$$

Differentiating the equation (3.35) and equating to zero, the maximum likelihood estimates (MLEs) for the μ and σ_{std} are:

$$\hat{\mu} = \frac{\mathbf{1}^T {\boldsymbol{\Psi_{krig}}}^{-1} \boldsymbol{y}}{\mathbf{1}^T {\boldsymbol{\Psi_{krig}}}^{-1} \mathbf{1}} \tag{3.36}$$

$$\widehat{\sigma_{std}^2} = \frac{(\boldsymbol{y} - \mathbf{1}\mu)^T {\boldsymbol{\Psi_{krig}}}^{-1} (\boldsymbol{y} - \mathbf{1}\mu)}{n} \tag{3.37}$$

Using equation (3.35) with MLEs calculated in (3.36) and (3.37) without the constant terms yields the *concentrated ln-likelihood function* (3.38), which is now dependent solely on unknown θ and p parameters. It is worth mentioning that commonly, the Gaussian basis function incorporates the constant $p = 2$, which reduces the model fitting time with little to no reduction in its quality.

$$\ln(L) \approx -\frac{n}{2} ln\left(\widehat{\sigma_{std}^2}\right) - \frac{1}{2} ln(\big| \boldsymbol{\Psi_{krig}} \big|) \tag{3.38}$$

The function (3.14) is usually maximized utilizing one of the global search optimization techniques, like simulated annealing or genetic algorithm. Once the thetas are optimized, the Kriging may be used for predictions, as shown in (3.31). To obtain the function value in the unknown location (within the design space), the correlation matrix $\boldsymbol{\Psi_{krig}}$ is augmented with the vector of correlations between the new prediction $y(\mathbf{x})$ and data used to fit the model, as shown in (3.39):

$$\boldsymbol{\psi} = \begin{pmatrix} cor\left[y(\boldsymbol{x}^{(1)}), y(\boldsymbol{x})\right] \\ \cdots \\ cor\left[y(\boldsymbol{x}^{(n)}), y(\boldsymbol{x})\right] \end{pmatrix} \tag{3.39}$$

The augmented correlation matrix $\widetilde{\boldsymbol{\Psi}}$ takes a form (3.40):

$$\widetilde{\boldsymbol{\Psi}} = \begin{pmatrix} \boldsymbol{\Psi} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^{\mathrm{T}} & 1 \end{pmatrix} \tag{3.40}$$

Again the ln-likelihood is calculated this time for the augmented data. As a result, the following formula for the Kriging prediction is achieved:

$$\hat{y}(\boldsymbol{x}) = \hat{\mu} + \boldsymbol{\psi}^{\mathrm{T}} \boldsymbol{\Psi}^{-1} (y - \mathbf{1}\hat{\mu}) \tag{3.41}$$

The formula (3.41) describes the interpolating Kriging metamodel. However, when dealing with noisy data (physical experiments), or when generalization properties are more important than the interpolation capabilities (numerical simulations), the regression Kriging might be used instead. It belongs to the approximators family, i.e. it does not represent the actual data when making a new prediction for the samples used to fit the model. To achieve such a surrogate model, the equation (3.41) is modified, by adding the constant term λ to the diagonal of the correlation matrix $\boldsymbol{\Psi}_{\boldsymbol{krig}}$, yielding the following regression Kriging equation (3.42):

$$\widehat{y_r}(\mathbf{x}) = \widehat{\mu_r} + \boldsymbol{\psi}^{\mathrm{T}} (\boldsymbol{\Psi}_{\boldsymbol{krig}} + \lambda \mathbf{I})^{-1} (y - \mathbf{1}\widehat{\mu_r}) \tag{3.42}$$

where:

$$\widehat{\mu_r} = \frac{\mathbf{1}^T (\boldsymbol{\Psi}_{\boldsymbol{krig}} + \lambda \mathbf{I})^{-1} \boldsymbol{y}}{\mathbf{1}^T (\boldsymbol{\Psi}_{\boldsymbol{krig}} + \lambda \mathbf{I})^{-1} \mathbf{1}} \tag{3.43}$$

In the Kriging method it is assumed that the data comes from the random process, so the uncertainty of the prediction, once the model is fitted, may be used for either the in-fill process [83], [92] or to assess the possible range of the output, instead of taking simply the mean output value. Researchers highlight that current trends in Kriging methods development are devoted to the aforementioned infill process [41] and minimization of the hyperparameters number [12].

3.2.4. The Artificial Neural Networks

The basic idea behind the artificial neural network is to mimic the behavior of the human nervous system. The main part of this system is a single neuron (Figure 3.14 a) which collects the information from other neurons via dendrites, aggregates the collected signals via a cell body, and sends the collated data via axon to the axon terminal. Other neurons dendrites are connected with those terminals via a connection called a synapse. As synapses vary in their sizes and amount of neurotransmitters, the same signal may cause different levels of activation between the connected neurons.

The analogy between McCulloch-Pits artificial neuron [68] introduced in the 1940s and the actual neuron is quite straightforward. The dendrites, which are connected via synapses with other neurons, are replaced by weighted inputs $x_i w_i$, where the magnitude w_i is equivalent to the size of the synapse, and x_i represents the input signal received by the dendrite (Figure 3.14 b). Then, all of the weighted input signals are summed in the node, the same as in the cell body. Whether or not the neuron is

activated, is dependent on the summed input. Depending on the activation function, the artificial neuron may respond in a different manner to the same weighted input.

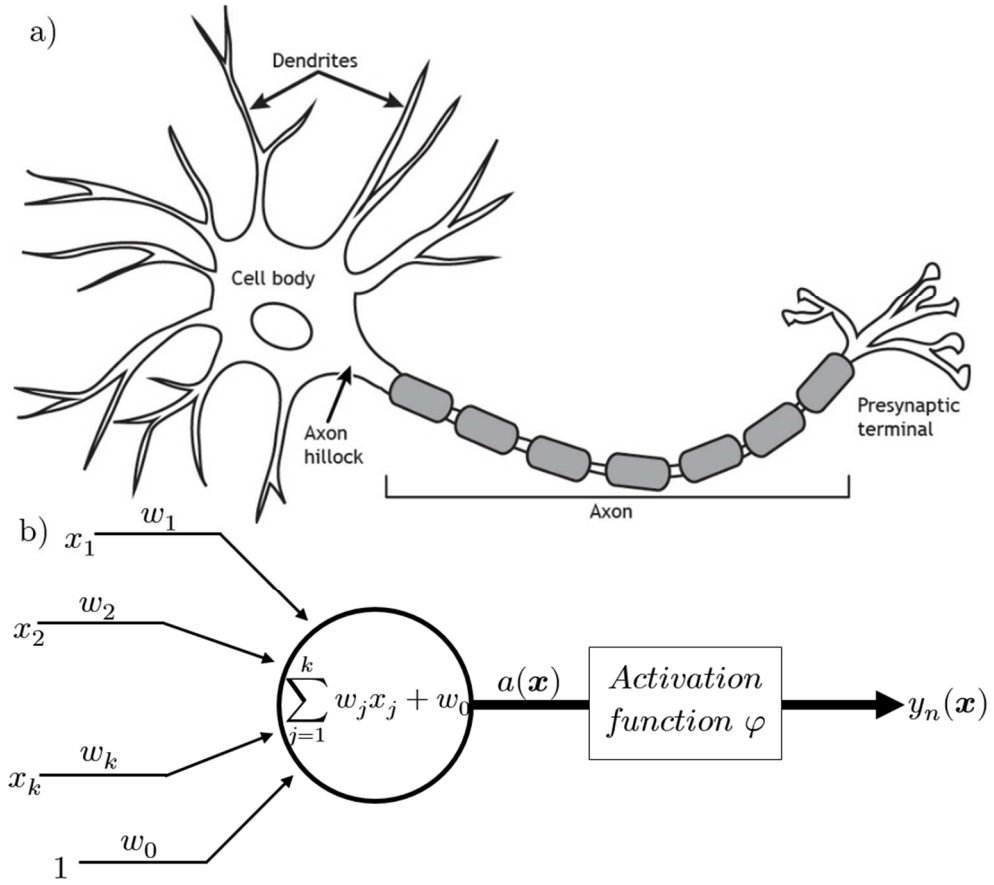

Figure 3.14. Model of a neuron: a) human (mammal) biological neuron [43], b) its artificial counterpart

To form a network of artificial neurons (Figure 3.15), single perceptrons are connected layer-to-layer, forming a dense grid of connections – a multi-layer perceptron (MLP).

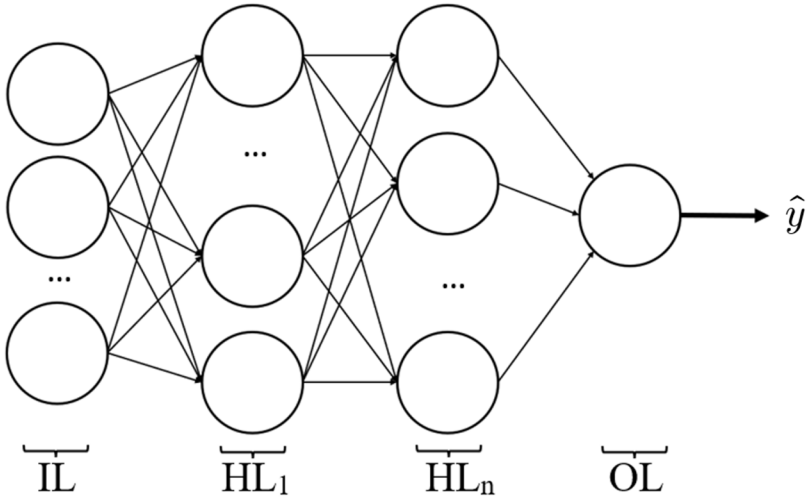

Figure 3.15. Neural network concept: multilayer perceptron network, where IL – input layer, HL_1 & HL_2 – hidden layer 1 & 2, OL – output layer

Each connection has its own weight, and each node has its bias term, which is equivalent to the weight of the unit signal. The output of the single perceptron, i.e. one node, is calculated using the formula (3.44). This bias notation is equivalent to $w_0 x_0$, where $x_0 = 1$.

$$y_n(\mathbf{x}) = \varphi\left(\sum_{j=1}^{k} w_j x_j + w_0\right) \tag{3.44}$$

The function $\boldsymbol{\varphi}$ is an activation function, which may be of many types. In practical applications, the most common types (yet one of the dozens available generally) are shown in equations (3.45)-(3.47)[2]:

Sigmoid activation function:

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{3.45}$$

Tanh activation function:

$$\varphi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \tanh(x) \tag{3.46}$$

ReLU (rectified linear unit) activation function:

$$\varphi(x) = \max\,(0, x) \tag{3.47}$$

The graphical representation of those functions is shown in Figure 3.16. The aforementioned type of ANN can be classified as the feed-forward network, in which the input data is transferred from the input neurons, through the sets of neurons in one or more hidden layers, and then passed through the output layer. Each of the subsequent layers works on the data processed by the previous layer. However, the main problem of the ANN is the determination of the weights and biases, which directly affects the way the data is modified throughout the layers.

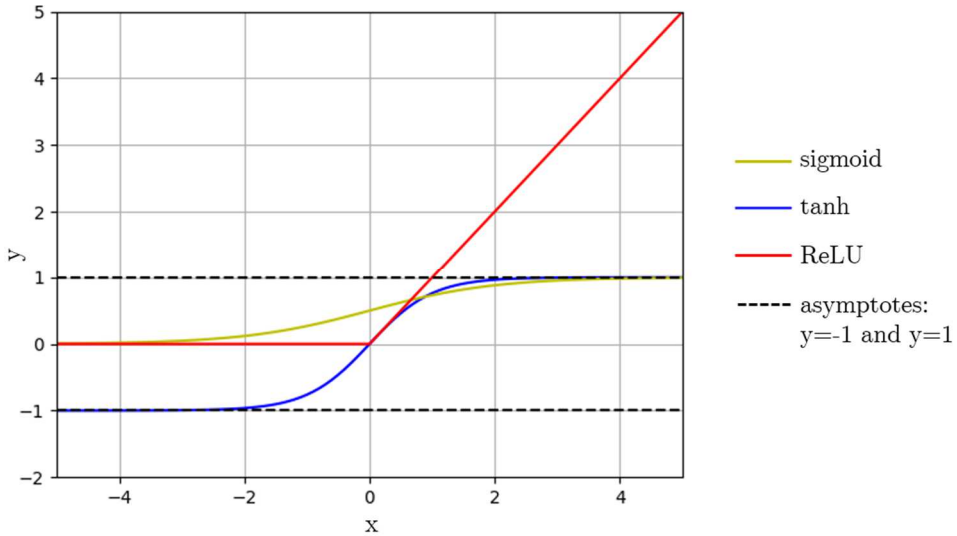

Figure 3.16. Neuron activation functions: sigmoid, tanh, and ReLu

The rapid growth of Artificial Neural Network (ANN) based metamodels started after the backpropagation learning algorithm was implemented broadly in the artificial neural networks [90], even though the first papers describing fundaments of this method were published over 20 years earlier [27], [54]. The backpropagation enabled the multilayer ANNs to learn complicated patterns of data efficiently.

Thanks to the aforementioned error backpropagation algorithm it is possible to achieve optimal, or at least quasi-optimal distribution of $\mathbf{w}$, during the supervised learning process. The network is fed with the input vectors $\mathbf{x}$ and the calculated $\hat{y}(\mathbf{x})$ is compared with the actual, known $y(\mathbf{x})$. The error metric (loss function) is usually the mean square error (MSE) between the target and predicted value (3.48):

$$E_{MSE} = \frac{1}{n}\sum_{i=0}^{n}(y^{(i)} - \hat{y}^{(i)})^2 \tag{3.48}$$

As the aim is to find $\mathbf{w}$ that is minimizing the E_{MSE} error, so the derivative of E_{MSE} must be calculated with respect to the $\mathbf{w}$. This cannot be calculated directly, but it is done using the rule of chain differentiation [35]. The weights update is done iteratively over *epochs*. The update scheme is shown in (3.49), which represents the delta rule:

$$\nabla w_i^{(j)} = \eta_1 \delta^{(j)} \frac{\partial \varphi(a)}{\partial a^{(j)}} x_i^{(j)} \tag{3.49}$$

where the η_1 represents the learning rate parameter and error $\delta^{(j)}$ is calculated differently for the last, output layer $\delta_o^{(j)}$ and hidden layer ($\delta_m^{(j)}$):

$$\delta^{(j)} = \begin{cases} \delta_o^{(j)} = y^{(j)} - {y_n}^{(j)} \ ; output\ layer \\ \delta_m^{(j)} = \sum_{k=1}^{n} w_m^{(k)(j)} \delta_k^{(j)} \ ; hidden\ layer(s) \end{cases} \tag{3.50}$$

and m is a neuron number in the hidden layer, n represents the total number of neurons from the subsequent kth layer, and (j) is the epoch number.

The learning rate parameter η_1 governs the pace at which the network is learning, with the usual values of $\eta_1 \in \langle 0.01, 5.0 \rangle$. A too-small value of η_1 result in a long learning process, i.e. an excessive number of epochs is needed to minimize the MSE. On the other hand, high η_1 values may cause the network not to learn at all, due to unstable gradient calculations. Another hyperparameter that influences the learning process greatly is the *momentum* η_2, which relates the current weight update with the weights update from the previous epoch, as in (3.51):

$$\nabla w_i^{(j)} = \eta_1 \delta^{(j)} \frac{\partial \varphi(a)}{\partial a^{(j)}} x_i^{(j)} + \eta_2 \nabla w_i^{(j-1)} \tag{3.51}$$

The momentum parameter can vary between $\eta_2 \in \langle 0.0, 1.0 \rangle$. Generally, learning with momentum reduces the number of epochs needed to train the network, as it allows for larger weight updates. Usually, the momentum method is paired with small learning rate parameters η_1, in order to mitigate the inadequate weight update, which is a response to the error function. As the weight update is associated with multiplying many gradients, including the activation function derivatives, it is important to avoid the problem of vanishing gradients, especially in deep neural networks (the number of hidden layers above one). This phenomenon happens for *tanh* and *sigmoid* (logistic) activation functions, as the derivative of $\varphi(x)$ is usually a small value, so when in the chain rule several such small values are multiplied, the resultant update of the weights may not yield any significant improvement. This may be solved by replacing the aforementioned activations functions with e.g. ReLU (3.47), which derivative is equal to one each time the neuron is activated.

The usage of ANNs in engineering is still intensive, as ANNs are recognized as a universal approximator[47] applicable in the regression (trend modeling) problems of engineering, as a metamodel cheap to execute and relatively easy to train, due to numerous existing dedicated programming frameworks. Another class of the ANNs used widely for metamodeling is the network with radial basis functions as hidden neuron activations [5], [103], due to their more succinct architecture (single hidden layer) and simpler training. However, the multilayer perceptron networks (MLPs) may take advantage of their extensive structure, numerous activation functions, and additional features like output normalization or network regularization [44], [104]. Also, they may incorporate numerous weight optimizers, including stochastic gradient descent (SGD) algorithm [86], SGD with momentum (including NADAM – Nesterov Adaptive Momentum [26]), and second order quasi-Newton L-BFGS (limited memory Broyden-Fletched-Goldfarb-Shanno [67]) algorithms, which are broadly used in many practical implementations of ANNs [64]. All of those features may seem like an excessive complexity, however, it provides flexibility in finding the optimal network topology, assuring a surrogate model with maximum efficiency.

4. Hybrid optimization

Numerous optimization techniques may be incorporated into engineering applications. Some of those algorithms were briefly described in Chapter 3.1, providing both the advantages and disadvantages of incorporating a certain type of optimization technique. There is a method, however, that enables to mitigate the drawbacks of individual algorithms called hybrid optimization, in which several optimization algorithms are coupled together [77], [96]. Hybrid optimization may incorporate a purely evolutionary approach, or may couple both the bio-inspired and gradient-based methods [17]. They may be run either in parallel or consecutively and may consist of two or more optimization techniques. This chapter consists of the description of such a hybrid algorithm applied to automotive shock absorber lower bracket topology optimization, consisting of a genetic algorithm and an evolution strategy.

4.1. Optimization problem formulation

The shock absorber consists of several structural components, as presented in Chapter 2.1.1. However, not all of those components may be freely modified to reduce their mass, as there are plenty of design requirements (like a need to achieve proper sealing) that forbid any arbitrary shape variations. Examples of such rules are e.g. the rod, which must always be cylindrical or the pressure tube must have a constant circular cross-section. One of the components that are not directly influenced by those requirements is the lower bracket, which in many cases is specifically designed for a particular application, taking into account the allowable design space, as well as load, mass, and manufacturing requirements.

The final form of the lower bracket is a subset of the total allowable volume, referred to as *design area* or *design space*. The volume of the design area is fulfilling any point in space in the vicinity of the shock absorber axis, that is not in collision with other suspension components, taking into account all possible suspension kinematic forms, its elastokinematic behavior (compliance), and required clearances. This yields a very uneven, complicated shape, within which the optimal material allocation is sought. An example of such a lower bracket design space, assembled together with the rest of the automotive shock absorber structural components is shown in Figure 4.1, where the considered design space is shown after the geometrical cleanup. The lack of possible material allocation in the central part of the bracket design space is a result of the vicinity of the drive shaft, while the upper corner and side chamfers are applied to avoid collision with the stabilizer bar, rim, tire, and brake calipers.

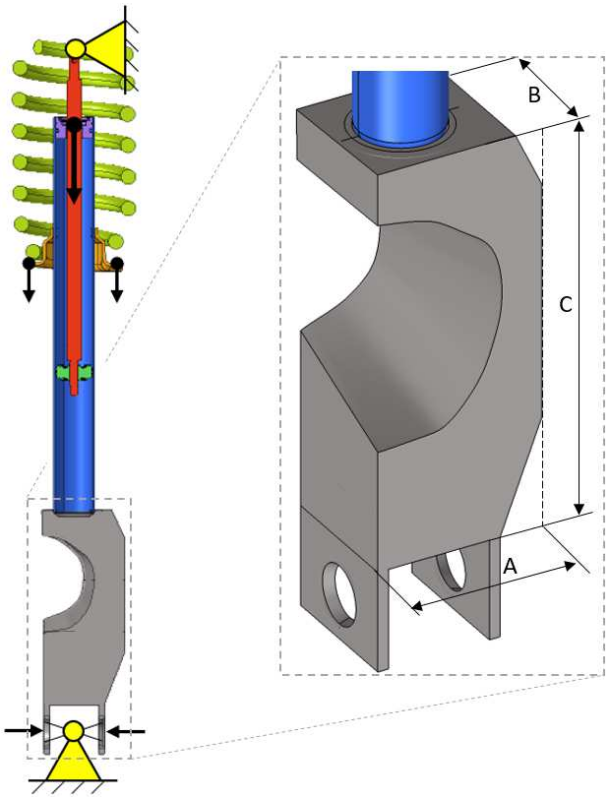

Figure 4.1. The design space (grey) within which the optimal shape of the lower bracket is sought

In the same figure, the typical loading scenario is shown, which represents the forces coming from vehicle suspension (jounce bumper and suspension spring) in the severe compression loading event, as highlighted in detail in Chapter 2.1.2, together with forces from the assembly process – bolt pre-tensioning that mounts the shock absorber with the swingarm via bushing. As the jounce bumper and suspension spring transfer all of the axial load (the contribution of damping force is insignificant), the sum of those forces is used to validate the design. The design validation is done based on vehicle manufacturers requirements, which specify i.a. the total shock absorber load at which no brakeage, crack, separation, or buckling may occur. The last condition is often the most challenging, as it tends to be the main failure mode of the shock absorbers. Especially, if the shock absorber is equipped with unsymmetric brackets which magnify the eccentric compression phenomenon while being subjected to excessive compression loading (as shown in Figure 2.6).

To represent these operating conditions most adequately, the constant suspension spring load is considered (as a pre-load), while the jounce bumper load is incrementally increased until the point of failure (buckling). The achieved buckling force (F_{max}) is then compared with the vehicle manufacturers requirement (F_{limit}), as the design must not fail below that limit (4.1).

$$F_{max} \geq F_{limit} \tag{4.1}$$

To obtain the F_{max}, which is the sum of the spring pre-load and incrementally updated jounce bumper load, the finite element method analysis is incorporated, as described in Chapter 2.2. A geometrical CAD model of the shock absorber (incorporating the design space) is used to build the numerical model, where each of the structural components is discretized using 2nd-order hexahedral or tetrahedral elements, which numerical formulation was given in Chapter 2.2.2. Loads are applied via the reference points connected with the structure through the MPC – multi-point constraints, i.e. distributing and kinematic couplings.

As the aim is to minimize the mass while maintaining the system stability (lack of premature buckling), the nonlinear FEM simulation is carried out in each design optimization iteration, incorporating load-controlled solution techniques described in Chapter 2.2.3. The FEM analysis stops once the system reaches its ultimate strength, i.e. there can be no further increase in the transferred load, which is shown in Figure 4.2 schematically by solid to dashed line transition. The F_{max} is obtained and recorded for each design (i.e. for each iteration of the optimization process). The red dotted line represents the F_{limit} requirement. In each iteration of the optimization process, a new (evolved) shape of the lower bracket is considered.

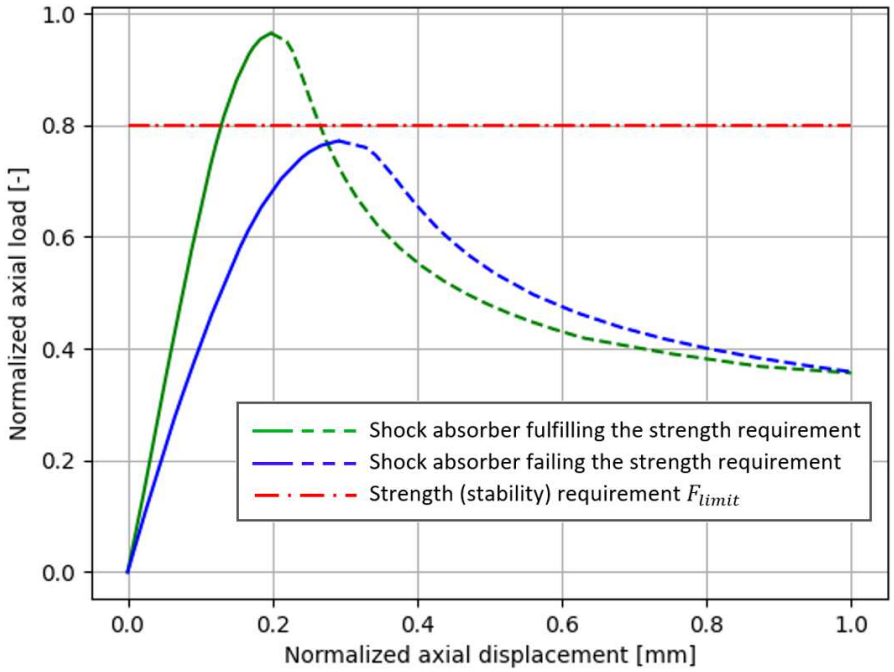

Figure 4.2. Load-deflection curves for two shapes of the lower bracket. Force-driven FEM analysis provides the solid line response (up to the point of maximum force), while the dotted lines represent post-buckling behavior, simulated using displacement-driven FEM analysis

At the beginning of each FEM simulation, the solver calculates the total mass of the system, using the elemental volumes and pre-defined material density. Therefore each design is characterized by two output variables: mass m and achieved

maximum load F_{max}. There are numerous different methods to iteratively modify the shape of any part, which are discussed in Chapter 3.1.3. To deal with the stability constraint, the modified, derivative-free Level-Set Method (LSM) is incorporated for changing the topology of the lower bracket. This method is similar in its formulation to [38], [39] but extended to three dimensions and coupled with filtering routines which will be covered in detail in Chapter 4.3. Also, it is modified to utilize global-range basis functions to reduce the total number of design variables.

What makes this method suitable for such applications is its simple parametrization, which can be coupled with any mathematical optimization algorithms described in Chapter 3.1.1 and Chapter 3.1.2, and therefore may easily handle FEM analyses that didn't converge to a given load value due to the loss of stability. The latter is a problem in most of the state-of-the-art topology optimization methods, as lack of FEM analysis convergence means e.g. lack of sensitivity calculations, and there is no step-back solution for algorithms like SIMP. Even very conservative density updates in the density-based methods do not guarantee avoidance of algorithm premature termination, which is often the case when dealing with unstable structures. On top of that, density-based methods may encounter equilibrium problems for intermediate-density FE subjected to large strains (which is the case for stability-based problems).

The aforementioned simple parametrization is yet capable of producing great variations in the shape of the part, which is not the case in sizing optimizations (when the object dimensions and their variability are pre-defined), even though a similar number of parameters may be involved.

On top of that, stress-based or strain-based techniques like Evolutionary Structural Optimization (ESO) may fail to provide a lightweight design, especially in parts subjected to eccentric compression and manufactured by forging. The reason for such limitation is the fact, that high-stress gradients are visible in the cross-sections (due to eccentricity), and maximum stress values are concentrated at the extreme layers of the section (as shown in Figure 4.3 for shock absorber subjected to axial compression). In this case, the only portion of material that is not subjected to excessive stress levels is in the middle of the cross-section. Unfortunately, this part of the cross-section cannot be removed without violating the manufacturing constraint (no hollow structures are allowed in forging), so the resultant envelope of the part is similar to the envelope of the initial design space. However, stress-based methods such as ESO may be used in the post-optimization phase to filter-out regions of neglectable stress, that do not necessarily contribute to the total strength of the part.

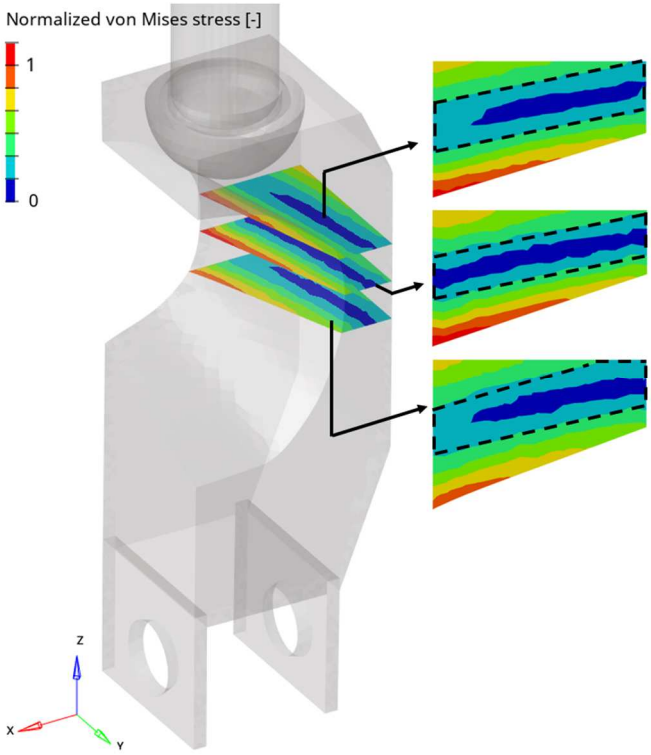

Figure 4.3. Normalized von Mises stress [-] in the critical cross-sections under compressive load. A value of 1 indicates the stress at the yield strength of the material (σ_0). The dotted area represents the volume of material that is stressed below 33% of the material yield strength.

On the other hand, the proposed version of the Level-Set Method is capable of modifying the aforementioned cross sections, even though the areas of maximum stress may be excluded from the design. In such a case, the reduced cross-section is subjected to higher stress, but it is done only if the stability constraint is met (4.1).

As the proposed version of the LSM algorithm does not change the geometrical model, but it affects the numerical model instead, the equation controlling the shape evolution (3.21) is rewritten to a new form (4.2). The boundary is naturally created by the edges and surfaces of the finite elements that remained in the numerical model after auxiliary field calculation and element removal, therefore the interface between the auxiliary field and the zero level set is not explicitly calculated (even though it is clear that the actual LSF boundary lays in the vicinity of the finite elements sides and vertices). This leaves a drawback of having uneven or sharp surfaces on the design contour.

$$\begin{cases} \phi(\mathbf{X}) > t_{LSM} & \rightarrow \quad \mathbf{X} \in \Omega \ (material) \\ \phi(\mathbf{X}) < t_{LSM} & \rightarrow \quad \mathbf{X} \in (D \backslash \Omega) \ (void) \end{cases} \tag{4.2}$$

The method of obtaining the design shape based on the auxiliary field (design translation) has been shown schematically in a 2D example in Figure 3.7 and Figure 3.8. During the design modification, there may be a situation, where mesh connectivity

is unsettled, i.e. a single 3D solid finite element (or a group of elements) is connected with the rest of the structure only via a single vertex, or not connected at all. In such a case, the numerical analysis will fail to converge, as poor nodal connectivity creates a mechanism. To overcome that issue, each of the finite elements in the considered design space is duplicated. The set of newly created elements is assigned with negligible stiffness, and its only role is to hold the modified, primary mesh together, even if the connectivity conditions are poor [20]. Such a glue-type mesh does not influence the system response (F_{max}), due to its marginal stiffness, but helps to overcome the numerical problems associated with hard-kill (0-1) topology change strategy, which is one of the main drawbacks compared to density-based methods like the SIMP. As this mesh does not require any new nodes, it does not directly increase the number of degrees of freedom of the numerical model, yet some additional, small computational cost associated with the structural response calculation of those elements is added to the analysis.

The shape of the part depends on the parameters s_i and d_{LSMi} specified for each of the radial basis functions, as shown in (3.22)-(3.24). The number and distribution of the basis functions are shown in the subsequent Chapter 4.2. The centers of the radial basis functions are referred to as *knots*, and the total amount of the knots is directly linked to the level of sophistication during design modification (number and size of details). At the same time, the number of optimization variables is related to the number of knots. In the considered application of the LSM, the Level-Set Function is calculated only for centroids of the finite elements that belong to the design space.

Having the set of design variables (**DV**), the constrained mass minimization problem can be formulated. The optimization of the lower bracket may be expressed in the following way (4.3):

$$\min_{\mathbf{DV}} f = \int_V \rho \, dV \tag{4.3}$$

subjected to the constraint:

$$q(\mathbf{DV}) > 0, \qquad q(\mathbf{DV}) = F_{max}(\mathbf{DV}) - F_{limit}, \tag{4.4}$$

Taking into account the discrete model and constant material density, equation (4.3) can be transformed into (4.5):

$$\min f = \left[\rho \sum_{i=1}^{N} V_i \right] \tag{4.5}$$

where V_i represent the volume of the ith finite element, and ρ represents material density.

As most of the mathematical optimization methods rely on the starting point, it is not easy to provide a good guess that will begin the design evolution phase. To overcome this issue, the randomness of initial design generation associated with the nature of the genetic algorithm is exploited. What is more, the genetic algorithms are the class of the global search methods, therefore wide space-search is anticipated, minimizing the risk of failing in the regions of local minima. Also, short-chromosome GAs (i.e. chromosomes with a low number of bits per gene) are very effective in global search applications, on the cost of their lower resolution [107]. That is why, they should be paired with the local search algorithms that exploit the vicinity of the GA-based solution. Previous studies [97] have shown, that evolution strategies may provide good local search capabilities when dealing with non-smooth functions (as the hard-kill FEM optimization). Therefore, the combination of the genetic algorithm together with the evolution strategy yields an efficient hybrid optimization scheme, which will be tuned and described in detail in Chapter 5. A similar algorithms layout, yet combined in parallel, was utilized in [119] for the optimization of power systems stability. In the considered shock absorber optimization case it is important to highlight, that all of the optimization runs are limited to 2000 objective function calls (FEM simulations) due to the total simulation time constraint. The hybrid optimization is then divided into 80%/20% parts, where the major part is related to optimization with the genetic algorithm, while the other 20% is spent on evolution strategy (EVOL), as shown in Figure 4.4. The genetic algorithm may be exploiting a single population (SPGA - basic formulation of the algorithm) or the multi-island version (MIGA).

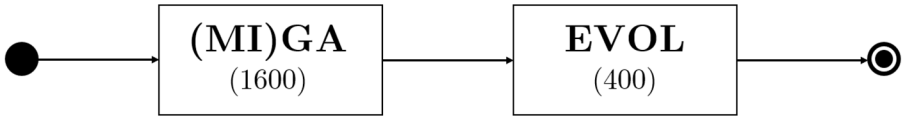

Figure 4.4. The proposed hybrid optimization flow [95]

During each hybrid optimization iteration, the procedure shown in Figure 4.5 is executed. The nonlinear FEM simulation is carried out using Abaqus/Standard [129], so the model modification is based on the Abaqus input file (*.inp). The penalty function calculation is based on the equation (3.8). Adjustment of the design variables vector is controlled by the optimization algorithm – either genetic algorithm or evolution strategy.

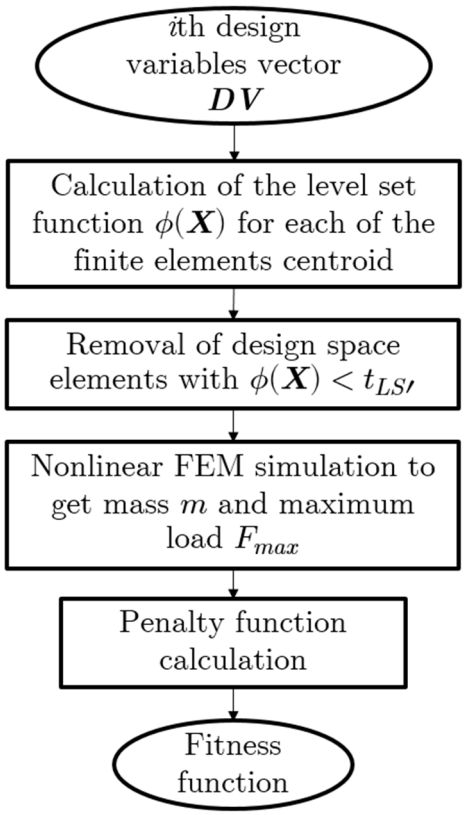

Figure 4.5. Single optimization iteration – design modification

The hybrid algorithm is terminated after a total of 2000 iterations (objective function evaluations). The individual with the highest fitness function value (i.e. lowest mass among individuals fulfilling the stability constraint) is then reported as the optimal one.

4.2. Interpolation schemes

There are an almost infinite number of possible combinations of the number and location of the knots (radial basis functions centers) that are used to build the auxiliary field (LSF). The most general form of the knot distribution is the uniform grid of radial basis function centers among the whole design space. This type of LSF interpolation scheme is the easiest to apply, as it does not require any prior knowledge about the expected, resultant shape. Also, it is capable of significant design modification, especially with the increased number of interpolation knots. A similar approach to the grid-like interpolation schemes can be found in [79], [108]. Examples of different, yet uniform knot distributions are shown in Figure 4.6 a)-d), with 25,27,28, and 54 knots in total. In cases of a), b), and d), the distribution of knots among the cross-section is the same in each layer of knots in the vertical direction. In the case of c), the knots, which are located in the corners of the circumscribed square, are rotated by 45° with every other layer.

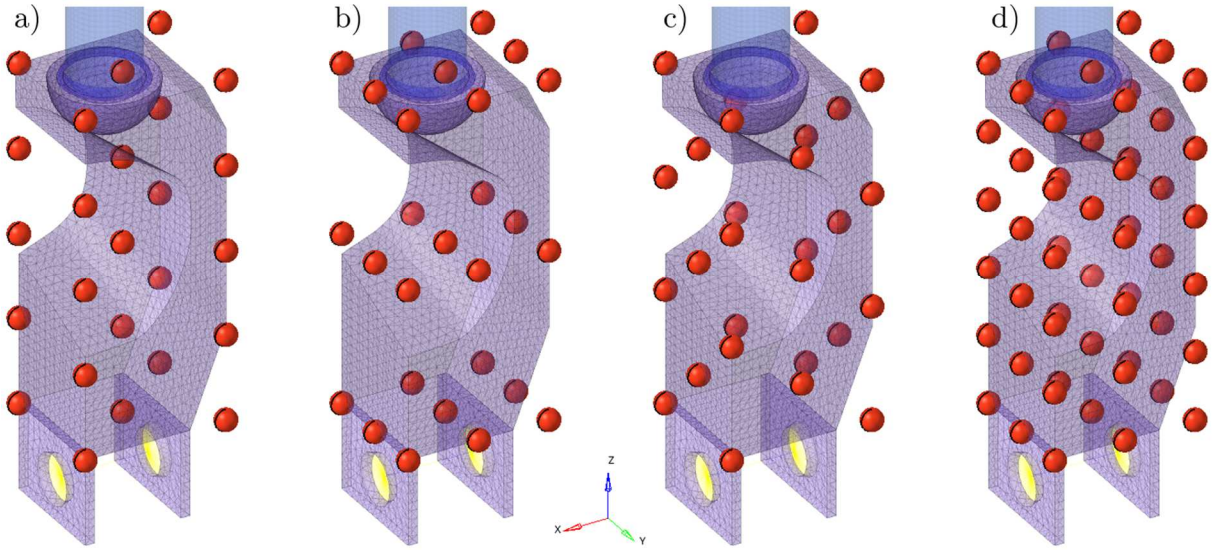

Figure 4.6. LSF interpolation knots distribution: a) 5x5, b) 9x3, c) 4x7, d) 9x6 knots

The non-uniform distribution of knots, however, may be encountered once there is an insight into the underlying nature of the problem, i.e. the areas, where the material will be allocated (or that will certainly not be allocated) are expected a priori. An example of such an application to shock absorber is shown in [97]. There is however a drawback to such an attitude, where each of the knots has a different distance to surrounding knots. That yields uneven widths of the basis functions, causing different sensitivities of scaling parameters, making the optimization process more complicated. Also, when the material cannot be allocated in certain areas (like the vacancy left for the driveshaft), placing the interpolation knot in the vicinity of such an empty area is increasing the optimization dimensionality without improvement in the possible shape quality, as the LSF calculated for any finite element will not be affected the scaling factors (s_i and d_{LSMi}) associated with this knot value. To overcome such an issue (like in Figure 4.6b, in the middle segment of knots), all of the parameters d_{LSMi} may be fixed before the optimization to act as a global RBFs (every RBF value is not negligible at any point of the design space). Then, equation (3.24) takes the form (4.6) for a three-dimensional, rectangular uneven prism:

$$\phi_i(\mathbf{X}) = s_i * e^{-\frac{\left(\frac{x-x_i}{A_{RBF}}\right)^2+\left(\frac{y-y_i}{B_{RBF}}\right)^2+\left(\frac{z-z_i}{C_{RBF}}\right)^2}{d_{RBF}^2}} \tag{4.6}$$

where A_{RBF}, B_{RBF}, and C_{RBF} are parameters used to normalize the uneven prism with edge lengths AxBxC (as shown in Figure 4.1) into a 1x1x1 cube. Parameter d_{RBF} is a constant value scaling the width of the RBF, similar to the θ in Figure 3.13. In such a case, the interpolation of the Level Set Function is done within the 1x1x1 cube. The total number of variables during optimization is therefore strongly limited, and is equal to the number of interpolation knots – and more specially – to the number of

parameters s_i scaling those i local LSFs. The interpolation scheme that was found to have good performance, i.e. ratio between the flexibility of the achieved design shape vs the interpolation complexity (number of variables), was the 3x3x3 knots structure presented in Figure 4.6b. Therefore it was used to modify the design using the developed S*patial Level-Set Method*.

The other utilized interpolation method uses a different approach to the shape modification, yet is also based on the interpolated auxiliary RBF field. In this method, referred to as the *Projection Method* (for clarity purposes), the design is modified by the two perpendicular projections of design contours. All of those contours are manipulated by the separate auxiliary fields. The common (shared) volume of the projected 2D shapes creates a new, intricate 3D structure, as shown schematically in Figure 4.7.

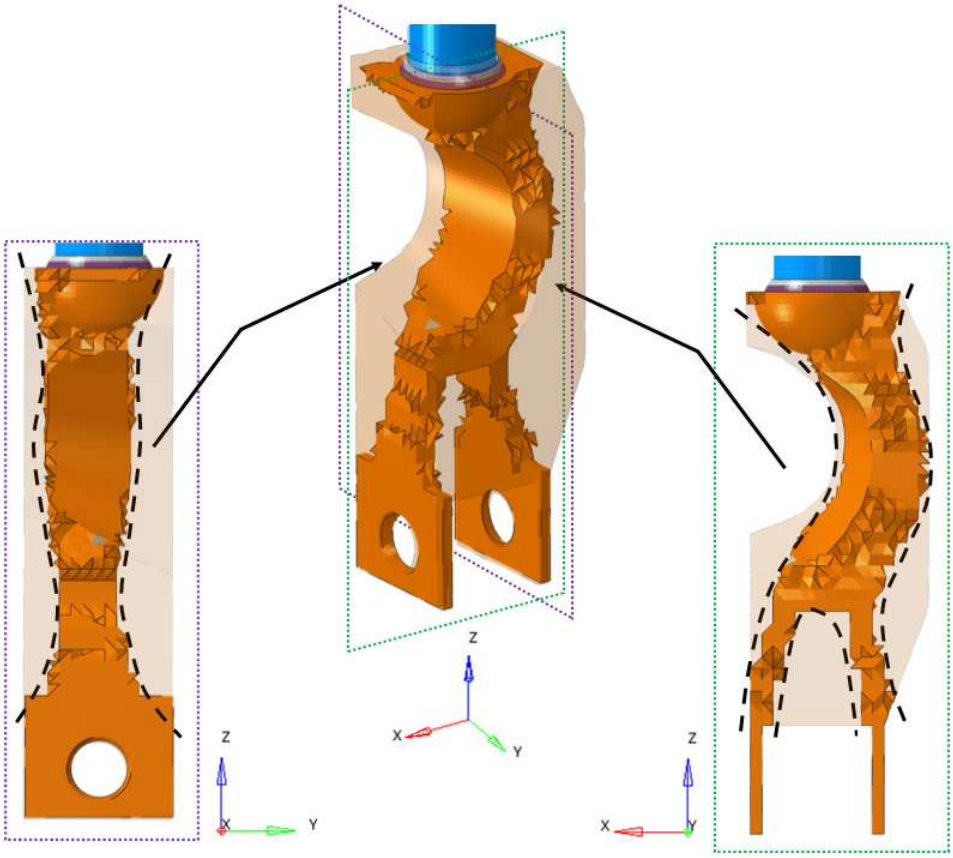

Figure 4.7. The projection method – the principle of shape generation

The contour of the 2D shape is obtained by comparing the coordinates of the finite elements with the calculated auxiliary field (similar to the *spatial LSM method*). However, in this case, the height of the RBF-based function (shown with the black dashed curves) of the XZ or YZ projection is compared with the horizontal coordinate of the finite elements. The horizontal coordinate refers to X-coordinate in the XZ projection and the Y-coordinate in the YZ projection. As there are two RBF-based functions in each projection, one related to positive and one to negative horizontal

coordinates, the resultant shape in the projection is a surface between those functions, as shown in Figure 4.8. The auxiliary functions are shown with solid black lines (contours), while the radial basis functions that are used to calculate them, are shown with red dashed lines, with the centers (knots) marked with yellow circles. Green straight lines represent the heights of the basis functions at their respective centers.

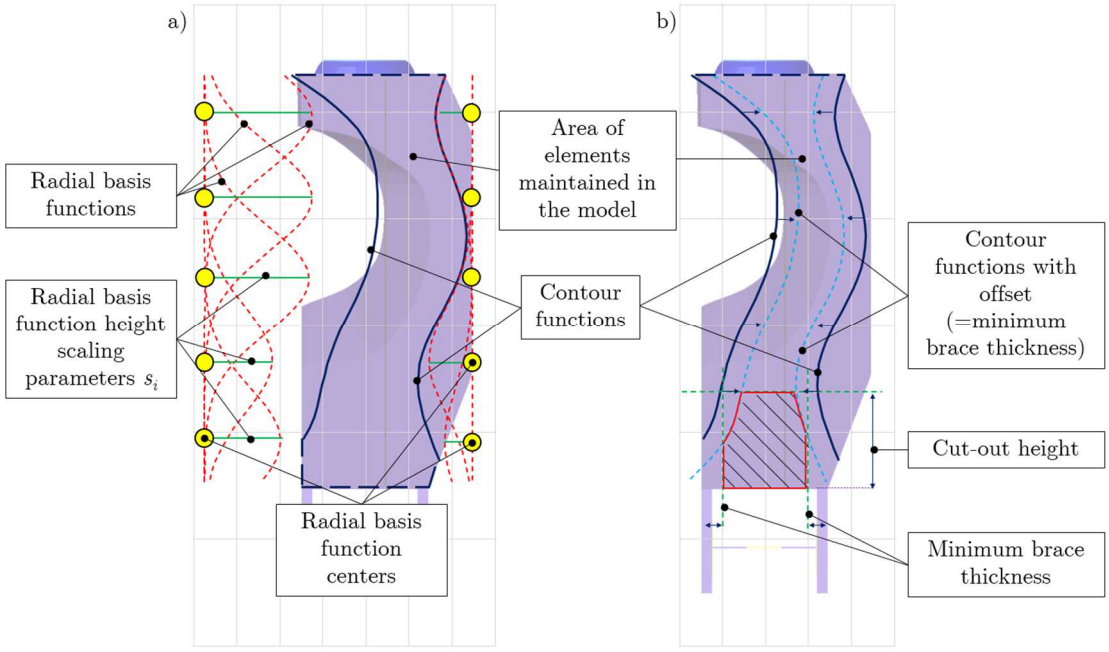

Figure 4.8. XZ shape modification using the *projection method* – a) auxiliary fields generation, b) the cut-out generation,

Also, for the XZ projection, the is an additional cut-out in between the bracket braces, as shown in Figure 4.8 b. Its height is directly manipulated as an additional design variable, while the resultant removal area is again a common surface between the offset contour functions, a minimum thickness of the braces, and the limiting height, as shown in Figure 4.9 by hatched area with red contour.

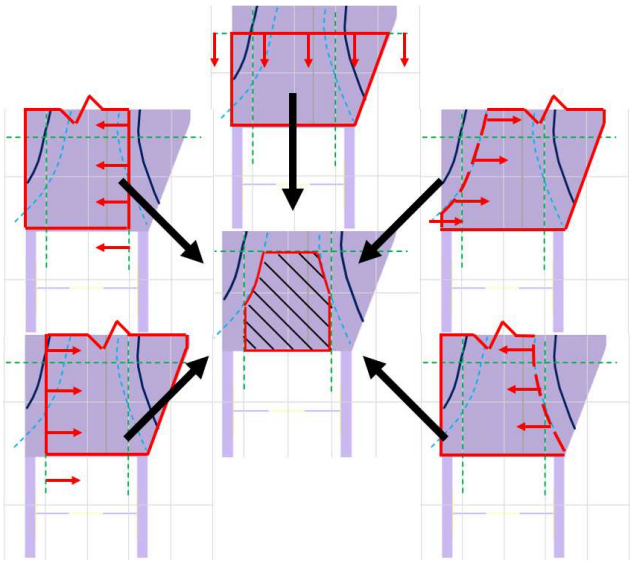

Figure 4.9. XZ shape modification– shared area for the cut-out

As a result of such parametrization, the total number of 22 parameters are included in the shape evolution:

- 20 parameters controlling the heights of the RBFs: each of 4 contours contain 5 RBFs
- 1 parameter controlling the height of the cut-out
- 1 parameter modifying the offset of the contour (related to the minimum thickness of the braces)

The RBFs that build up the contour are scaled and offset, to match the allowable design space in the most efficient manner. Firstly, the resultant contour line cannot cross the line of load application (line of symmetry in between braces which is equivalent to the central vertical axis of the shock absorber) – to avoid overlapping of the opposite contours. Secondly, the offset O_{pm} is adjusted in such a way, that the connection between the brace and the main part of the bracket is not removed during the optimization – the continuity of the material must be maintained in each iteration. To achieve those goals, two additional parameters - O_{pm} and C_{pm} - are added to (3.22) and (3.24), as shown in (4.7) and (4.8) respectively.

$$\phi(\mathbf{X}, \mathbf{s}) = \sum_{i=1}^{n} \phi_i(\mathbf{X}, s_i) + O_{pm} \tag{4.7}$$

$$\phi_i(\mathbf{X}, s_i) = C_{pm} * s_i * e^{-\frac{\|x - x_i\|^2}{d_i^2}} \tag{4.8}$$

where O_{pm} is the offset and C_{pm} is the constant used to scale the RBFs. As a result of such modifications, the main variables responsible for shape modifications - s_i – take any value from [0;1]. The contour functions are then interpolated in between the 5 RBF centers (per each side and each projection) assuring that the resultant shape is smooth with gradual transitions.

4.3. Manufacturing constraints

The intricate shapes that are generated as a result of either the *spatial LSM* or the *projection method* must fulfill the manufacturing requirements associated with the bracket production method. In the considered cases, the brackets are made by closed-die forging, followed by the machining of the contact regions. Those regions are the interface between the bracket and the tube, and the interface between the braces and bushing or swingarm attachment. The forging process requires that there are neither any undercuts in the forging direction (X-direction according to Figure 4.7) nor holes in the structure. Generally, a blunt, rounded shape without any sharp transitions is desired for forging applications.

The global-range RBF interpolation is used in both: the *projection method* and *spatial LSM*. Therefore, in contrary to the auxiliary field obtained with numerous local-range RBFs, the resultant shape of the structure is smooth and it lacks any rapid cross-sectional changes. In the case of the *spatial LSM*, it is possible to achieve such an auxiliary field, that the bracket body would consist of infeasible cross-sections, as shown in Figure 4.10. To overcome this problem, an additional filtering routine is added during the optimization process. Similar problems are addressed in the literature [114], however different approach regarding the cross-sectional material distribution is utilized. For the proposed method, during the optimization, after each design variables adjustment (knot values s_i), the body of the bracket is scanned iteratively to verify the feasibility of regions – as shown in Figure 4.10. The XY cross-section is checked iteratively over the height of the bracket in intervals of $2 * \textit{finite element size}$, and the centroidal coordinates of the FEs are projected onto the XY plane. Then, any discontinuities in the auxiliary field with $\phi > t_{LSM}$ along the forging directions are removed, assuring feasible shape of the cross-section, and therefore, of the whole part, once each of the sections is processed. This routine is shown schematically in Figure 4.10, where the discontinuous material distribution (along the forging direction) is mitigated, once the material is distributed evenly between the upper and lower contour of the cross-section. The algorithm of the filtering routine is shown in Figure 4.11.

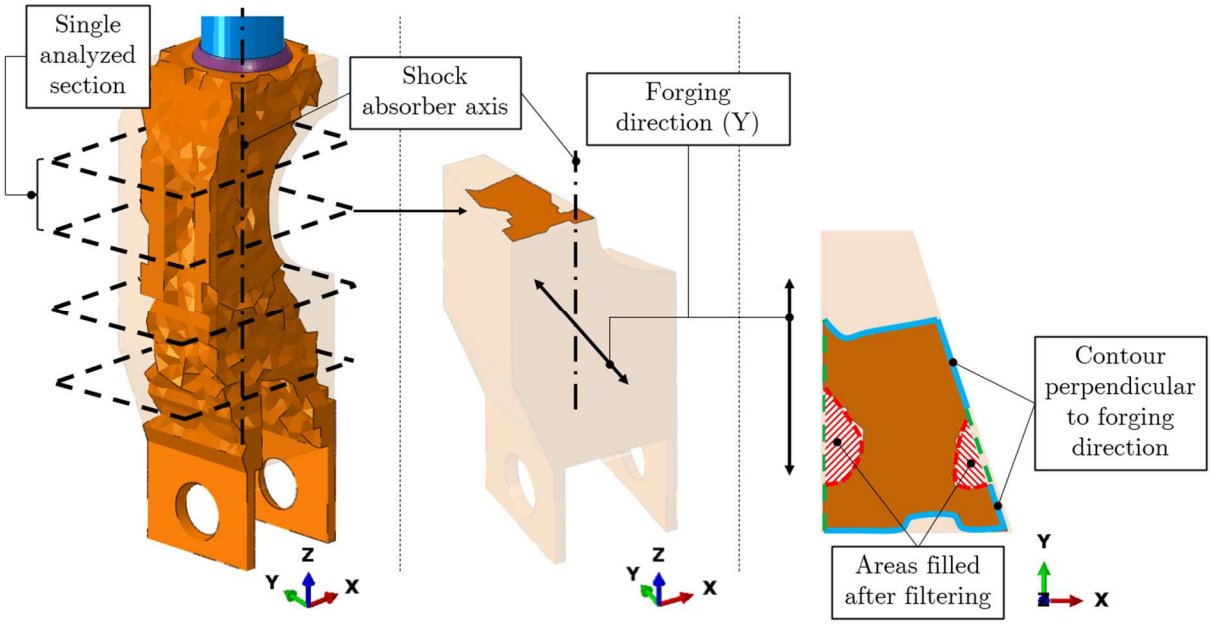

Figure 4.10. The unconnected regions resulting from the *spatial Level-Set Method*

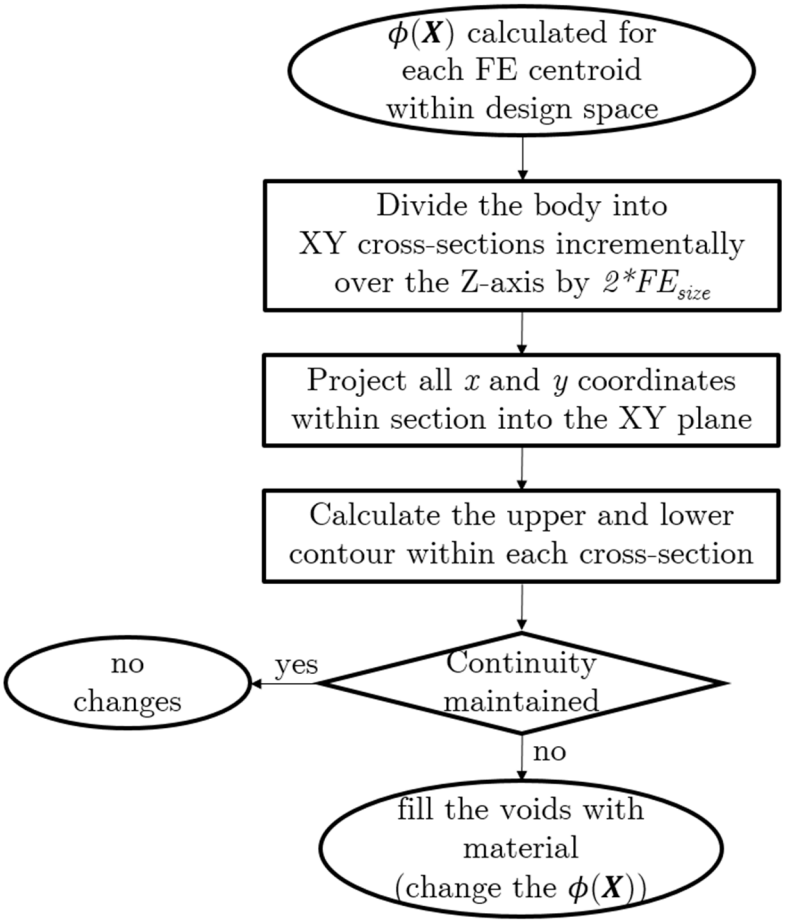

Figure 4.11. The forging feasibility filter - algorithm

The forging manufacturing constraints are met by default in the *projection method*, as the shape is modified along the forging direction. The only case in which feasibility issues may be observed is when the design space is concave in the XY cross-section with respect to the forging direction. However, such a problem is not associated with the shock absorber lower bracket optimization, therefore it does not require further attention, making the *projection method* less complicated compared to the *spatial LSM*.

5. Hybrid optimization parameters tuning

The optimization methods described in the previous chapter are able to provide an intricate shape of the automotive shock absorber lower bracket that is fulfilling the design requirements (stability) and manufacturing conditions (a shape that is achievable by the forging process). The main drawback of the proposed optimization methods is the computational effort needed to reach the optimal design (or at least quasi-optimal in the close neighborhood of the actual optimum). To take the most out of the computational resources available in a given optimization time (2000 objective function calls), a proper setting of the hybrid algorithm is required. This time limitation is caused by the narrow timing associated with the design development period. As genetic algorithms may be freely adjusted to the user needs, different combinations of genetic operators could be utilized. However, it may have a significant influence on the algorithm convergence rate and the achieved level of solution optimality. Researchers point out that efficient sizing of the population (in evolutionary optimization) is a critical aspect of getting increasing the algorithm efficiency within the fixed computational (simulation) budget [110]. They also highlight that a proper choice of GA settings is crucial to find a balance between the exploration and the exploitation capabilities of the algorithms. What is more, the significance of the mutation operator is highlighted as one of the primary features that enable to switch between local basins, which can be later exploited by the local-search part of the hybrid algorithm.

To find the set of genetic operators as well as the penalty function parameters, the DOE approach is utilized. This is the method commonly seen in the literature, however, the full factorial designs are the most widely adopted [75]. On the other hand, the usage of fractional factorial DOE instead of full factorial may significantly reduce the time of the analysis, due to decreased amount of required data. The main drawback of the fractional factorial approach is the aliasing of the parameters, i.e. when the interactions between certain factors are indistinguishable from the influence caused by the isolated parameters. The effect of aliasing may not be that problematic after all, as high-order interactions between genetic operators are rarely observed in the literature [71]. One phenomenon that cannot be captured correctly by both the full factorial and fractional factorial DOE is the nonlinear relationship between the parameters and the measured response. It was found during the parameter tuning process, that indeed there are factors presenting strongly nonlinear effects on the algorithm performance. Therefore, the Response Surface Method (RSM) approach was adopted in the studies, utilizing the Box-Behnken full-quadratic model fitting, as described in Chapter 3.2.2, without the need to run a highly expensive 3-level full factorial design. Still, the calculation

effort to perform such a study, including multiple repetitions of each parameter combination, is extensive and may not be feasible to complete such a study using expensive FEM-based simulations (as the ones presented in previous chapters). Therefore, the usage of surrogate-modeling is exploited, to greatly reduce the computation time. A comparison of both approaches to the hybrid optimization algorithm parameters tuning process is presented in Figure 5.1. The steps highlighted in red color represent the most time-consuming part of the process.

For the considered case, a purely FEM-based approach resulted in approximately 1.5 million function calls, as each of the parameter combinations must be verified 10x to include the random nature of the genetic algorithm, and a total of 73 different combinations were verified, which will be described in detail in the subsequent chapters. A single optimization run consists of 2000 objective function calls.

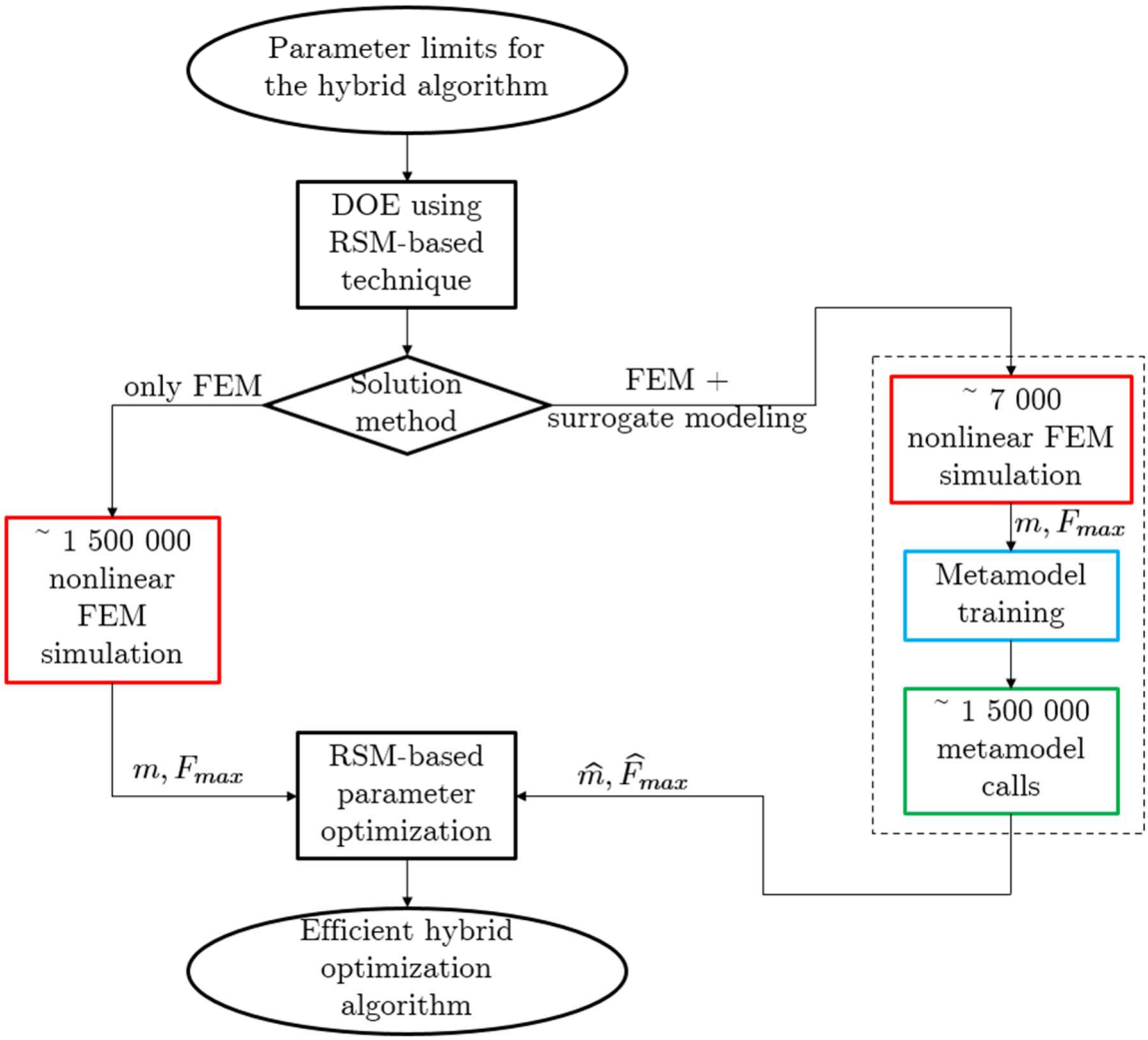

Figure 5.1. Hybrid optimization algorithm parameters tuning – alternative approaches

In the metamodel-based approach, the expensive FEM simulations are used solely to train the metamodels. In total, the metamodel training and validation required 7 000

FEM simulations, which is roughly 0.5% of the required computational effort of the previously described approach. Once the metamodels were trained and validated, they were called 1.5 million times to provide the response for the design points specified by the DOE (with the Box-Behnken sampling plan). The response, however, is barely an approximation, as it comes from the metamodel prediction. Still, it is very useful to predict trends and general behavior of the underlying functions or processes. The surrogates training and validation phase was done separately for both responses - the bracket mass m and for the load at the loss of stability F_{max}.

The total time needed to perform the step marked with green color is similar to a single nonlinear FEM simulation. However, the time needed to perform the step associated with training the metamodel (marked with blue color) is not negligible, even though it can be fully automated. In the considered case, hundreds of different artificial neural network topologies were tested along with the Kriging models.

5.1. Metamodel of the FEM simulation

The general idea behind metamodeling has been presented in Chapter 3.2. In this chapter, the most efficient surrogate model that represents the FEM-based simulation used for the optimization process (utilizing the spatial Level-Set method) is sought. To start the metamodel fitting process, the sets of training and validation data are needed. The Optimal Latin Hypercube Sampling (OLHS) method was used to generate separate sets of data for the training and validation phases. The sizes of the sampling plans were chosen to match the allowable time for data generation and desired level of stratification. The plans were constructed in such a way, that the ratio between the training data and validation data in each sampling plan is 4:1. As a result, two sets of plans were generated:

- First plan: 5400 training samples ($OLHS_1$) and 1350 validation samples ($OLHS_2$)
- Second plan: 1350 training samples($OLHS_2$) and 340 validation samples ($OLHS_3$)

The validation samples from the first plan served also as the training samples for the second, smaller plan, for which additional 340 samples were generated to fulfill the validation purpose. In this way, adding a minor amount of samples (340) enabled to obtain two sets of data, that could be then used to verify the impact of the sampling size on the quality of the metamodel (its ability for accurate predictions). As the number of variables in the considered spatial LSM optimization was 27 (parameters s_i scaling the ith radial basis function), the ratio between the number of available training samples and the number of variables in the process is 200 for the first plan and 50 for the second plan.

The metamodel training was conducted for different topologies of artificial neural networks and Kriging. The performance of the metamodel was compared using the coefficient of determination - R^2 defined by equation (5.1), calculated for the validation set. Therefore, the quality of the surrogate prediction could be verified on the previously unseen data. Using the R^2 obtained for the training data could be misleading due to two reasons: firstly, interpolators like the Kriging always reflect the training data perfectly ($R^2 = 1$) even though their predictive capabilities may be limited. Secondly, when surrogates show high R^2 for the training data with low values for the validation data, usually, it means that the model is overfitted, as shown in Chapter 3.2.

$$R^2 = 1 - \frac{\sum_{i=0}^{n_t}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=0}^{n_t}(y^{(i)} - \bar{y})^2} = 1 - \frac{MSE}{Var(y)} \tag{5.1}$$

The MSE represents a mean square error and $Var(y)$ is the variance of the actual, not predicted response. The comparison of the obtained R^2 is shown in Table 5.1. The coefficient of determination is one of many different metamodel metrics, yet it was used primarily in literature studies [115]. The values highlighted with bold font are associated with the surrogates that showed the best performance among all of the analyzed models. They were also used for further metamodel-based studies of the hybrid algorithm parameters.

Table 5.1. Resultant coefficients of determination for validation data (best-fitted models) [95]

Response	***Fmax***		***Mass***	
Model	**ANN**	**Kriging**	**ANN**	**Kriging**
First plan	**0.765**	0.716	0.942	**0.960**
Second plan	0.698	0.620	0.926	0.940

For the *Mass* response, the anisotropic Kriging was found to achieve the highest values of the coefficient of determination R^2 among the analyzed metamodels. In the case of the F_{max} response, the artificial neural network with two hidden layers was the best-performing metamodel. It consisted of the 27-9-39-1 architecture, with rectified linear units (ReLU) activation functions in both hidden layers. The usage of regularization and normalization did not bring much improvement in the model fitting, as the R^2 oscillated around the same values as in the model without those features. However, the process of finding the best network topology could be avoided when the drop-out method was incorporated. The drawback of such a method is the need to use bigger network sizes, i.e. the ANN with 27-216-216-1 neurons and drop-out probabilities of 0.1 and 0.25 (for 1st and 2nd hidden layer units respectively) achieved the R^2 of 0.758, which is close to the aforementioned 27-9-39-1 ANN without the dropout. Even if seeking the most effective topology is not performed, the considered drop-out-based

ANN has much more weights and biases to optimize, therefore the training process is greatly extended. The achieved coefficient of determination for validation data, for different topologies of $27 - hn_1 - hn_2 - 1$ network for the F_{max} response is shown in Figure 5.2 (where hn_i represents number of neurons in the ith layer of the network). Incorporation of the normalization in the ANN, i.e. replacement of the linear unit with a sigmoidal activation function in the output layer, yielded almost the same results as the cheaper-to-evaluate ANN with the linear unit.

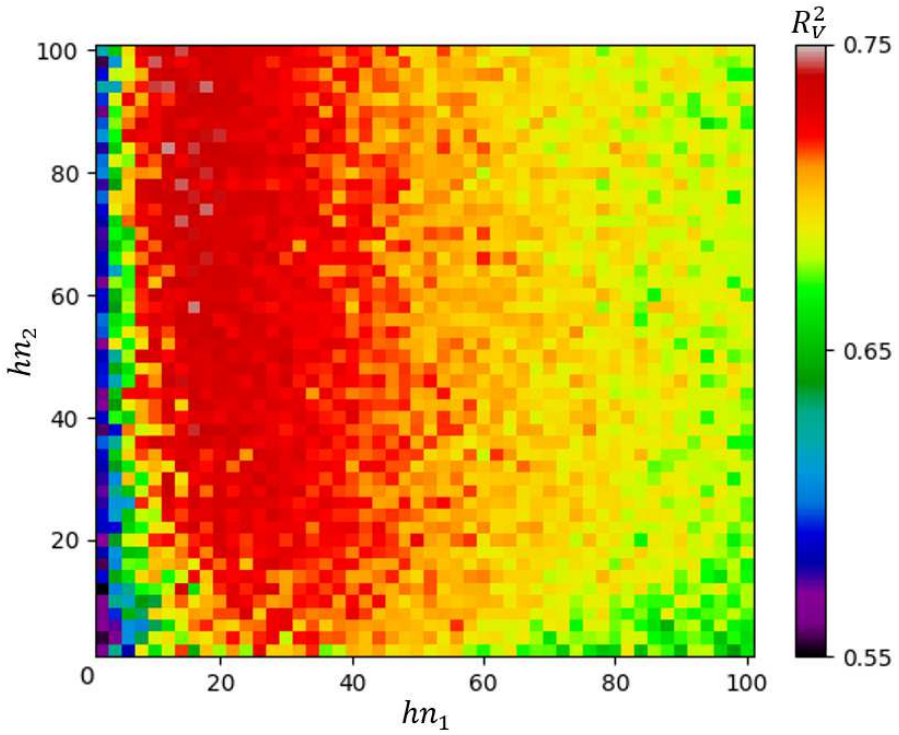

Figure 5.2. The average validation data R^2 for different network topologies (ReLU neurons in both hidden layers and linear output neuron)

The choice of the algorithm optimizing weights and biases of the network greatly influence the training time and achieved R^2. The algorithms described in Chapter 3.2.4 were used to perform the ANN training. The one found as the most efficient was the L-BFGS quasi-Newton algorithm, and the convergence of the ANN training is shown in Figure 5.3Figure 5.3. ANN training for the F_{max} response. Even though the resultant values of the coefficient of determination were very similar among the aforementioned algorithms, the L-BFGS was able to train the network in 50-100 epochs, while the NADAM and SGD algorithms required 2000-8000 and 8000-16000 epochs respectively. The training process was repeated for different hyperparameters values, like the learning rate η_1, momentum η_2, decay, etc., and the highest R^2 among different algorithm hyperparameters values was reported for a particular network topology.

The size of the training set played also a significant role in the process, as firstly – the training takes more time (especially for the Kriging method) once the number of

training samples grows. Secondly, the quality of all metamodels decreased with the lower number of samples. Yet the *Mass* response achieved only a negligible drop in surrogates performance (~2%). The F_{max} response, on the other hand, showed high sensitivity to the number of training samples, as the R^2 dropped 9%-13% for both examined metamodels when the smaller dataset was used for the training.

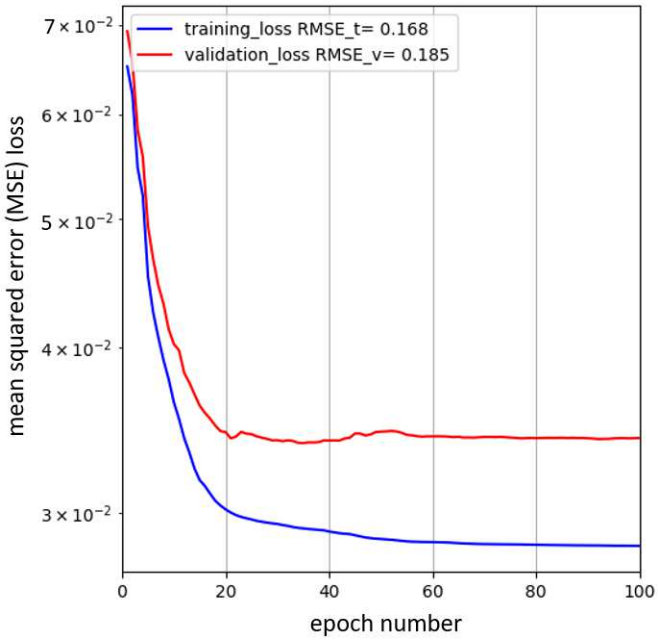

Figure 5.3. ANN training for the F_{max} response [95]

Each of the metamodel training processes was repeated 10x times, as the initial weights and biases in ANNs, and initial theta values in the Kriging, are chosen randomly. Therefore, as the model fitting process is dependent on the starting values, different results were obtained, even though deterministic algorithms were used to find the optimal weights, biases, and thetas. The best-fitted model, among all the repetitive iterations, was reported for a particular network topology or Kriging model.

Two surrogate models are described in this chapter, i.e.:

- ANN with 2 hidden layers for F_{max} response, and
- anisotropic Kriging for the *Mass* response

were used to mimic the behavior of the nonlinear FEM analysis, in which the shape of the bracket was modified, according to the given RBF knots scale factors s_i. Therefore, once the 27-knot values (s_i) were given, a *Mass* of the bracket and the force at the loss of shock absorber stability F_{max} could be calculated in a fraction of a second. Such a combination of surrogate models could be used to perform the tuning of the hybrid optimization algorithm parameters.

5.2. Hybrid optimization algorithm parameters definition

The proposed hybrid optimization algorithm has 9 parameters in total that can be tuned to achieve the highest efficiency. Current research papers highlight that the choice of the appropriate degree of crossover and mutation operators should be an essential part of the research in the field of evolutionary optimization [52]. Most of those parameters are associated with genetic operators, however, two of them concern penalty function. If all of those parameters were tuned together, the Box-Behnken sampling plan would consist of 130 parameter combinations. To reduce the size of the RSM design plans, the study (parameter tuning) was divided into two steps.

The aim of the first stage was to optimize the common genetic operators associated with both versions of genetic algorithms (multi-island and single population GA) – crossover, mutation, and number of individuals in the population. Also, the penalty function parameters were included in this study, as the level of penalization directly influences the chance of infeasible designs being used as parents for further generations in both of the considered evolutionary algorithms. As the total number of objective function calls in the proposed algorithm was fixed to 1600 for the GA, one parameter (referred to as *Population ratio*) was sufficient to describe the resultant structure of the population, as shown in (5.2)-(5.3):

$$R_{pop} = \frac{n_{GA_{ind}}}{n_{GA_{gener}}} \tag{5.2}$$

$$n_{GA_{gener}} = \sqrt{\frac{1600}{R_{pop}}} \tag{5.3}$$

The list of the tunable parameters and their ranges of variability for the first stage of the RSM DOE is shown in Table 5.2.

Table 5.2. Parameters for DOE performed on single population GA and EVOL [95]

No.	Symbol	Parameter name	Min level	Max level
1	R_{pop}	Population ratio	0.16	1
2	p_{GA_co}	Crossover rate	0.5	1
3	p_{GA_mut}	Mutation rate	0.01	0.1
4	P_m	Penalty multiplier	10	1 000
5	P_e	Penalty exponent	1	3

The aim of the second stage was to optimize four genetic operators associated strictly with the multi-island version of the GA, as listed in Table 5.3. The structure of the total population is a function of two variables, as having fixed 1600 objective function calls, only two out of three parameters could be mutually exclusive, as shown in (5.4).

$$n_{GA_{isl}} * n_{GA_{ind}} * n_{GA_{gener}} = 1600 \tag{5.4}$$

The other (common) genetic operators, as well as the penalty function parameters, were held fixed on the levels resulting from the first study.

Table 5.3. Parameters for DOE performed on multi-island GA and EVOL [95]

No.	Symbol	Parameter name	Min level	Max level
1	n_{GA_isl}	Number of islands	2	4
2	n_{GA_gener}	Number of generations	40	100
3	p_{GA_migr}	Migration rate	0.05	0.2
4	n_{GA_migint}	Migration interval	4	20

Therefore, instead of 130 parameter combinations, the two-stage RSM DOE approach resulted in 73 parameter combinations – 46 for the first stage, and 27 for the second stage. The ranges of variability for the parameters were adjusted based on literature studies [40]. All of the analyzed parameters maintained constant during a single operation of the algorithm (no adaptive formulations were included).

Due to the stochastic nature of the evolutionary optimization, each of the hybrid algorithm runs with adjusted parameters (according to the RSM DOE design points) was repeated 10 times, and the average best individual was reported. The response of each of the parameters combinations was then used to optimize the hybrid algorithm parameters using the RSM. The coefficients of determination for both fitted fully-quadratic response surfaces are 0.88 and 0.82 for the single- and multi-population version of the hybrid algorithm respectively.

5.3. Hybrid optimization algorithm parameters tuning

Once the polynomial regression surfaces are fitted to the provided data, the influence of each parameter or its interactions with other parameters is directly visible from the polynomial coefficients vector $\boldsymbol{\beta}$. As the fully-quadratic model is used, the nonlinear response may be reflected during data analysis, i.e. it could be visible which of the hybrid optimization algorithm parameters causes the major impact on the bracket mass minimization capabilities and what is the nature of this input-output relation.

The analysis of the main effects plot (for averaged responses) from the first study (Figure 5.4) shows, that the rate of mutation $p_{GA_{mut}}$ has the most significant influence on the final value of the objective function. Not only the influence is significant in terms of observed impact on the objective function, but it is also strongly nonlinear. The parameters associated with the penalty function show a nonlinear influence on the performance of the hybrid optimization algorithm, but their impact is lower than the one from the mutation rate, yet still important. Surprisingly, the influence of the

population parameter R_{pop} (ratio between individuals and generations) and the crossover rate $p_{GA_{co}}$ is statistically insignificant for the considered case.

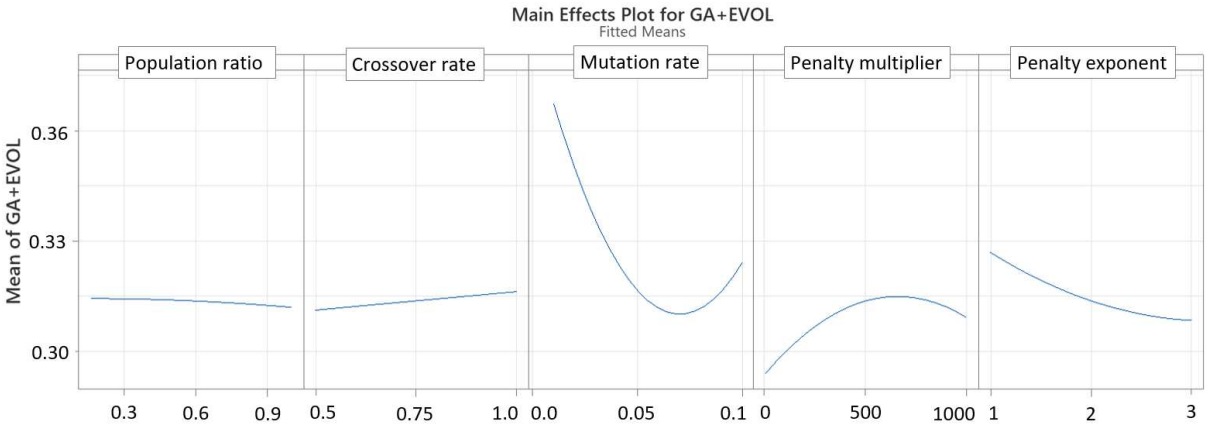

Figure 5.4. The main effects plot for the hybrid optimization using single-population GA+EVOL [95]

The effects of the penalty functions are not as widely studied in the literature, as the genetic operators. However, in the study performed for the proposed hybrid optimization, the penalty function not only showed a significant impact on the objective function but also the interactions between penalty parameters proved to be strong and nonlinear. The dependency between the penalty multiplier and penalty exponent and their impact on the final objective function value is shown in Figure 5.5. While increasing the penalty multiplier, the opposite trend may be observed when the penalty exponent is changed from $P_e = 1$ to $P_e = 3$, i.e. $P_e = 1$ showed decrease in the final objective function with an increasing penalty multiplier P_m, while $P_e = 3$ showed increase in the final objective function. Therefore, the choice of one of the penalty parameters should not be performed without consideration of the second parameter. Penalty function parameters are the only two parameters directly influencing the behavior of both – the GA and the EVOL.

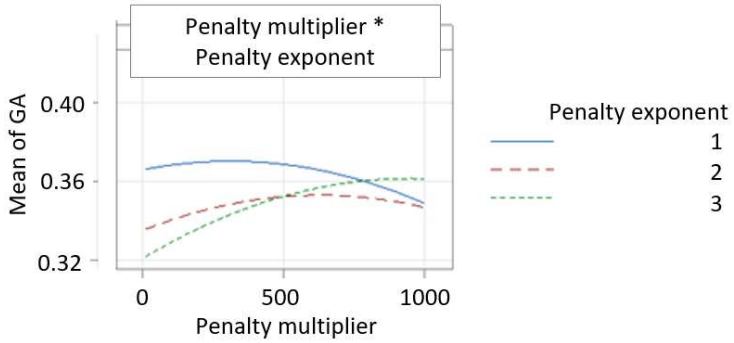

Figure 5.5. The interaction plot for the penalty function parameters [95]

As the response surface method utilize the full-quadratic polynomial regression model, finding the minimum value of the five-variable function can be performed efficiently and precisely. The outcome of such optimization is shown in Figure 5.6.

The resultant, tuned parameters of the hybrid algorithm form a genetic algorithm with a single population of $n_{GA\_ind} = 16$ individuals, reproduced over $n_{GA\_gener} = 100$ successive generations, with the probability of the individual crossover $p_{GA\_co} = 0.5$ and probability of the gene mutation $p_{GA_{mut}} = 0.059$. The penalty function is strongly nonlinear, with the multiplier $P_m = 10$ and exponent $P_e = 3$.

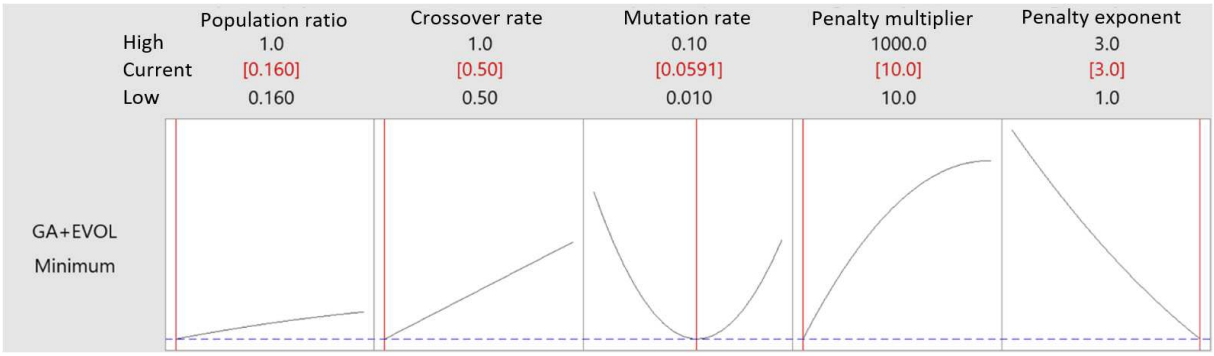

Figure 5.6. The optimal values of the considered parameters for the hybrid optimization algorithm with single population GA + EVOL [95]

To verify the correctness of the tuned parameters, the metamodel-based hybrid optimization was rerun with the adjusted values (as per Figure 5.6). This optimization was as well repeated 10 times to include the immanent randomness associated with the nature of bio-inspired algorithms. The convergence of the algorithm is shown in Figure 5.7. For clarity purposes, the objective function (mass) is shown as a function of the number of individuals (objective function calls), instead of the number of generations. The dashed vertical line represents the transition between the genetic algorithm and evolution strategy, while the blue dashed line represents the average objective function (from the analyzed 10 runs). The resultant mass of the average best individual was 0.332 after 100 generations of the genetic algorithm and furtherly reduced to 0.294 after 100 generations of evolution strategy (each consisting of 4 individuals).

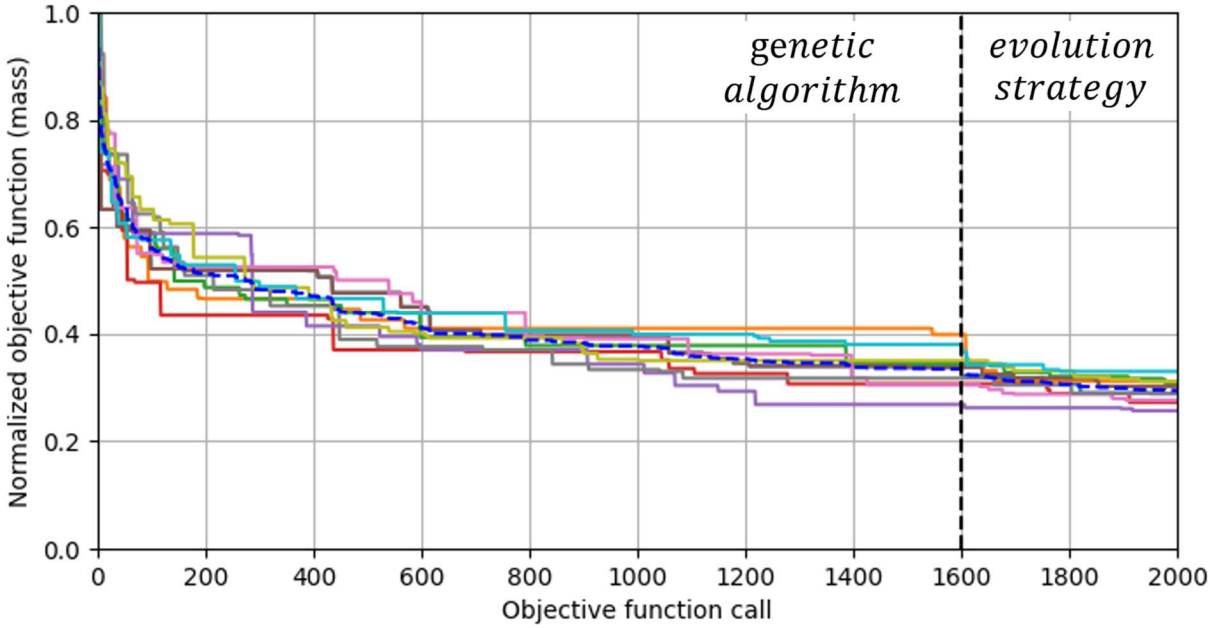

Figure 5.7. The convergence of the hybrid optimization after tuning – the single population version of the algorithm

All of the aforementioned tuned parameters, besides the population ratio, are then used in the second stage of the RSM optimization. The first parameter – population ratio – is not used, as the multi-island version of the GA consists of different structures of the population.

The hybrid algorithm with the multi-island version of the genetic algorithm was analyzed in terms of the main effects, as shown in Figure 5.8. The only parameter that was found to be statistically significant, was the number of islands $n_{GA_{isl}}$. Yet the number of generations, migration rate, and migration interval again showed a nonlinear impact on the final objective function value. The number of islands (autonomous populations) had the highest impact on the observed optimization capabilities, with fewer islands resulting in better optimization capabilities. Even though the number of islands showed the highest impact on mass minimization capabilities, its influence was found to be almost linear.

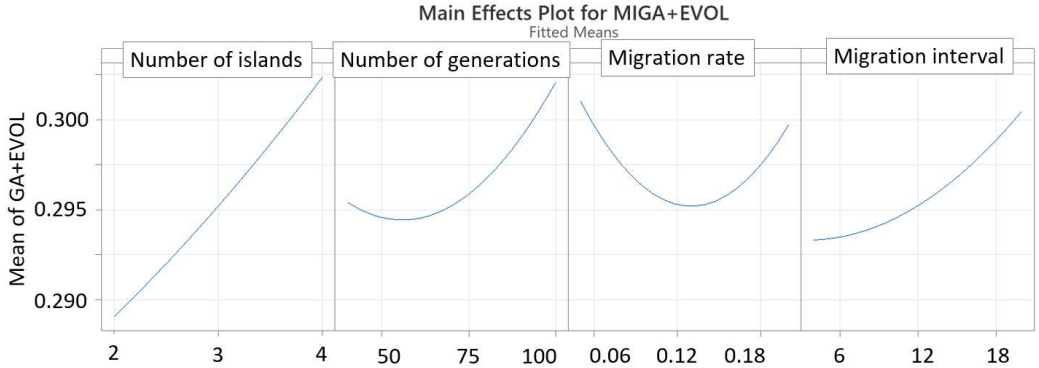

Figure 5.8. The main effects plot for the hybrid optimization using multi-island GA+EVOL [95]

Again, strong interactions between parameters were found. Two of the statistically significant ones were highlighted with red borders in Figure 5.9.

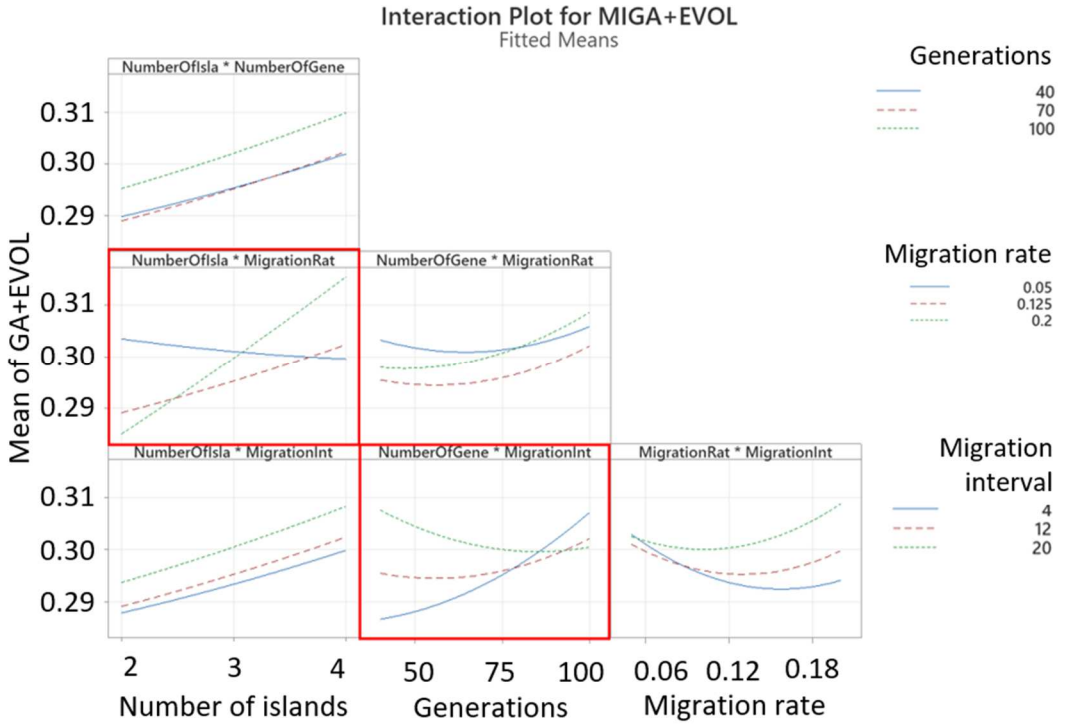

Figure 5.9. The interaction plot for the multi-island genetic algorithm parameters [95]

Firstly, the number of islands and the migration rate showed mutual impact, where a high migration rate p_{GA_migr} occurred to be more effective for two islands, while a low migration rate was more effective for a higher number of autonomous populations. Secondly, the interaction between the migration interval $n_{GA_{mint}}$ and the number of generations $n_{GA_{gener}}$ showed the opposite trends when the extreme values of the parameters were analyzed. For a multi-island genetic algorithm with a relatively low number of generations ($n_{GA_{gener}} = 40$), frequent migration occurred to be efficient. On the other hand, for populations consisting of a small number of individuals, yet including a high number of generations ($n_{GA_{gener}} = 100$), less frequent migration provided higher optimization efficiency.

Similarly to the hybrid algorithm with single-population GA, the parameters of the hybrid algorithm with multi-island GA and EVOL were tuned by the RSM optimization process. In this case, all of the analyzed parameters were fixed at one of their extreme values. The optimal set of the algorithm consisted of $n_{GA\_isl} = 2$ autonomous islands, $n_{GA\_gener} = 40$ generations and migration rate $p_{GA\_migr} = 0.2$ performed in intervals of $n_{GA\_migint} = 4$ generations.

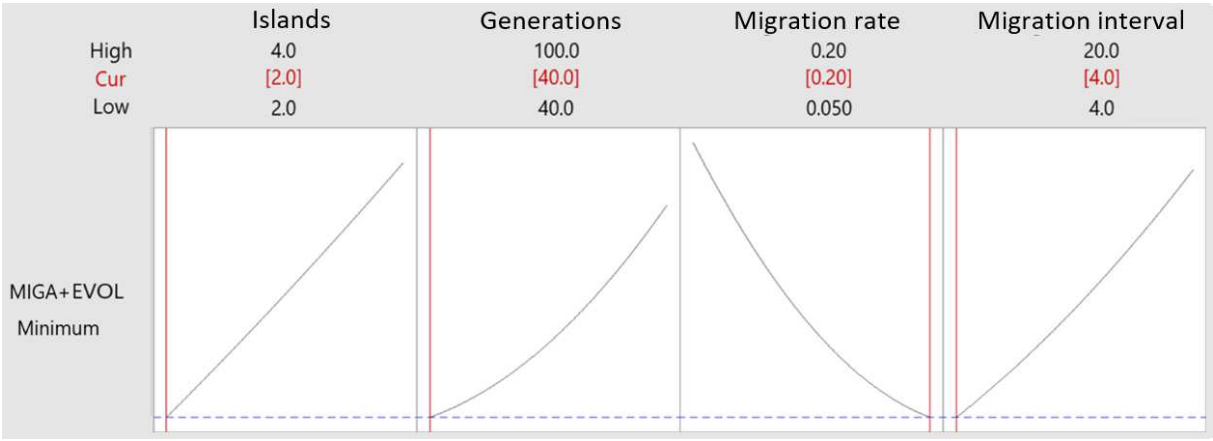

Figure 5.10. The optimal values of the considered parameters for the hybrid optimization algorithm with multi-island GA + EVOL [95]

Again, the MIGA-based hybrid algorithm with tuned parameters was run 10 times to verify the impact of the adjusted parameters on the algorithm optimization capabilities. The convergence is again shown as a function of the number of individuals (or the number of objective function calls) in Figure 5.11. After the GA phase, consisting of 40 generations, the average objective function value was 0.333. Subsequent 100 generations of the evolution strategy (each consisting of 4 individuals) furtherly reduced the objective function to 0.285.

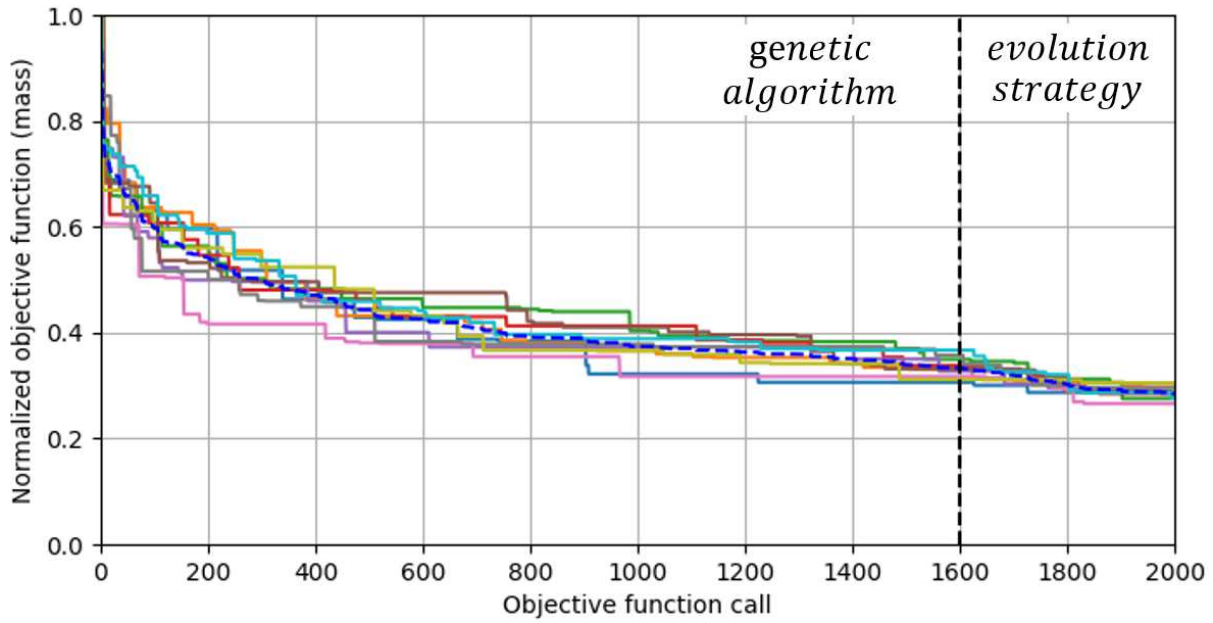

Figure 5.11. The convergence of the hybrid optimization after tuning – the multi-island version of the algorithm

The last part of the study was the verification of the genetic algorithm performance without the local search performed by the evolution strategy in order to justify the algorithm hybridization. To maintain the same number of allowable objective function calls, the operation of the genetic algorithm was extended from

$n_{GA\_iter} = 1600$ to $n_{GA\_iter} = 2000$. The study was performed for both – the single-population GA and the multi-island GA. With the additional 400 function calls, the GA could utilize either additional individuals in the populations or additional generations to compensate for the lack of an evolution strategy. The metamodel-based optimization of all considered combinations was performed, with 10 repetitions of each combination. The results are shown in Figure 5.12, where the dashed lines represent the average objective function from the 10 repeated runs. The structure of the particular genetic algorithm is described symbolically, i.e. in the form of $n_{GA_{isl}} x\ n_{GA_{ind}}\ x\ n_{GA_{gener}}$.

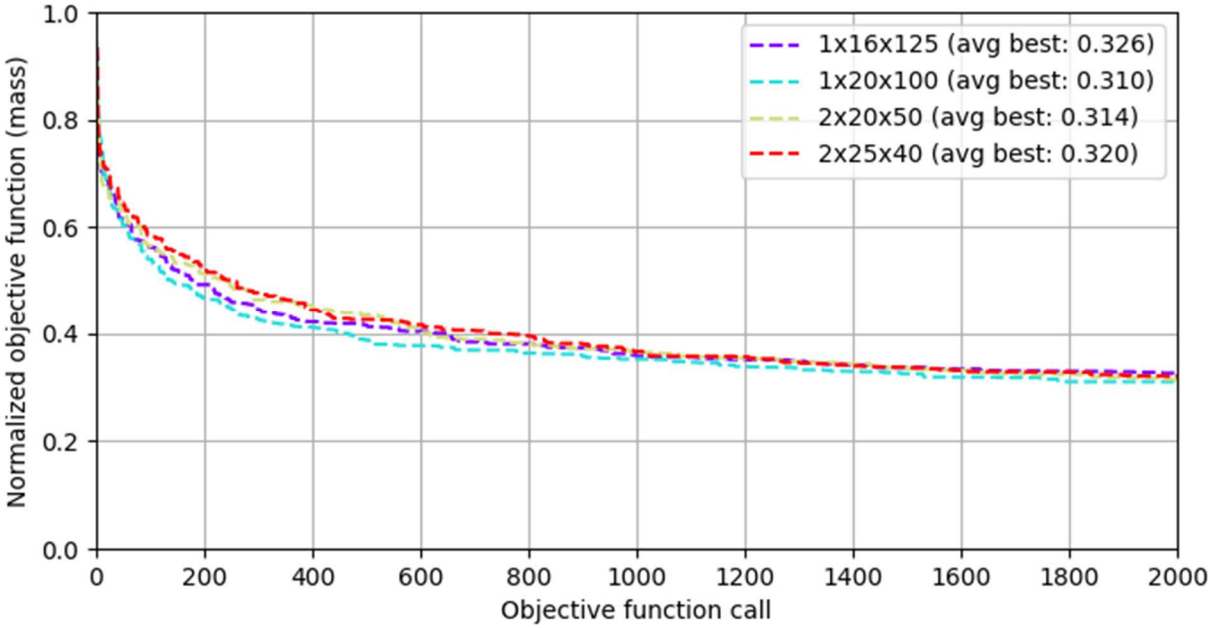

Figure 5.12. Genetic algorithm with 2000 objective function calls – average convergence from four runs with different structures of the population

The comparison of the average best individuals from each of the algorithms is shown in Table 5.4. It is clearly seen that the objective function value (the mass of the bracket) is the lowest for the proposed hybrid algorithm. The differences between the average of 10 runs for a hybrid algorithm incorporating a single population and a multi-island genetic algorithm are barely noticeable. On the other hand, the average best individuals coming from the application of solely genetic algorithm have higher mass and higher spread in the results.

Table 5.4. Average best individuals for different algorithm properties

Strategy	GA				EVOL	Average best
	Islands	Individuals	Generations	Total runs	Total runs	
Hybrid - single population	1	16	100	1 600	400	0.284
Hybrid - multi island	2	20	40	1 600	400	0.285
SPGA, more generations	1	16	125	2 000	0	0.326
SPGA, more individuals	1	20	100	2 000	0	0.310
MIGA - more generations	2	20	50	2 000	0	0.314
MIGA - more individuals	2	25	40	2 000	0	0.320

The genetic algorithm with a single population (SPGA) performed better when more individuals were added to the population. On the other hand, the multi-island version of the genetic algorithm (MIGA) converged to the lower objective function value when more generations were included in the evolution of the population. Those observations might lead to the conclusion, that the SPGA is underpopulated, yet with sufficient steps of evolution (generations), while the MIGA, which incorporates more randomness in the design search (multiple populations, more individuals), on the contrary, has an insufficient number of generations that allow the individuals to cross and mutate. Those inferences are strictly connected to the constraint of having only 2000 objective function calls, which is severely limiting the allowable evolution of the population, yet it is constrained by the allowable algorithm execution time.

6. Shock absorber optimization methodology – implementation examples

Two industrial examples were used to verify the ability of the proposed optimization methods and algorithms to minimize the mass of the automotive shock absorber while fulfilling the constraint related to the system stability. Both examples were verified using the spatial LSM and the projection method, as well as metamodel-based optimization.

6.1. Optimization of the passive shock absorber

The first example concerned the passive shock absorber design that was used in the previous chapters to present the general optimization methodology. The subsequent chapters contain model information, two methods of solution (optimization), and verification of the chosen design using the final shape of the part, after the adjustment and smoothing using CAD software.

6.1.1. The passive shock absorber model

The geometry of the passive shock absorber was presented in Figure 4.1. The shock absorber consisted of three major structural parts that are included in the analysis: the lower bracket (the aim of the topology optimization), the tube, and the rod. For all of those components, the elastic-plastic material behavior with hardening was assumed. The maximum force that the structure was able to withstand (having the material allocated within the whole design space) without losing stability was normalized to 1.0. The F_{limit} for the considered case was 0.833. The mass of the total allowable design space was also normalized to 1.0, while the aim of the optimization was to minimize the mass of the bracket while fulfilling the structural and manufacturing constraints. Both responses, similarly to other quantities presented in previous chapters, were normalized to avoid the order-of-magnitude differences in fitness function calculations, like mass [kg] and force [N], which differ in magnitude by a factor of ~50 000. On top of that, normalization is suitable for maintaining business confidentiality.

6.1.2. Topology optimization - spatial LSM

The first topology optimization was done using the spatial version of the level set method. The interpolation knots distribution was realized as shown in Figure 4.6

(with 3x3x3 knots in X, Y, and Z directions). Each design variable (radial basis function scaling factors) was also normalized to a range of [0,1]. Both hybrid algorithms were used in the optimization to verify their performance – one with a single population (SPGA hybrid algorithm) and the second with a multi-island implementation of the genetic algorithm (MIGA hybrid algorithm). All of the genetic operator parameters as well as penalty function parameters are the result of the tuning process shown in Chapter 5.3.

Both SPGA and MIGA-based hybrid algorithms were run 3 times to explore the randomness of the search. The convergence of those algorithms, as well as the average response of each of those algorithms, are shown in Figure 6.1. In the considered example, the difference between SPGA and MIGA average resultant mass was 0.4% after the genetic algorithm operation, and 1.7% after the evolution strategy. The MIGA-based optimization achieved the lowest final mass of the bracket and was also more efficient on average.

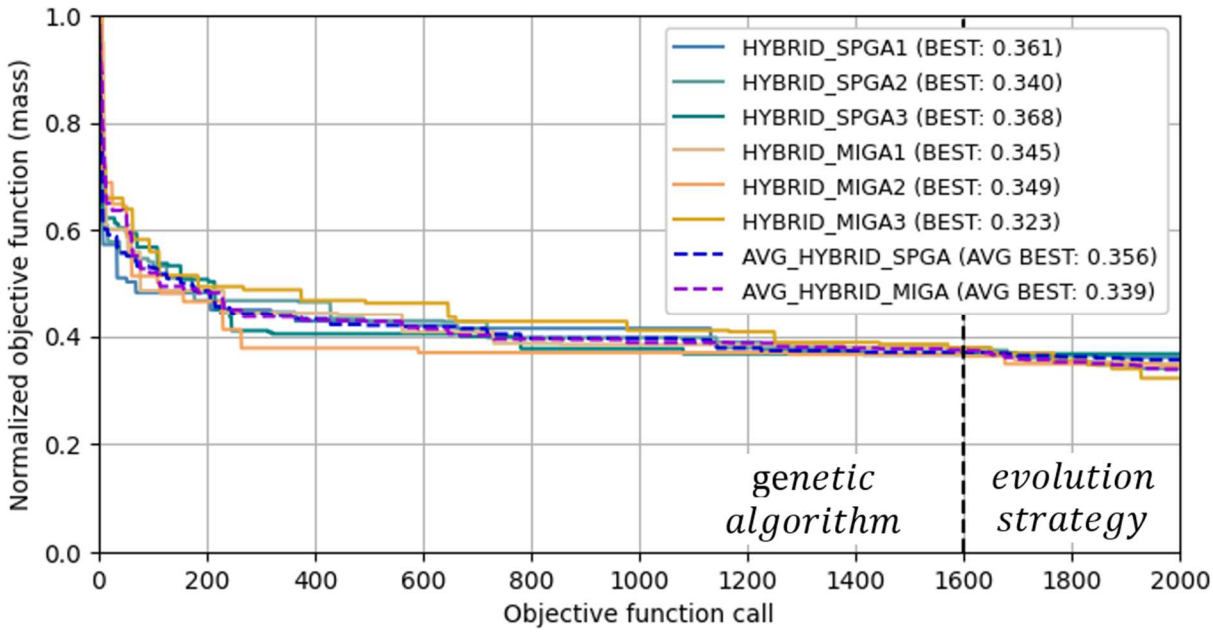

Figure 6.1. The convergence of the hybrid optimization algorithm – spatial LSM for the passive shock absorber constrained mass minimization

The evolution of the shape during the *Hybrid SPGA*1 optimization is shown in Figure 6.2. The first 7 shapes represent the evolution during the genetic algorithm operation, while the last two shapes reflect the changes induced by the evolution strategy. Only the individuals fulfilling the stability constraint are shown. The yellow areas of the bracket are non-design areas, i.e. the attachment of the tube (welding zone) and the attachment of the lower bushing.

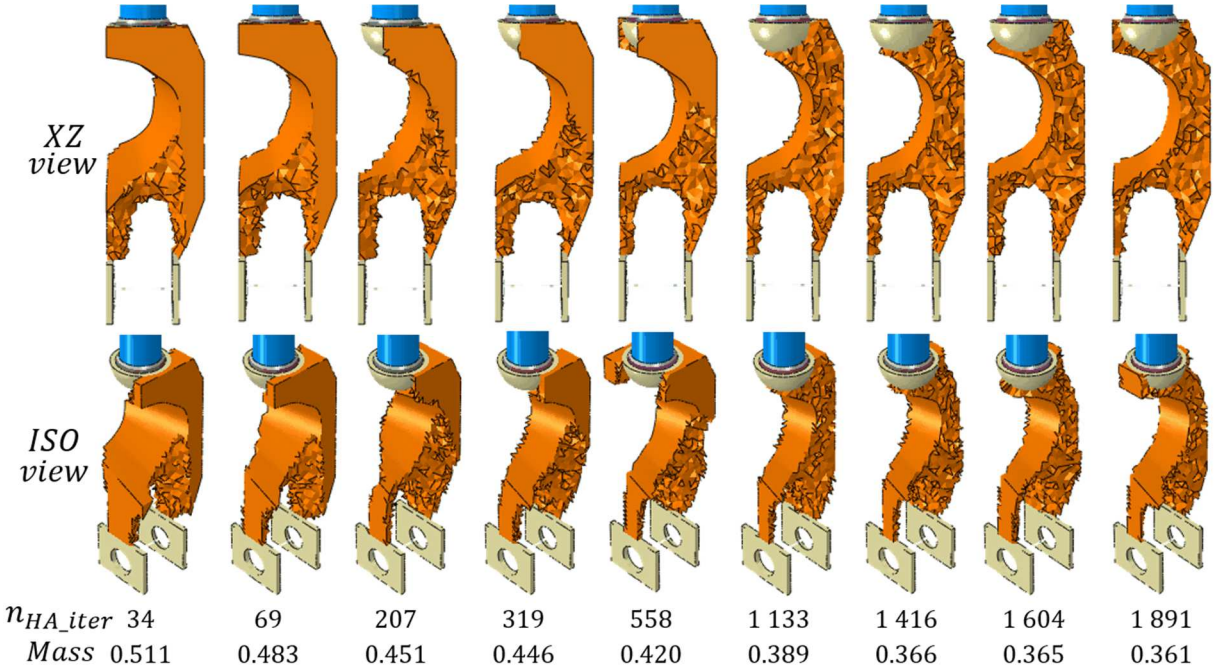

Figure 6.2. The evolution of the bracket topology – *Hybrid SPGA*1 run

The evolution of the shape during the *Hybrid MIGA*1 optimization is shown in Figure 6.3:

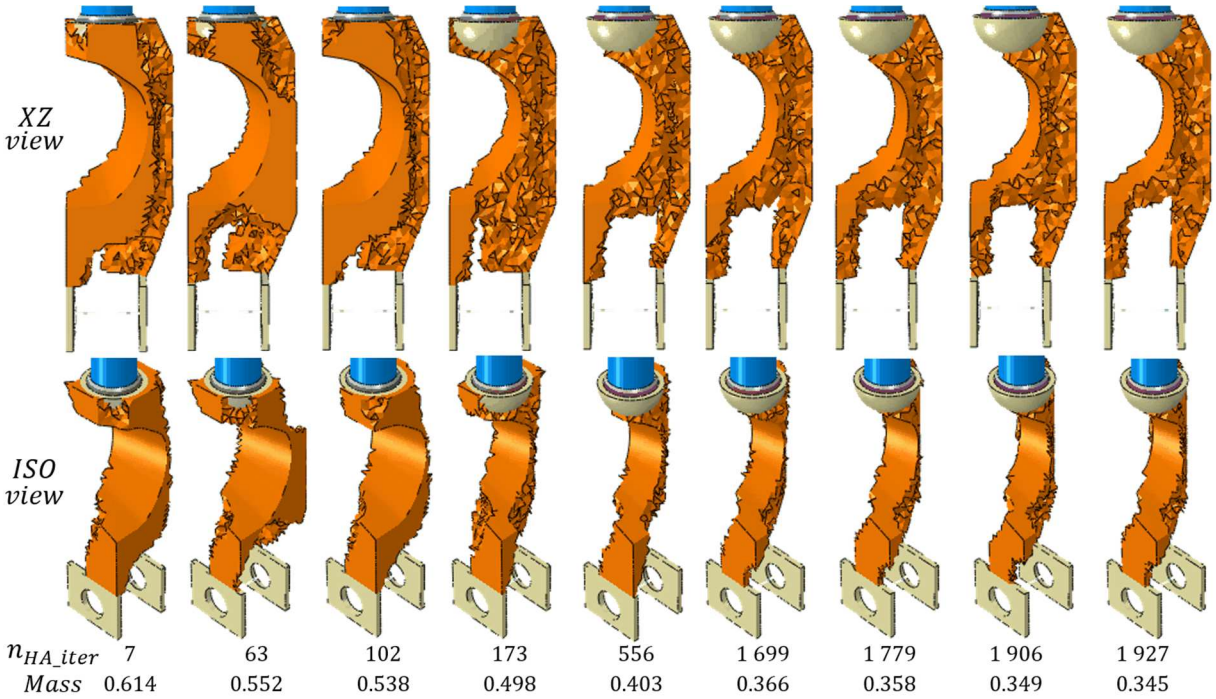

Figure 6.3. The evolution of the passive shock absorber bracket topology – *Hybrid MIGA*1 run

This time, the first 5 shapes reflect the changes induced by the genetic algorithm, while the last 4 are the effect of the evolution strategy operation.

6.1.3. Topology optimization - projection method

The second method of optimization that was used to minimize the mass of the bracket while fulfilling the stability constraint was the *projection method*. However, the simplicity of this method allowed for the implementation of efficient surrogate models, as with a limited number of objective function calls (needed to create the training and validation data), the metamodels were able to reflect the underlying simulation system precisely. For a considered case of the passive shock absorber optimization, the mass minimization was performed solely with the Kriging surrogate models to predict the $Mass$ and F_{max} design responses. The reason for the choice of the Kriging instead of ANN is the lack of need for finding the optimal topology of the ANN, its activation functions, etc., with comparable efficiency in prediction.

Three sets of data were used to verify the metamodel prediction quality: 4 400, 1 100, and 275 samples, each consisting of two separate OLHS sampling plans with a 4:1 ratio between training and validation data (as described in Chapter 5.1). The coefficients of determination (for validation data) of the resultant Kriging models after the fitting process are shown in Table 6.1 for the aforementioned sampling plans. The sizes of the sampling plans were chosen to represent the same ratio between the number of variables and the number of samples as in the case of metamodels used for tuning hybrid optimization algorithm parameters.

Table 6.1. The coefficient determination for Kriging (validation data) for various sampling plan sizes – passive shock absorber optimization

Sampling plan size	**4 400**	**1 100**	**275**
$R_V^2 - Mass$	0.997	0.995	0.989
$R_V^2 - F_{max}$	0.997	0.997	0.987

The Kriging *Fmax* predictions vs target data (for training and validation data sets) for the max and min sampling plans are shown in Figure 6.4. The coefficient of

determination for the training data is equal to 1.0, as the Kriging interpolates the data, so all of the training points are reflected precisely.

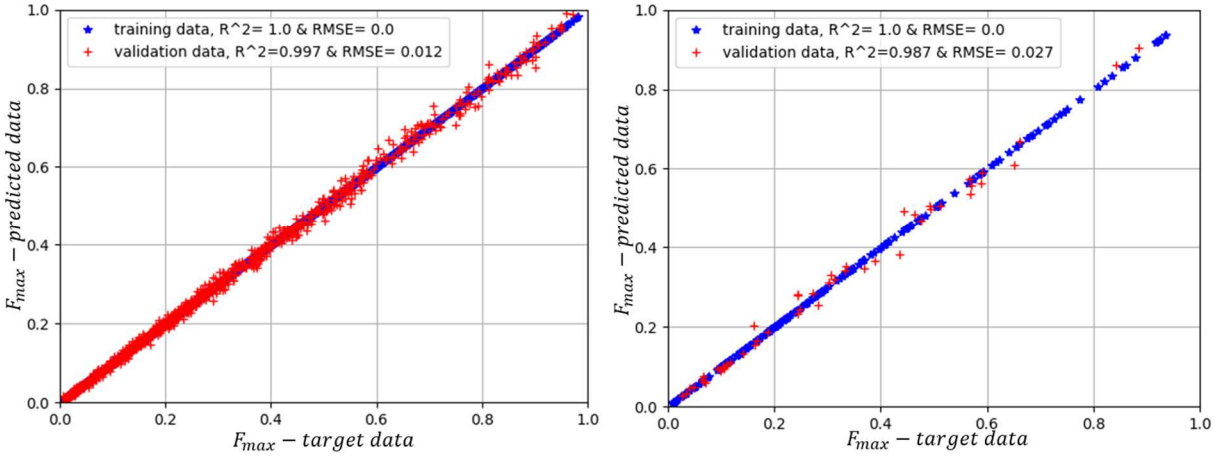

Figure 6.4. The Kriging prediction - F_{max} – for 4 400 and 275 samples used for OLHS – passive shock absorber optimization

The optimization was done using the same hybrid algorithm that was used in the case of the *spatial LSM*. The results of the optimization are shown in Figure 6.5. The algorithm converges quicker compared to the plots from the *spatial LSM* optimization. Also, the spread between runs is lower compared to the more sophisticated, spatial variant of the optimization method.

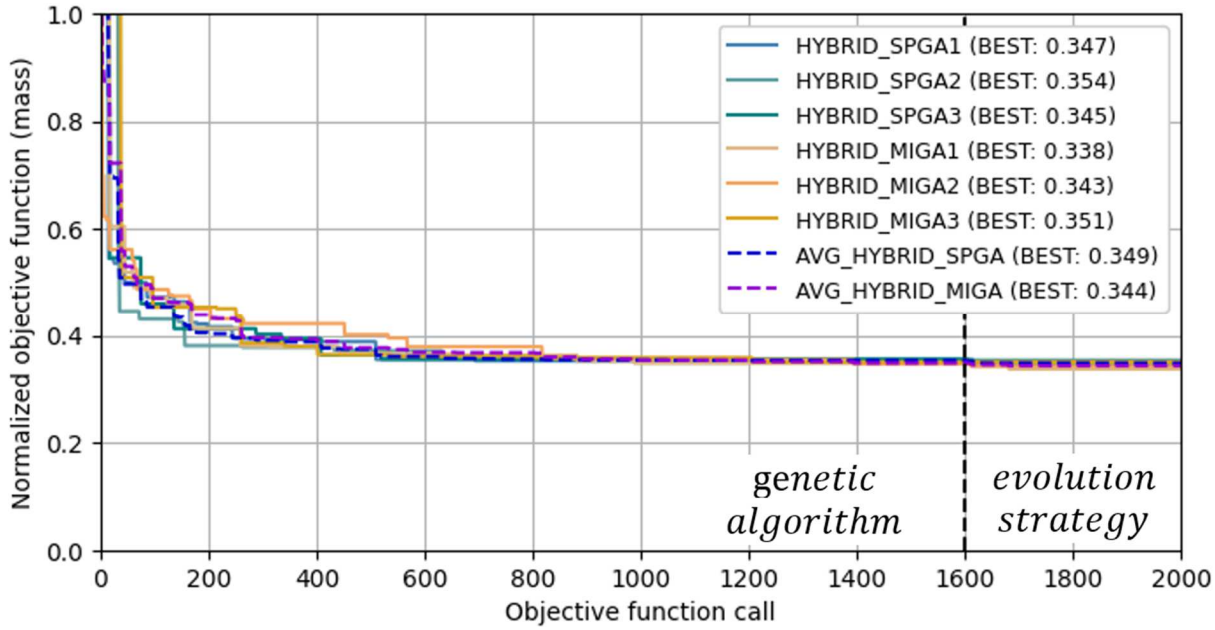

Figure 6.5. The convergence of the hybrid optimization algorithm – projection method for the passive shock absorber constrained mass minimization

The *projection method* was not able to outperform the *spatial LSM* in terms of the best-fitted individuals, even though the average best-fitted individual for *Hybrid SPGA* had a lower final mass compared to the *spatial LSM* counterpart.

From all of the obtained results, the *Hybrid MIGA* optimization incorporating the *spatial LSM* was most efficient in the constrained mass minimization.

6.1.1. Verification of results

The result of the optimization is a FEM model with coarse edges and geometrical imperfections, coming from the hard-kill optimization method and passable resolution of the contour, resulting from the used FEM discretization. In order to verify if the shape of the real bracket obtained using the proposed method was fulfilling the strength requirements, the CAD model was created based on the best individual from the *Hybrid MIGA*1 *spatial LSM* optimization. The CAD model was closely reflecting the obtained FEM model, with adjustments to the manufacturing details that were not reflected precisely in the FEM model (transition radii, member size, curvature, etc.). The comparison of the CAD and FEM models is shown in Figure 6.6. The shapes of the non-design areas (such as the lower part of the braces or the upper attachment with the tube) were adjusted to match the optimal layout of the bracket.

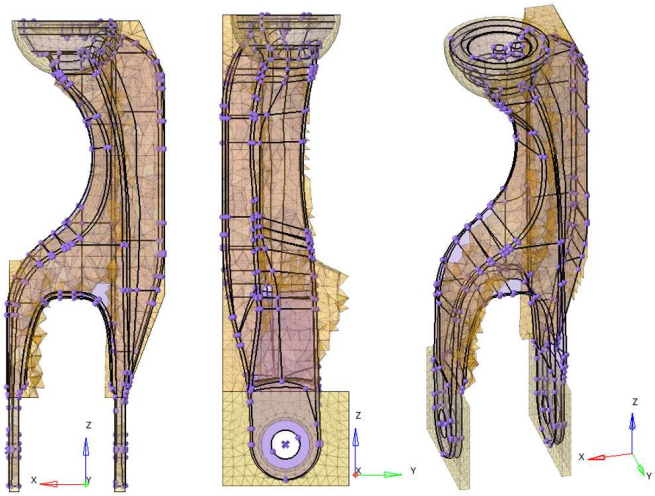

Figure 6.6. The comparison between the CAD and the FEM models – passive shock absorber after topology optimization

The same boundary conditions were used as in the optimization analysis, yet the shock absorber model was more detailed. The lower connection was changed from rigid to deformable body, the rod with a piston were modeled using 3D solid elements and the spring characteristic was nonlinear. Firstly, the load-deflection response of the system was measured, as shown in Figure 6.7a. Once the shock absorber was preloaded (lower bushing assembly loads and spring preload), the jounce bumper load was added incrementally. This part of the analysis was displacement-driven, so the reported total axial load was a sum of the jounce bumper reaction force and the axial component of the applied spring load. The shape of the deformed system is shown in Figure 6.7b,

where the areas of plastic deformation were visualized in the cross-section (perpendicular to the direction of the least lateral stiffness).

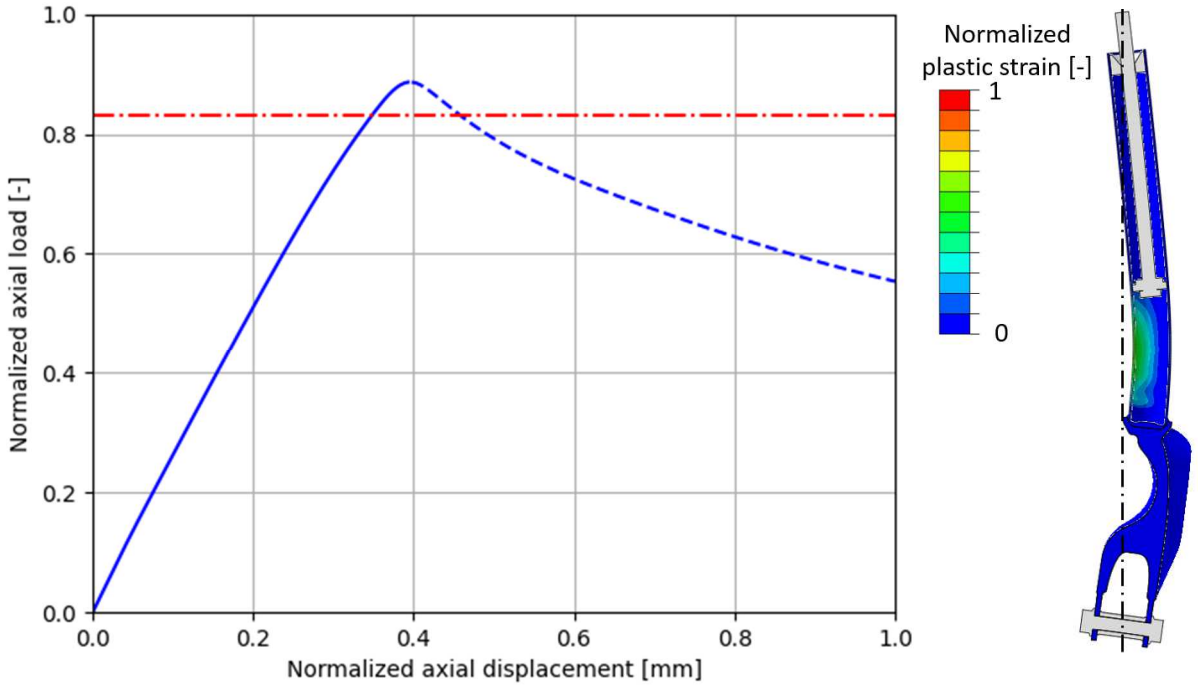

Figure 6.7. The simulation results of the optimized passive shock absorber: a) the load-deflection curve, b) normalized plastic strain distribution under max deformation

The maximum compression force before losing stability was $F_{max} = 0.886$, so 6% above the F_{limit} requirement. The analysis of stress and strain distribution over the shock absorber showed that the failure mode of the system was the tube, which was not part of the topology optimization, yet its strength is directly linked to the stiffness of the lower bracket (that is why changes in bracket material layout affected the constrained force F_{max} during the optimization). The overview of the normalized von Mises stress distribution for F_{limit} and for F_{max} is shown in Figure 6.8. For the F_{limit} load case, the plastic deformation starts to arise through the thickness of the compressed side of the tube. At the same time, localized plastic deformation is observed in the most out-of-axis cross-section of the bracket. For the F_{max} load case, the stress distribution over the bracket is almost the same as for the F_{limit} load case, however, the plastic deformation at the tube level spread significantly, both – through the thickness and along the height of the tube.

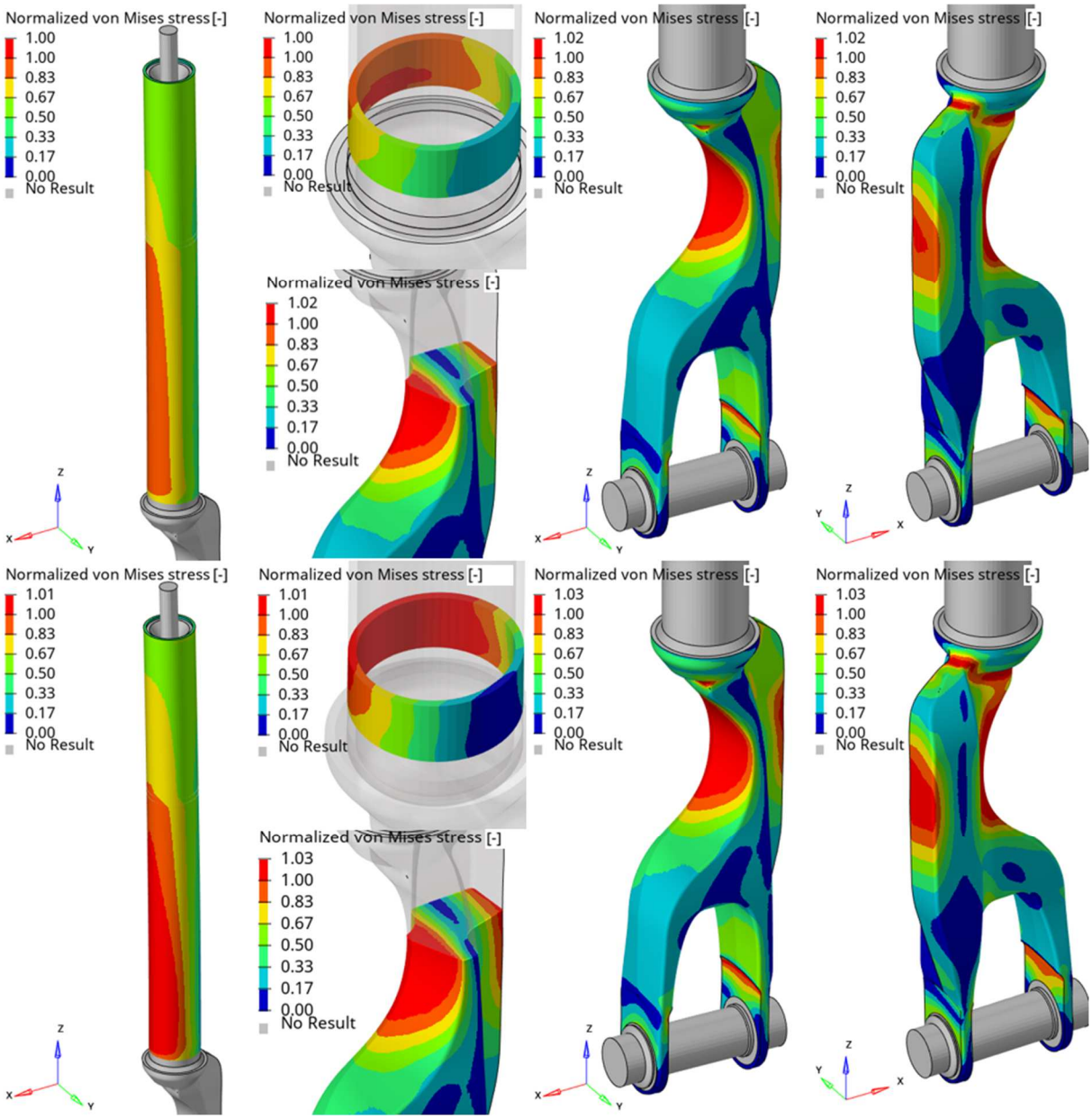

Figure 6.8. Normalized von Mises stress [-] distribution over the passive shock absorber tube and the bracket: a) at the F_{limit} load, b) at the F_{max} load

The stress measured at the bracket and at the tube was higher than the yield stress of the respective materials used for those components in both – the F_{limit} and F_{max} cases. Such a situation is desired, as the shock absorber may function correctly before the stability is lost, and the levels of plastic strain are below 6% of the total strain measured at the ultimate strength of the material, so the risk of material failure is considered to be negligible. The reason for losing stability is the behavior of the tube – the effect of the plastic hinge is observed at the cross-section in the height of maximum stress of the tube. Therefore the proposed passive shock absorber design was considered to be successfully optimized in terms of mass while fulfilling the stability constraint.

6.2. Optimization of the semi-active shock absorber

The second example of the proposed methodology was the semi-active shock absorber, which despite having the same role in the suspension, has a different structure compared to its passive counterpart.

6.2.1. The semi-active shock absorber model

The semi-active shock absorber that is subjected to mass minimization with stability constraint is shown in Figure 6.9. The grey part is the lower bracket, which is connected with the chassis via the bushing, similarly to the passive shock absorber, but it is connected with the tube by torqued thread connection instead of welding. All of the structural components are described using elastic-plastic material law with hardening.

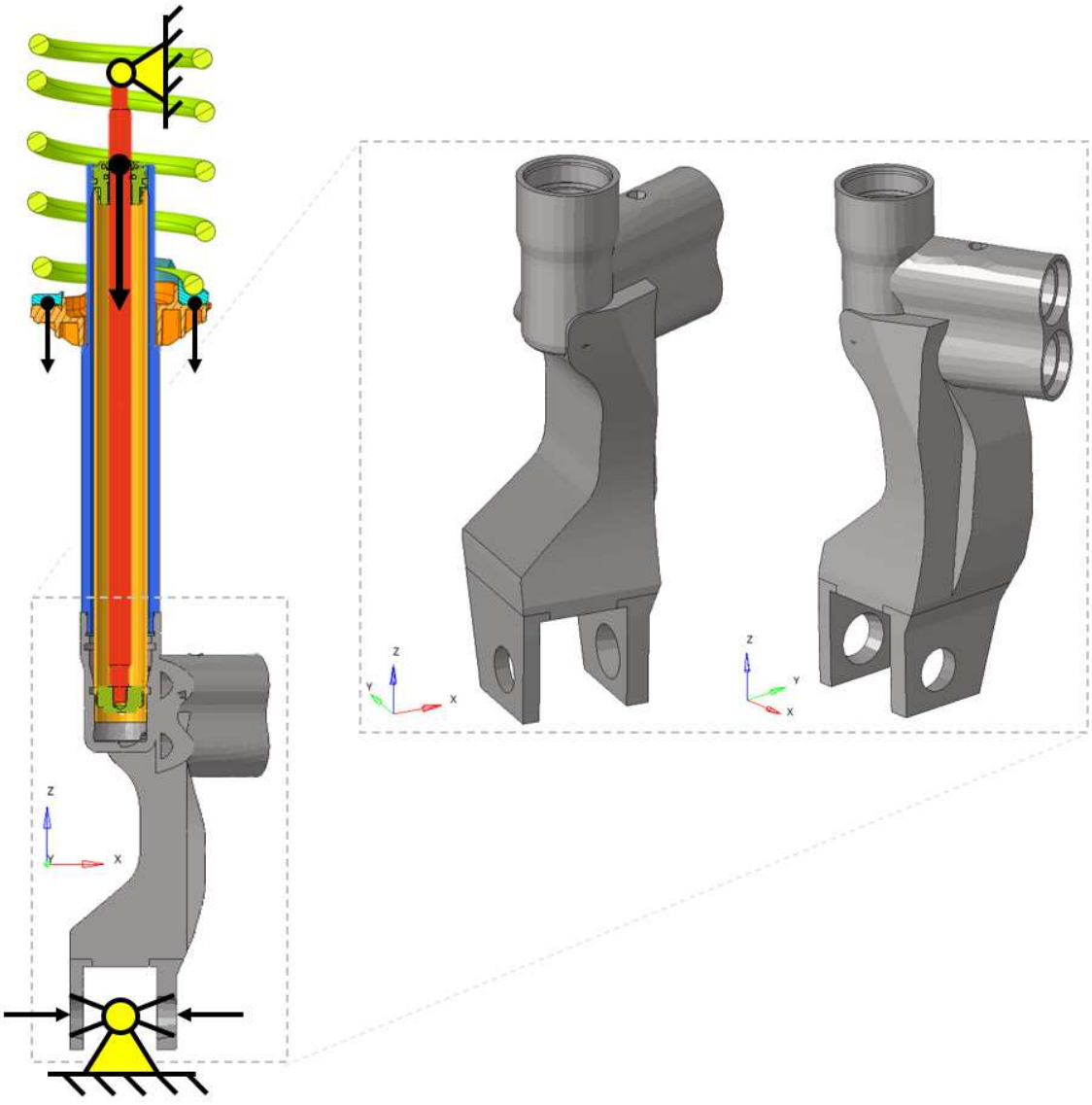

Figure 6.9. The model of the semi-active shock absorber

The same type of loading conditions are present as in the case of the previous example: assembly loads (modular spring and bushing bolt pre-loads) are applied before the incrementally increased jounce bumper axial load is applied to the shock absorber. The maximum force that the structure was able to withstand (having the material allocated within the whole design space) without losing stability was normalized to 1.0, the same as for the example described in Chapter 6.1. The constraint associated with minimum force at the loss of stability (F_{limit}) was 0.846. The mass of the total allowable design space was again normalized to 1.0.

6.2.2. Topology optimization - spatial LSM

The first topology optimization was done, similarly to the previous chapter, using the spatial version of the level set method. The same interpolation scheme was used (with 3x3x3 knots in X, Y, and Z directions). All of the optimization parameters (genetic operator values, number of objective function calls) were exactly the same as in the previous study. The only difference is the constant scaling factors $A_{RBF}, B_{RBF}, C_{RBF}$ from (4.6), which were adjusted to the size of the new part (design space).

The convergence of the proposed hybrid algorithms, as well as their average response, are shown in Figure 6.10:

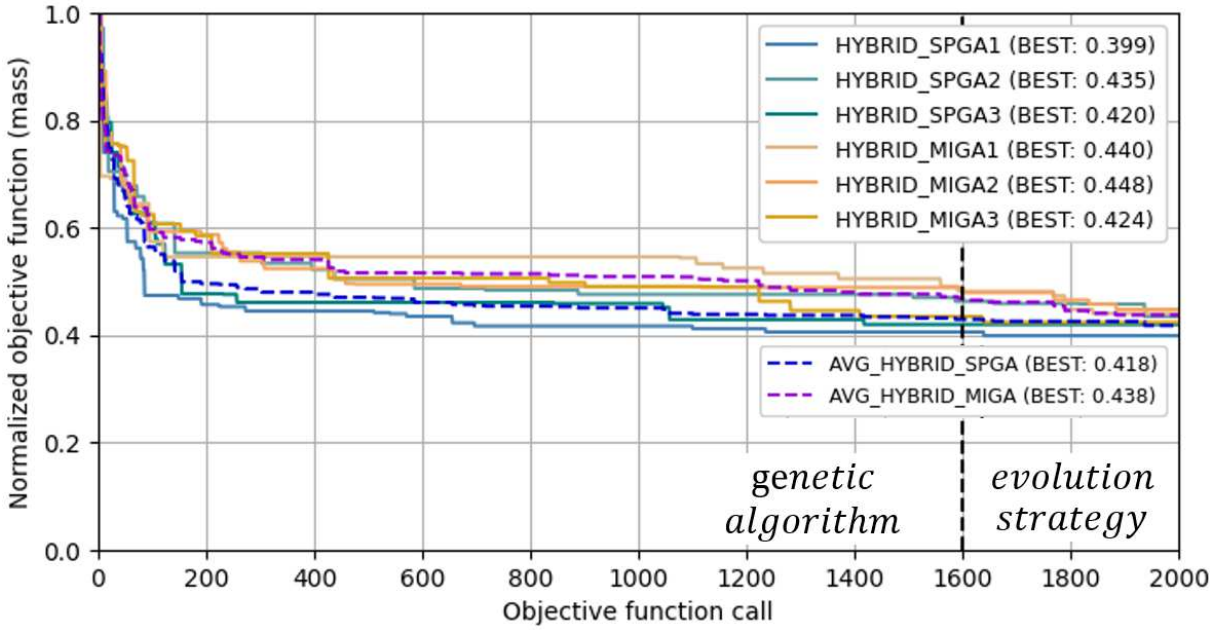

Figure 6.10. The convergence of the hybrid optimization algorithm – spatial LSM for the semi-active shock absorber constrained mass minimization

In the considered example, the difference between SPGA and MIGA average resultant mass was 3.8% after the genetic algorithm operation, and 1.9% after the

evolution strategy. In the case of a semi-active shock absorber, the SPGA-based optimization achieved the lowest final mass of the bracket and was also more efficient on average. The bracket shape evolution shape during the *Hybrid MIGA*1 optimization is shown in Figure 6.11:

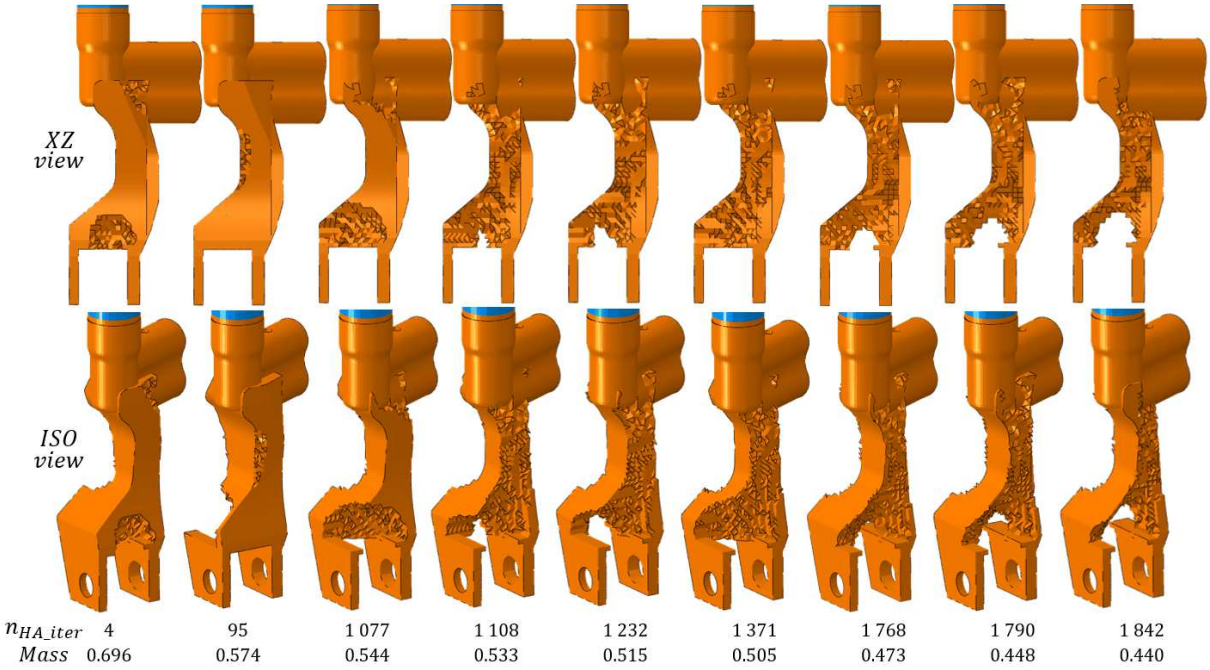

Figure 6.11. The evolution of the semi-active shock absorber bracket topology – *Hybrid MIGA*1 run

The first six shapes represent the evolution during the genetic algorithm operation, while the last three shapes reflect the changes induced by the evolution strategy. Similarly to the passive shock absorber bracket, the front part of the semi-active shock absorber bracket was mostly affected by the material removal, as its contribution to bending resistance was less significant compared to the rear part of the bracket. Additionally, the evolution strategy enabled to form a cavity between the braces, which has a similar form to the one seen in the previous example.

6.2.3. Topology optimization - projection method

The second method of optimization that was used to minimize the mass of the semi-active shock absorber bracket while fulfilling the stability constraint was the *projection method*. Again, due to the simplicity of this method, the Kriging-based surrogate modeling was used for optimization purposes. The same sizes of sampling plans were verified for their ability to create representative metamodels. The resultant coefficients of determination for the considered sampling plans are shown in Table 6.2. This time, the desired accuracy of the F_{max} prediction was maintained only for the

biggest sampling plan. The degradation of the R_V^2 for F_{max} response was observed for the medium and smallest of the sampling plans, while the R_V^2 for the *Mass* response maintained at acceptable level for all of the analyzed plans.

Table 6.2. The coefficient determination for Kriging (validation data) for various sampling plan sizes – semi-active shock absorber optimization

Sampling plan size	**4 400**	**1 100**	**275**
$R_V^2 - Mass$	0.996	0.992	0.982
$R_V^2 - F_{max}$	0.935	0.876	0.865

The Kriging F_{max} predictions vs target data (for training and validation data sets) for the max and min sampling plans are shown in Figure 6.12. The scatter in the F_{max} prediction for the validation samples is higher compared to the passive shock absorber bracket case (Figure 6.4), which is especially seen in the region of reduced strength (F_{max} values below 0.5).

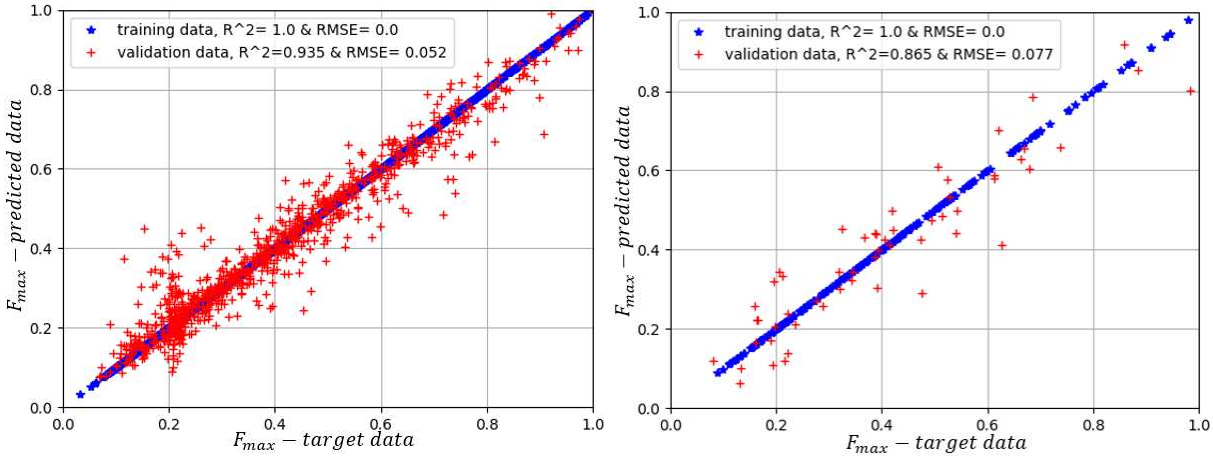

Figure 6.12. The Kriging prediction - F_{max} – for 4 400 and 275 samples used for OLHS – semi-active shock absorber optimization

The constrained mass minimization was performed using previously described versions of the hybrid algorithm. The optimization of the semi-active shock absorber bracket using metamodeling again converged faster compared to the *spatial LSM*, yet the convergence was not as sudden as in the case of the passive shock absorber example.

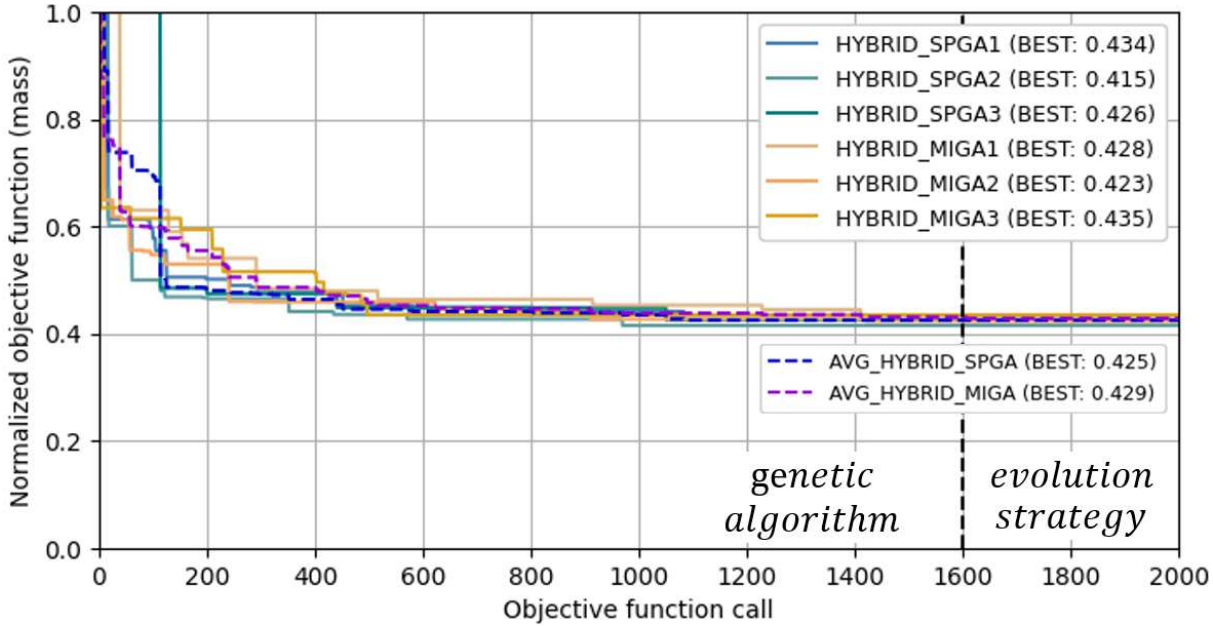

Figure 6.13. The convergence of the hybrid optimization algorithm – projection method for the semi-active shock absorber constrained mass minimization

This time, the average result from the MIGA-based *projection method* optimization was lower compared to the *spatial LSM* counterpart, with 0.429 vs 0.438 mass after optimization. Similarly to the passive shock absorber optimization, the lowest achieved mass value after optimization was observed for the hybrid algorithm incorporating the *spatial LSM*, yet in the semi-active shock absorber case, it was achieved using the hybrid algorithm with single population GA instead of multi-island GA. The evolution strategy in the semi-active shock absorber optimization utilizing the *projection method* brought minor improvement in the designs, with less than 1% of mass reduction over 400 objective function calls.

One of the main threats associated with metamodel-based optimization is the possibility of constraint violation due to the limited accuracy of the surrogates (approximated response). In most of the best-fitted individuals from the metamodel-based *projection method* optimization, the stability constraint was not fulfilled once the FEM analysis was run using the optimized design variables. Even though the magnitude of violation was moderate, additional FEM-based optimization was needed to resolve the premature loss of stability of the semi-active shock absorber. Therefore, in such cases, the hybrid optimization incorporating a metamodel-based genetic algorithm coupled with a FEM-based evolution strategy may be used to resolve the problem of constraint fulfillment, as the necessary design variables changes are limited and well-handled by the local search $(1+\lambda)\,ES$.

6.2.4. Verification of results

To verify if the optimized shape of the semi-active shock absorber bracket is indeed fulfilling the design requirements using real geometry, the CAD model was created based on the best individual from the *Hybrid MIGA*1 *spatial LSM* optimization. The CAD model was closely reflecting the obtained FEM model, with minor adjustments needed to maintain the manufacturing feasibility and smoothness of the final part. The comparison of the CAD and FEM models is shown in Figure 6.14. The shapes of the non-design areas (such as the lower part of braces or the transition between the bracket body and the tube cavity) were adjusted to match the optimal layout of the bracket, assuring a lack of sharp or uneven transitions between adjacent surfaces.

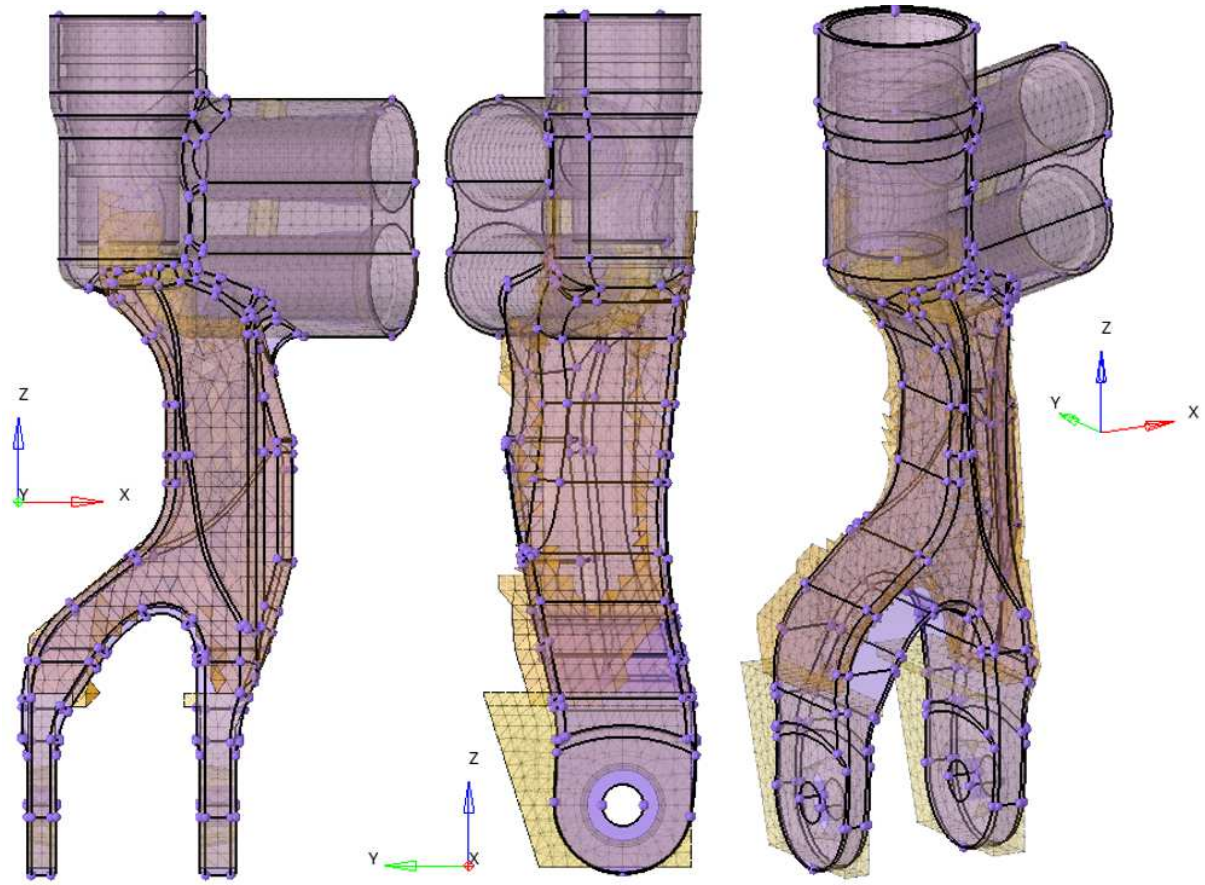

Figure 6.14. The comparison between the CAD and the FEM models – semi-active shock absorber after topology optimization

The model was verified using the same boundary conditions as in the optimization analysis, but again the shock absorber and the analysis were more detailed, as described in Chapter 6.1.1. Additionally, the connection between the tube and the bracket was modeled using contact interaction, therefore the stiffness of this connection could decrease under severe bending loads (as the contact interaction cannot transfer tensile loads when subjected to bending).

The first verified simulation output was the load-deflection response of the system, as shown in Figure 6.15a. Once the shock absorber was preloaded (lower bushing assembly loads, spring preload, and tube tightening), the jounce bumper load was added incrementally. The shape of the deformed system is shown in Figure 6.15b, where the areas of plastic deformation were visualized in the cross-section (perpendicular to the direction of the least lateral stiffness). This time, the plastic deformation was visible almost solely within the bracket, with the tube stressed below its yield stress.

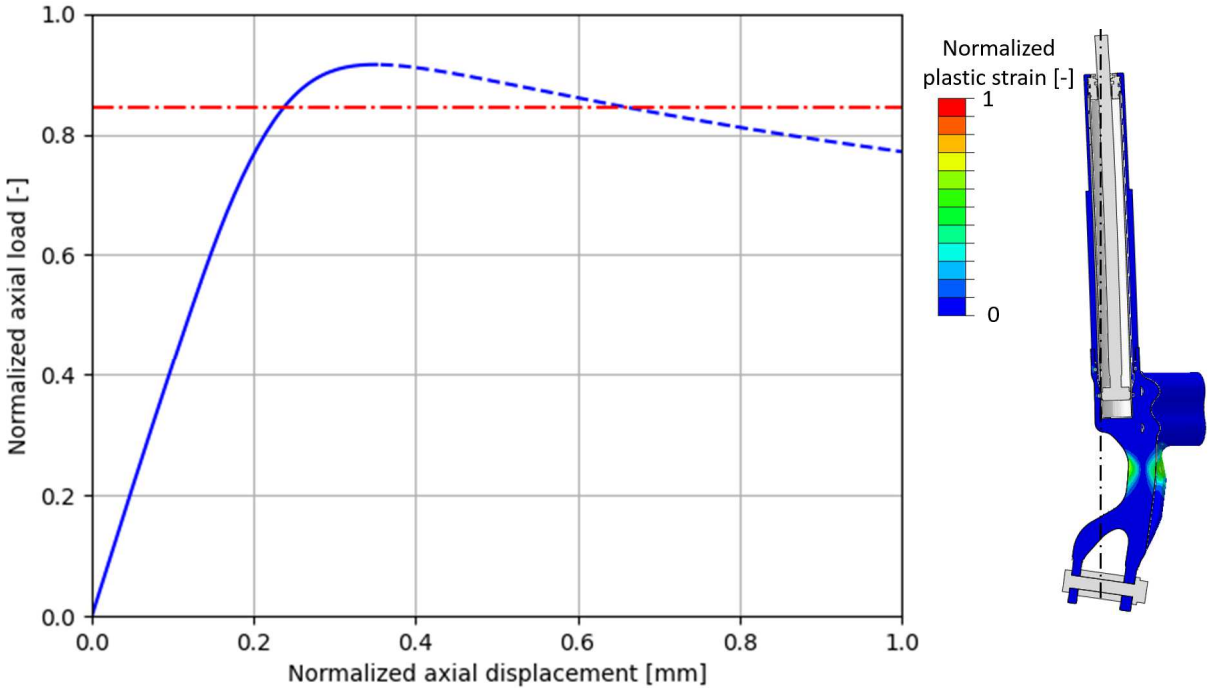

Figure 6.15. The simulation results of the optimized semi-active shock absorber: a) the load-deflection curve, b) normalized plastic strain distribution under max deformation

The maximum compression force before losing stability was $F_{max} = 0.915$, so 7% above the F_{limit} requirement. The analysis of stress and strain distribution over the shock absorber showed that the failure mode of the system was the lower bracket. The overview of the normalized von Mises stress distribution for F_{limit} and for F_{max} is shown in Figure 6.16. For the F_{limit} load case, the plastic deformation is observed for both the compressed and tensioned part of the bracket cross-section. Those plastic strain areas propagate until F_{max} load is reached, and no additional load can be transferred by the system. The plastic strain at the tube is localized to the diameters transition only, and associated with the notch generated by that transition.

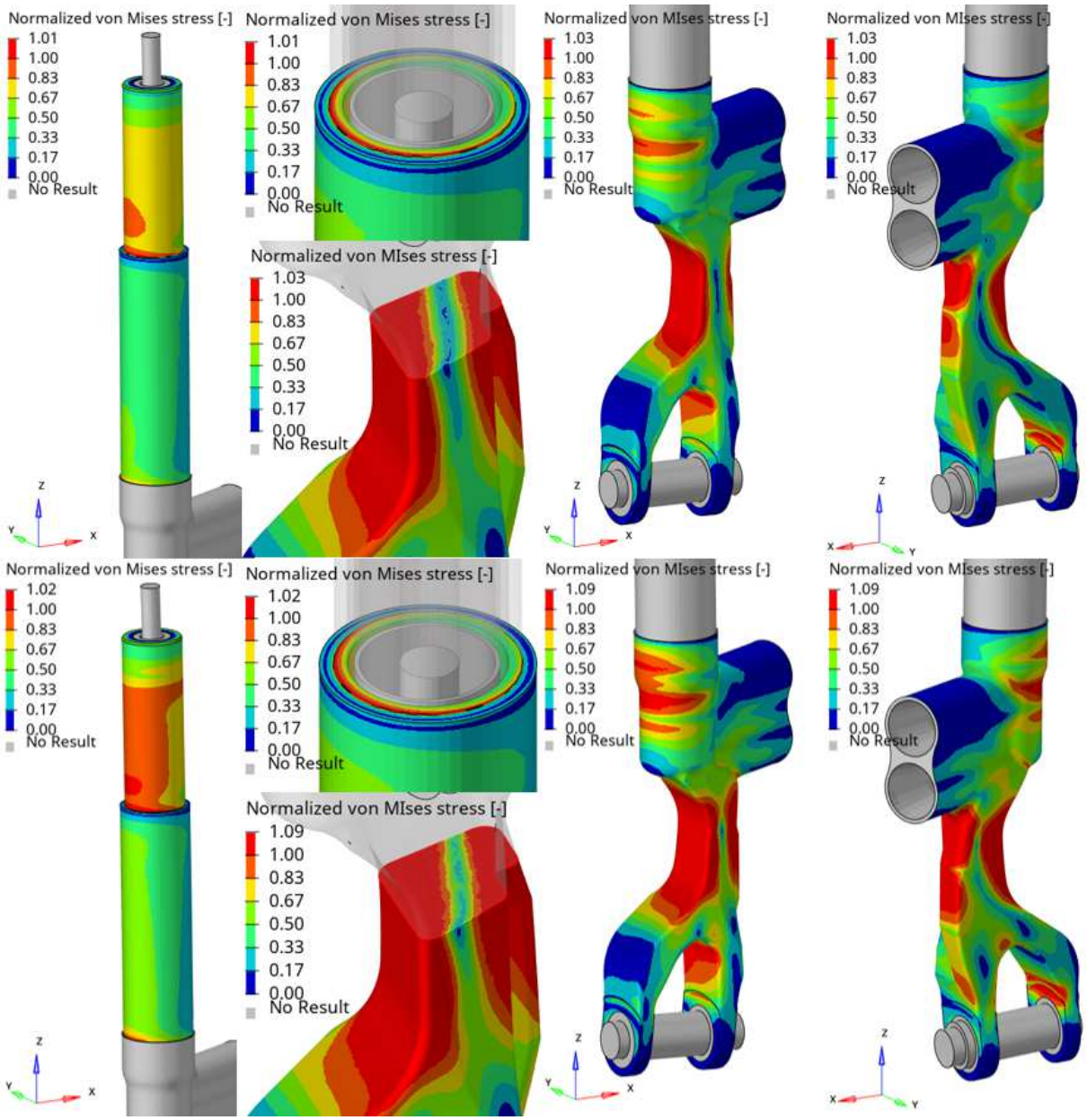

Figure 6.16. Normalized von Mises stress [-] distribution over the semi-active shock absorber tube and the bracket: a) at the F_{limit} load, b) at the F_{max} load

The reason for losing stability in the case of the semi-active shock absorber is the behavior of the bracket – the effect of the plastic hinge is observed at the cross-section at the height of the maximum drive shaft cavity. In contrary to the passive shock absorber, the transition between the beginning of plastic deformation, through the F_{limit} until the F_{max} is gradual. However, as the bracket is the main contributor to the failure of the system, it could explain why the metamodel training required more samples to achieve the desired prediction accuracy; in the passive shock absorber, the stiffness of the bracket was the main factor driving the strength of the system, however, in the semi-active shock absorber case, the plastic behavior of the bracket is the dominant factor. Stiffness is easier to predict, as it is a function of geometry, while the

plastic deformation is dependent on both – the geometry and the post-yield material behavior (the stress redistribution over the critical cross-section is not linear). Nonetheless, performed analyses showed that the proposed semi-active shock absorber design could be considered as successfully optimized in terms of mass while fulfilling the stability constraint.

6.3. Results summary

The constrained mass minimization examples shown in the previous section were solved using SPGA and MIGA-based hybrid algorithms, by means of either FEM or metamodeling techniques. The summary of all results is shown in Table 6.3, with the best and the average best individuals among the passive and semi-active shock absorber optimization highlighted in bold font. In all of the analyzed cases, the algorithms utilizing the *spatial LSM* were able to achieve the lowest mass among all combinations, regardless if the minimum or the averaged results were taken into consideration. However, the *projection method* achieves a lower spread in the final mass of the optimized components and usually converges quicker compared to the *spatial LSM* counterpart. In the majority of the analyzed examples, utilization of the evolution strategy (with 400 objective function calls) helped to reduce furtherly the objective function (*Mass*), even if marginal or no improvement was observed among the last 400 iterations of the genetic algorithm. That ultimately confirms the effectiveness of the GA & EVOL combination, regardless of the utilized type of the GA.

Table 6.3. The summary of optimization results

Algorithm	Spatial Level-Set Method (FEM)				Projection Method (Metamodel)			
	SPGA-based		MIGA-based		SPGA-based		MIGA-based	
Individual	Best	Average	Best	Average	Best	Average	Best	Average
Passive shock absorber	0.340	0.356	**0.323**	**0.339**	0.345	0.349	0.338	0.344
Semi-active shock absorber	**0.399**	**0.418**	0.424	0.438	0.415	0.425	0.423	0.429

The conducted full-scale, detailed FEM analyses confirmed that the obtained topologies (after hybrid optimization) were able to fulfill the strength requirements regarding the shock absorber stability in complicated loading conditions. Even though different components were causing the failure of the system (associated with the plastic hinge phenomenon), the moment of failure resulted from the mutual interactions between the strength and stiffness of the considered structural components, among which the optimized bracket played a crucial role in assuring the desired stability.

For the passive shock absorber, the failure was associated with the material plastic flow propagating through the thin wall of the slender tube, with the bracket operating mostly within elastic strain limits. The load-deflection characteristic showed a sudden drop in the system stiffness in the vicinity of the F_{max}, associated with the aforementioned cross-sectional plastic deformation. Any sudden changes in structural behavior are usually undesired, yet they are hard to overcome without significant design changes. Here, the optimized bracket allowed the structure to behave almost linearly through the whole operational load range and the failure occurred after the required F_{limit} was achieved. Such a situation is quite different from the semi-active shock absorber case, which utilizes less slender profiles, yet lower-strength materials. The optimized geometry of the bracket was sufficiently strong to withstand the F_{limit} requirement, yet the failure mechanism was strictly associated with the cross-sectional plastic deformation of the bracket instead of the tube. The transition between the linear (elastic) operating region and the peak of the load-deflection curve was smooth, and so was the post-buckling behavior, which is again opposite to the passive shock absorber case.

In both of the examined examples, the resultant bracket topologies consist of a cavity occurring in between the braces. This may seem counter-intuitive, as the material is removed from the area that is subjected to bending during the assembly process. However, during the bracket assembly loading step, the lower bushing (consisting of a simplified bolt and sleeve) is pre-tensioned with a certain load, that aims to generate the friction between the bushing components and the bracket. To do so, firstly the gap between the assembly clearance between the sleeve and the braces must be resolved. This results in minor (yet noticeable) brace bending which is a function of the clearance magnitude. Therefore, it can be treated as kinematic loading which is only required to close the given assembly clearance (gap). In such a case, high compliance (low bending stiffness) of the braces is desirable, in order to avoid excessive bending resistance which contributes to additional stress in the structure and reduces the resultant friction force in the bushing connection (as the pre-tensioning force is unnecessarily spent on bending).

The fitting of the Kriging model (based on the OLHS plans) enabled to see the contribution of specific parameters (RBF scaling factors s_i in both methods and braces-related cavity parameters in the *projection method*). The most important insight was the fact, that there are no parameters that could be easily excluded to perform dimensionality reduction. This could be anticipated, as both proposed methods are based on the global-range RBFs (either auxiliary field in *spatial LSM* or contour curves in the *projection method*).

7. Summary

This chapter is divided into subsequent sections: 7.1 in which the conclusions regarding the proposed methodology are withdrawn, 7.2 which describes the implementation of the methodology within the company, and 7.3 which presents the planned, future research directions.

7.1. Conclusions

The work presented in this dissertation concerns the problem of automotive shock absorber mass optimization with consideration of the stability of the suspension system. As this problem is not trivial, multiple steps were necessary to take in order to finally propose an efficient optimization method.

Firstly, the method of numerical analysis capable of capturing all of the nonlinearity sources was chosen. The finite element method utilizing the Newton-Raphson method for solving the force-driven, nonlinear static strength problem was incorporated in the research. This type of method is able to capture the peak of the load-bearing capacity within a very small tolerance (e.g. 0.01%) without the drawbacks of either the Riks method (also known as the Arc-length method) or the displacement-driven N-R method. The FEM simulation allowed to measure two desired outputs, that were then used for optimization purposes – the total mass of the considered component ($Mass$) and the maximum load that can be transferred by the shock absorber without losing the stability (F_{max}).

Secondly, two methods of topology and shape modifications were proposed. The first of those methods, referred to as the *spatial Level-Set Method,* is able to generate any arbitrary form of the optimized body within the given resolution. One of the crucial steps was to find the perfect balance between the anticipated resolution of the method and its complexity (type and number of variables). The resultant setting of the method consisted of 27 variables, all of which were the scaling factors of the radial basis functions. Other variables, such as the threshold value, position of the RBF centers, or type of the RBF were fixed on constant levels based on the conducted studies. A dedicated filtering routine was developed to assure the manufacturability of the optimized components. During each iteration of the optimization, the final topology of the considered component was iteratively scanned, and infeasible areas were adjusted based on the assumed forging direction. The manufacturability was realized on the cost of additional optimization complexity, i.e. the nonlinear relation between the design variables (RBF scaling factors) and design outputs ($F_{max}, Mass$) got even more nonlinear. The second of the proposed topology and shape modification methods was

the *projection method*, in which the common volume of two perpendicular plane projections formed the final body of the structure. Again, the RBF-based type of field description was utilized to affect the resultant shape. This method utilized 22 parameters, with 20 directly responsible for the external contours of the projections (2 contours with 5 RBFs per side) and 2 parameters responsible for the cavity in between the braces. Contrary to the *spatial LSM*, this method did not utilize any additional filtering routine, as the projection itself assures that the resultant shape is forgeable. The *projection method* is a simpler method, as the resultant design freedom is reduced by the capabilities of the perpendicular projections. However, the resolution of the contour is slightly higher compared to the *spatial LSM.*

Thirdly, the hybrid optimization algorithm was proposed, which consists of the genetic algorithm and evolution strategy realized consecutively. The motivation for such a combination was the global-search capabilities of the short-chromosome genetic algorithm and the local-search abilities of the evolution strategy, with both being well-suited for not smooth functions, such as F_{max} and $Mass$ responses. The problem with the lack of smoothness is a result of the hard-kill element approach associated with the implemented version of the topology optimization method. The responses values change in a quantified way, i.e. they are constant over a small ranges of design parameters and change to another level once another finite element is added or removed. This phenomenon makes the usage of gradient methods (incorporating finite-difference calculations) inefficient. What is more, the proposed hybrid algorithm does not require any prior knowledge about the shock absorber layout, as the initial individuals (initial population) are generated randomly, so there is no need for specifying the starting point for the optimization. The latter is the main drawback of other global and local-search techniques, such as Simulated Annealing, Hooke-Jeeves method, Nelder-Mead Simplex, or any of the gradient-based techniques. Even the multi-start approach utilizing the aforementioned methods (i.e. numerous different random individuals used as starting points for separate independent optimizations) does not guarantee achieving a better solution within the given optimization time limit than the proposed evolutionary-based hybrid optimization algorithm. On top of that, the proposed version of the hybrid algorithm was compared with a purely GA-based approach (without the local search) realized over the increased number of objective function calls (either by additional generations or by the increased size of the population) and outperformed it in every case.

Furthermore, the efficiency of the proposed hybrid optimization algorithm was increased by the tuning process, which was realized by means of surrogate modeling. The reason for such a choice was the fact, that it was nearly impossible to verify all of the considered parameter combinations utilizing the costly finite element method.

Therefore, small, yet trend-representative sets of training and validation samples were generated using FEM simulations and then used to build the metamodels. Afterward, those models were called over 1.5 million times during the tuning process. Two types of surrogates were considered: interpolating Kriging and approximating artificial neural networks. Both of those models were built and tuned by the dissertation author using Python programming language and associated (freeware) site-packages. The choice of the models was based on the literature study of dominant and universal models used in engineering regression analysis and related fields of research. Both metamodel types showed different strengths and weaknesses, with ANNs being able to quickly learn the underlying nature of the problem (for a given network layout), on the cost of a broad search of optimal network topology and its hyperparameters. Kriging on the other hand occurred to be easier to adjust, with very few parameters to tune (bounds, optimizer, number of optimizer restarts, or type of the correlation function). However, the Kriging method is susceptible to design points that are in very close proximity, therefore they usually require optimization of the sampling plan, which was performed by usage of the optimal Latin hypercube sampling method, which maximizes the minimum distance between the sampling points. The research was extended to consider the size of the training and validation sampling plans. The *Mass* response occurred to be predicted well even with a limited number of available data to feed the metamodels. On the other hand, the quality of F_{max} response prediction was sensitive to the size of the sampling plan, regardless of the type of utilized surrogate model. Most importantly, the trained surrogate models could predict the new, unseen data within a small fraction of a second, which allowed them to be utilized in the hybrid optimization algorithm parameters tuning process. As a result, the efficiency of the proposed hybrid algorithm was increased compared to the average from the study by 5-12%, depending on the version of the algorithm. Two of the aforementioned versions were proposed: one consisting of a single population, subjected to relatively long evolution, and a second, consisting of multiple populations (multi-island version of the algorithm) having more individuals but less than half of the generations compared to its single population counterpart. Even though both of those GA variants obtain their quasi-optimal solution in a different way, the local-search capabilities of the evolution strategy enable to find the final individual with a very similar fitness function (on average). The single-population GA cultivates the initial population over a long period of generations by means of selection, cross-over, and mutation operators, while the proposed multi-island GA exploits the randomness of the numerous individuals spread over the design domain. On top of that, the MIGA not only utilizes the aforementioned genetic operators but also interchange individuals between the autonomous population, increasing the diversity on each of the islands. On average, both methods (SGPA and

MIGA) should yield very similar results, however, a higher spread was observed for the single-population GA.

Lastly, the verification of the proposed hybrid optimization algorithm after the parameter tuning process was performed on two industrial cases. In both of the considered examples, the optimization was performed using the *spatial LSM* and the *projection method,* incorporating the parameter-tuned SPGA and MIGA-based hybrid optimization algorithms. The *spatial LSM* showed good capabilities in generating arbitrary topologies, with the material allocated in the most critical parts of the available design space. Thanks to the implemented filtering routines, the final geometries were forgeable, despite having intricate resultant shapes after the optimization. The metamodel-based optimization utilizing the *projection method* achieved slightly less-fitted individuals, on the benefit of the lower spread between results and quicker convergence. The resultant shapes were less complicated compared to the *spatial LSM* due to the inherent simplicity of the proposed method. In the case of the passive shock absorber, when the bracket stiffness was the critical issue (not its strength per se) the metamodeling occurred to be an interesting choice for the optimization, as even an extremely low number of training samples was sufficient to construct an efficient surrogate able to correctly predict the system responses F_{max} and $Mass$. However, in the case of the semi-active shock absorber, when not only the stiffness but also the elastic-plastic behavior of the bracket was crucial, the metamodeling required more effort to construct a similarly efficient predictor. Even with a good approximator ($R_V^2 = 0.935$), the final individual from the metamodel-based optimization required additional processing using FEM-based optimization to fall into an unpenalized region of sufficient stability. However, the *spatial LSM*-based topologies of the lower shock absorber brackets after optimization were used to form refined (smoothed) 3D CAD models, representing closely the resultant shapes. They were then verified in a nonlinear, system-level FEM analysis without any simplifications adopted during the optimization phase, and according to the sophisticated standards required by both the company (dissertation author employer) and the car manufacturers. Performed simulations confirmed the fulfillment of the stability requirements, correct structural behavior, and lack of additional failure modes introduced by the mass-optimized brackets geometry. The SPGA-based hybrid algorithm showed better performance in optimization of the semi-active shock absorber, where the system failure was related strictly to the component subjected to topology optimization, in both – the *spatial LSM* and *projection method* cases. On the other hand, the MIGA-based hybrid algorithm performed better in the case of passive shock absorber optimization, again for both considered shape modification methods. In that case, the failure mode was connected to the neighboring component (the tube), yet it

may yield the conclusion, that MIGA-based HA should be used for slender shock absorbers with the tube being the failure mode, while the SGPA-based HA should be incorporated in optimization of shock absorbers where the brackets are the main failure mode of the system (as in the semi-active shock absorber optimization case).

In conclusion, all of the objectives associated with the doctoral project were met, and the presented dissertation delivers a few original insights into the fields of interest:

1. The computationally-efficient, nonlinear, force-driven FEM-based numerical simulation which is capable of predicting the peak force assuring the stability of the shock absorber (or any other column-type structure) subjected to the extreme compression loads.
2. Coupling the aforementioned computer simulation with one of two proposed methods of shape modifications, that allows for obtaining a wide range of shapes
3. Two shape modification methods utilizing implicit geometrical representation of topology by means of radial basis functions, that can be easily coupled with any arbitrary mathematical optimization algorithms, and which assures the given manufacturability constraint (forgeable shape) using dedicated routine or the nature of the method itself
4. The hybrid optimization algorithm that couples the genetic algorithm with the evolution strategy, which can be easily parallelized to achieve high computational efficiency, and which is able to efficiently optimize the structures showing non-smooth responses, such as ones associated with hard-kill optimization methods
5. Method of tuning the optimization algorithm parameters utilizing the wide set of metamodeling techniques, including the response surface method, artificial neural networks, and Krigings

The aforementioned conclusions can be considered as unequivocally confirming the thesis formulated in Chapter 1.2, i.e. it is possible to formulate an optimization method and algorithm that allows for mass minimization of the shock absorber taking into consideration the stability of the whole system under extreme compression loads and its manufacturing requirements. Besides the points listed above, numerous procedures, routines, and computer programs were developed to support the data preparation, optimization process, and results processing, with the dominant usage of the Python programming language.

7.2. Industrial implementation of optimization methodology

The proposed optimization methodology has been gradually implemented in the company during the doctoral project execution. It allowed to couple the methods developed during the doctoral project not only with stability-related problems but also with ones related to stiffness and fatigue constraints. As for the main aim of the implementation, i.e. efficient optimization technique for automotive shock absorbers subjected to stability constraints, all of the assumed milestones have been met, with most of them before the actual due date. The proposed method allowed to perform the topology optimization tasks without encountering the problems typically seen in such aspects before, i.e. lack of algorithms convergence or infeasibility of the quasi-optimal solutions caused by poor manufacturability of the final topologies. Examples presented in this dissertation show that the utilization of the method enabled a 60-68% reduction of the initial (allowable) material volume, providing both cost and performance-effectiveness, which is crucial for the shock absorber manufacturing company (dissertation authors employer).

The initial doctoral project assumptions were not including the metamodeling part of the presented work, which after all turned out to be very useful and usable in daily engineering challenges, not only related to optimization. The results of shock absorber optimization utilizing artificial intelligence were presented at industrial conferences throughout the implementation doctoral project. Utilizing the Python-based frameworks and site-packages, the metamodeling tools developed during the project were exceeding the commercial solutions in terms of flexibility and e.g. training speed, thanks to the newest achievements in computer programming methods allowing for maximum exploitation of available computer resources.

7.3. Future tasks

There are several points that the author of the dissertation is planning to realize regardless of the finished doctoral project, which are a result of new possibilities associated with the developed methods:

1. Replacing the currently used objective function relying on the mass of the shock absorber with the manufacturing cost of the parts being optimized. This can be done in cooperation with e.g. bracket suppliers, that provide insights that couple the geometrical features of the parts with the final cost of the part (like maximum circumference, smoothness of transitions, minimum or maximum thicknesses, etc.)
2. Adding additional (partially heuristic) constraints to the optimization that can narrow down the set of admissible solutions, like the axial or lateral stiffness, stress response at the shock absorber components (besides the component subjected to the topology optimization), or minimum cross-sectional moment of inertia (in respect to the central axis of the shock absorber)
3. Adding additional optimization variables associated with simultaneous modification of other shock absorber components (like local thinning of tubes in semi-active solutions)
4. Creating the post-processing routine to smooth the resultant harsh contour of the part, which could increase the efficiency of the CAD model buildup

References

[1] M. Abdi, "Evolutionary Topology Optimization of Continuum Structures using X-FEM and Isovalues of Structural Performance," 2015.

[2] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2018.

[3] G. Allaire, C. Dapogny, and F. Jouve, "Shape and topology optimization," in *Handbook of Numerical Analysis*, vol. 22, Elsevier, 2021, pp. 1–132.

[4] G. Allaire, F. Jouve, and A.-M. Toader, "Structural optimization using sensitivity analysis and a level-set method," *Journal of Computational Physics*, vol. 194, no. 1, pp. 363–393, Feb. 2004.

[5] W. J. Al-Mudhafar and K. Sepehrnoori, "Designed Simulations for Optimization of Hydraulic Fracture Design and Production Well Constraints in Shale Gas Reservoirs with Reduced-Physics Metamodeling," in *Day 4 Thu, June 14, 2018*, Copenhagen, Denmark, 2018, p. D042S014R002.

[6] J. Arabas and J. Tenniel, *Wykłady z algorytmów ewolucyjnych*, Wyd. 2. Warszawa: Wydawnictwa Naukowo-Techniczne, 2004.

[7] K.-J. Bathe and K.-J. Bathe, *Finite element procedures*. Englewood Cliffs, N.J: Prentice Hall, 1996.

[8] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural Optimization*, vol. 1, no. 4, pp. 193–202, Dec. 1989.

[9] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197–224, Nov. 1988.

[10] M. P. Bendsøe and O. Sigmund, "Material interpolation schemes in topology optimization," *Archive of Applied Mechanics (Ingenieur Archiv)*, vol. 69, no. 9–10, pp. 635–654, Nov. 1999.

[11] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies - A comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[12] A. Bhosekar and M. Ierapetritou, "Advances in surrogate based modeling, feasibility analysis, and optimization: A review," *Computers & Chemical Engineering*, vol. 108, pp. 250–267, Jan. 2018.

[13] G. E. P. Box and D. W. Behnken, "Some New Three Level Designs for the Study of Quantitative Variables," *Technometrics*, vol. 2, no. 4, pp. 455–475, Nov. 1960.

[14] G. E. P. Box and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 13, no. 1, pp. 1–38, Jan. 1951.

[15] C. G. Broyden, "The convergence of a class of double rank minimization algorithms," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.

[16] T. Burczyński, W. Kuś, A. Długosz, and P. Orantek, "Optimization and defect identification using distributed evolutionary algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 4, pp. 337–344, Jun. 2004.

[17] T. Burczyński and P. Orantek, "Coupling of genetic and gradient algorithms," *Proceedings of conference on evolutionary algorithms and global optimization, Złoty Potok*, pp. 112–114, 1999.

[18] T. Burczyński, A. Poteralski, and M. Szczepanik, "Immune and Swarm Optimization of Structures," in *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, vol. 36, D. Greiner, B. Galván, J. Périaux, N. Gauger, K. Giannakoglou, and G. Winter, Eds. Cham: Springer International Publishing, 2015, pp. 295–308.

[19] A. Čerškus, V. Ušinskis, N. Šešok, I. Iljin, and V. Bučinskas, "Optimization of Damping in a Semi-Active Car Suspension System with Various Locations of Masses," *Applied Sciences*, vol. 13, no. 9, p. 5371, Apr. 2023.

[20] T. Y. Chen and Y. H. Chiou, "Structural Topology Optimization Using Genetic Algorithms," *Proceedings of the World Congress on Engineering*, vol. 3, 2013.

[21] I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, and T. Dhaene, "Blind Kriging: Implementation and performance analysis," *Advances in Engineering Software*, vol. 49, pp. 1–13, Jul. 2012.

[22] R. Courant, "Variational methods for the solution of problems of equilibrium and vibrations," *Bull. Amer. Math. Soc.*, vol. 49, no. 1, pp. 1–23, 1943.

[23] F. Czerwinski, "Current Trends in Automotive Lightweighting Strategies and Materials," *Materials*, vol. 14, no. 21, p. 6631, Nov. 2021.

[24] F. Del Pero, L. Berzi, A. Antonacci, and M. Delogu, "Automotive Lightweight Design: Simulation Modeling of Mass-Related Consumption for Electric Vehicles," *Machines*, vol. 8, no. 3, p. 51, Sep. 2020.

[25] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, "Level-set methods for structural topology optimization: a review," *Struct Multidisc Optim*, vol. 48, no. 3, pp. 437–472, Sep. 2013.

[26] T. Dozat, "Incorporating Nesterov Momentum into Adam," presented at the International Conference on Learning Representations, San Juan, Puerto Rico, 2016.

[27] S. Dreyfus, "The numerical solution of variational problems," *Journal of Mathematical Analysis and Applications*, vol. 5, no. 1, pp. 30–45, Aug. 1962.

[28] F. Ferrari and O. Sigmund, "Towards solving large-scale topology optimization problems with buckling constraints at the cost of linear analyses," *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112911, May 2020.

[29] R. Fletcher, "A new approach to variable metric algorithms," *Computer J.*, vol. 13, pp. 317–322, 1970.

[30] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*. Wiley, 1966.

[31] S. S. Garud, I. A. Karimi, and M. Kraft, "Design of computer experiments: A review," *Computers & Chemical Engineering*, vol. 106, pp. 71–95, Nov. 2017.

[32] D. E. Goldberg and K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in *Foundations of Genetic Algorithms*, vol. 1, Elsevier, 1991, pp. 69–93.

[33] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Math. Comp.*, vol. 24, no. 109, pp. 23–26, 1970.

[34] I. Griva, S. Nash, and A. Sofer, *Linear and nonlinear optimization*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2009.

[35] C. Grosan and A. Abraham, *Intelligent Systems: A Modern Approach*, vol. 17. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[36] J. Gu, "An efficient multiple meta-model-based global optimization method for computationally intensive problems," *Advances in Engineering Software*, vol. 152, p. 102958, Feb. 2021.

[37] M. Guiggiani, *The Science of Vehicle Dynamics: Handling, Braking, and Ride of Road and Race Cars*. Dordrecht: Springer Netherlands, 2014.

[38] D. Guirguis and M. F. Aly, "A derivative-free level-set method for topology optimization," *Finite Elements in Analysis and Design*, vol. 120, pp. 41–56, Nov. 2016.

[39] D. Guirguis, W. W. Melek, and M. F. Aly, "High-resolution non-gradient topology optimization," *Journal of Computational Physics*, vol. 372, pp. 107–125, Nov. 2018.

[40] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach," *Information*, vol. 10, no. 12, p. 390, Dec. 2019.

[41] C. He, Y. Zhang, D. Gong, and X. Ji, "A review of surrogate-assisted evolutionary algorithms for expensive optimization problems," *Expert Systems with Applications*, vol. 217, p. 119495, May 2023.

[42] B. Heissing and M. Ersoy, Eds., *Chassis handbook: fundamentals, driving dynamics, components, mechatronics, perspectives*, 1st edition. Wiesbaden: Vieweg + Teubner, 2011.

[43] C. Henley, "The Neuron," Jan. 2021.

[44] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors." arXiv, 03-Jul-2012.

[45] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1st MIT Press ed. Cambridge, Mass: MIT Press, 1992.

[46] E. Holmberg, B. Torstenfelt, and A. Klarbring, "Stress constrained topology optimization," *Struct Multidisc Optim*, vol. 48, no. 1, pp. 33–47, Jul. 2013.

[47] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[48] A. Hrennikoff, "Solution of Problems of Elasticity by the Framework Method," *Journal of Applied Mechanics*, vol. 8, no. 4, pp. A169–A175, Dec. 1941.

[49] A. G. Ivakhnenko and V. G. Lapa, *Cybernetic Predicting Devices*. CCM Information Corporation, 1965.

[50] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments," *Journal of Statistical Planning and Inference*, vol. 134, no. 1, pp. 268–287, Sep. 2005.

[51] S. Junk and N. Rothe, "Lightweight design of automotive components using generative design with fiber-reinforced additive manufacturing," *Procedia CIRP*, vol. 109, pp. 119–124, 2022.

[52] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.

[53] A. Keane, A. Forrester, and A. Sobester, *Engineering Design via Surrogate Modelling: A Practical Guide*. Washington, DC: American Institute of Aeronautics and Astronautics, Inc., 2008.

[54] H. J. Kelley, "Gradient Theory of Optimal Flight Paths," *ARS Journal*, vol. 30, no. 10, pp. 947–954, Oct. 1960.

[55] R. Kieszek, S. Kachel, and A. Kozakiewicz, "Modification of Genetic Algorithm Based on Extinction Events and Migration," *Applied Sciences*, vol. 13, no. 9, p. 5584, Apr. 2023.

[56] M. Kleiber and A. Borkowski, Eds., *Handbook of computational solid mechanics: survey and comparison of contemporary methods*. Berlin ; New York: Springer, 1998.

[57] Y. S. Kong, S. Abdullah, M. Z. Omar, and S. M. Haris, "Topological and Topographical Optimization of Automotive Spring Lower Seat," *Lat. Am. j. solids struct.*, vol. 13, no. 7, pp. 1388–1405, Jul. 2016.

[58] D. G. Krige, "A statistical approach to some mine valuations and allied problems at the Witwatersrand," Master's thesis, University of Witwatersrand, 1951.

[59] W. Kus, "Evolutionary Optimization of Forging Anvils Using Grid Based on Alchemi Framework," in *2006 Second IEEE International Conference on e-Science and Grid Computing (e-Science'06)*, Amsterdam, The Netherlands, 2006, pp. 121–121.

[60] W. Kuś and T. Burczyński, "Distributed Evolutionary Algorithms in Optimization of Nonlinear Solids," in *IUTAM Symposium on Evolutionary Methods in Mechanics*, vol. 117, T. Burczyński and A. Osyczka, Eds. Dordrecht: Kluwer Academic Publishers, 2004, pp. 229–239.

[61] W. Kuś and T. Burczyński, "Parallel Bioinspired Algorithms in Optimization of Structures," in *Parallel Processing and Applied Mathematics*, vol. 4967, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1285–1292.

[62] W. Kuś and T. Burczyński, "Distributed evolutionary algorithm and grids in optimization of structures," *Inteligencia Artificial*, vol. 9, pp. 49–53, 2005.

[63] J. Kusiak, A. Danielewska-Tułecka, and P. Oprocha, *Optymalizacja: wybrane metody z przykładami zastosowań*. Warszawa: Wydawnictwo Naukowe PWN, 2009.

[64] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Y. Ng, "On Optimization Methods for Deep Learning," 2011.

[65] E. Lindgaard and J. Dahl, "On compliance and buckling objective functions in topology optimization of snap-through problems," *Struct Multidisc Optim*, vol. 47, no. 3, pp. 409–421, Mar. 2013.

[66] E. Lindgaard and E. Lund, "Optimization formulations for the maximum nonlinear buckling load of composite structures," *Struct Multidisc Optim*, vol. 43, no. 5, pp. 631–646, May 2011.

[67] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1–3, pp. 503–528, Aug. 1989.

[68] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[69] Z. Michalewicz and Z. Nahorski, *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Wyd. 3. Warszawa: Wydawnictwa Naukowo-Techniczne, 2003.

[70] A. C. Mitra, M. V. Patil, and N. Banerjee, "Optimization of Vehicle Suspension Parameters for Ride Comfort Based on RSM," *J. Inst. Eng. India Ser. C*, vol. 96, no. 2, pp. 165–173, Apr. 2015.

[71] M. Mobin, S. M. Mousavi, M. Komaki, and M. Tavana, "A hybrid desirability function approach for tuning parameters in evolutionary optimization algorithms," *Measurement*, vol. 114, pp. 417–427, Jan. 2018.

[72] T. Mollik, Y. Geng, M. R. E. U. Shougat, T. Fitzgerald, and E. Perkins, "Genetic algorithm shape optimization to manipulate the nonlinear response of a clamped-clamped beam," *Heliyon*, vol. 8, no. 11, p. e11833, Nov. 2022.

[73] L. A. T. Mororó and F. P. Van Der Meer, "Combining the thick level set method with plasticity," *European Journal of Mechanics - A/Solids*, vol. 79, p. 103857, Jan. 2020.

[74] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of Statistical Planning and Inference*, vol. 43, no. 3, pp. 381–402, Feb. 1995.

[75] M. Mosayebi and M. Sodhi, "Tuning genetic algorithm parameters using design of experiments," in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, Cancún Mexico, 2020, pp. 1937–1944.

[76] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer New York, 2006.

[77] P. Orantek, "Zastosowanie algorytmów hybrydowych w zagadnieniach optymalizacji i identyfikacji dynamicznych układów mechanicznych," PhD Thesis, Silesian University of Technology, Gliwice, 2002.

[78] S. N. Patnaik, J. D. Guptill, and L. Berke, "Merits and limitations of optimality criteria method for structural optimization," *NASA Technical Paper*, vol. 3373, 1993.

[79] A. Poteralski, "Optymalizacja strukturalna przestrzennych układów mechanicznych z wykorzystaniem algorytmów ewolucyjnych," PhD Thesis, Silesian University of Technology, Gliwice, 2004.

[80] O. M. Querin, G. P. Steven, and Y. M. Xie, "Evolutionary structural optimisation (ESO) using a bidirectional algorithm," *Engineering Computations*, vol. 15, no. 8, pp. 1031–1048, Dec. 1998.

[81] E. Raponi, M. Bujny, M. Olhofer, N. Aulig, S. Boria, and F. Duddeck, "Kriging-assisted topology optimization of crash structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 348, pp. 730–752, May 2019.

[82] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[83] C. M. C. Razali, S. S. Abdullah, A. Parnianifard, and A. Faruq, "Adaptive infill sampling strategy for metamodeling: Challenge and future research directions," *Bulletin EEI*, vol. 9, no. 5, pp. 2020–2029, Oct. 2020.

[84] I. Rechenberg, "Evolutionsstrategien," in *Simulationsmethoden in der Medizin und Biologie*, vol. 8, B. Schneider and U. Ranft, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 83–114.

[85] J. N. Reddy, *An introduction to the finite element method*, 3rd ed. New York, NY: McGraw-Hill Higher Education, 2006.

[86] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[87] S. Rosenthal, F. Maaß, M. Kamaliev, M. Hahn, S. Gies, and A. E. Tekkaya, "Lightweight in Automotive Components by Forming Technology," *Automot. Innov.*, vol. 3, no. 3, pp. 195–209, Sep. 2020.

[88] E. Roux and P.-O. Bouchard, "Kriging metamodel global optimization of clinching joining processes accounting for ductile damage," *Journal of Materials Processing Technology*, vol. 213, no. 7, pp. 1038–1047, Jul. 2013.

[89] M. J. de Ruiter and F. van Keulen, "Topology Optimization: Approaching the Material Distribution Problem using a Topological Function Description," presented at the The Fifth International Conference on Computational Structures Technology, Leuven, Belgium, pp. 111–119.

[90] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[91] J. B. Russ and H. Waisman, "A novel elastoplastic topology optimization formulation for enhanced failure resistance via local ductile failure constraints and linear buckling analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113478, Jan. 2021.

[92] M. J. Sasena, P. Papalambros, and P. Goovaerts, "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization," *Engineering Optimization*, vol. 34, no. 3, pp. 263–278, Jan. 2002.

[93] S. M. Savaresi, C. Poussot-Vassal, C. Spelta, L. Dugard, and O. Sename, *Semi-Active Suspension Control Design for Vehicles*, 1e éd. Oxford Burlington, Mass: Butterworth-Heinemann, 2010.

[94] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: Mit einer vergleichenden Einführung in die Hill-Climbing- und Zufallsstrategie*. Basel: Birkhäuser Basel, 1977.

[95] P. Sebastjan and W. Kuś, "Method for Parameter Tuning of Hybrid Optimization Algorithms for Problems with High Computational Costs of Objective Function Evaluations," *Applied Sciences*, vol. 13, no. 10, p. 6307, May 2023.

[96] P. Sebastjan and W. Kuś, "Hybrid Shape Optimization of Automotive Spring Seat," *Int.J Automot. Technol.*, vol. 23, no. 4, pp. 957–965, Aug. 2022.

[97] P. Sebastjan and W. Kuś, "Optimization of material distribution for forged automotive components using hybrid optimization techniques," *cmms*, vol. 21, no. 2, 2021.

[98] J. A. Sethian and A. Wiegmann, "Structural Boundary Design via Level Set and Immersed Interface Methods," *Journal of Computational Physics*, vol. 163, no. 2, pp. 489–528, Sep. 2000.

[99] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comp.*, vol. 24, no. 111, pp. 647–656, 1970.

[100] G. A. da Silva, A. T. Beck, and O. Sigmund, "Topology optimization of compliant mechanisms considering stress constraints, manufacturing uncertainty and geometric nonlinearity," *Computer Methods in Applied Mechanics and Engineering*, vol. 365, p. 112972, Jun. 2020.

[101] J. A. Snyman and D. N. Wilke, *Practical Mathematical Optimization*, vol. 133. Cham: Springer International Publishing, 2018.

[102] J. V. Soares Do Amaral, J. A. B. Montevechi, R. D. C. Miranda, and W. T. D. S. Junior, "Metamodel-based simulation optimization: A systematic literature review," *Simulation Modelling Practice and Theory*, vol. 114, p. 102403, Jan. 2022.

[103] C.-Y. Song, "A Study on Learning Parameters in Application of Radial Basis Function Neural Network Model to Rotor Blade Design Approximation," *Applied Sciences*, vol. 11, no. 13, p. 6133, Jul. 2021.

[104] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[105] M. Stolpe and K. Svanberg, "An alternative interpolation scheme for minimum compliance topology optimization," *Struct Multidisc Optim*, vol. 22, no. 2, pp. 116–124, Sep. 2001.

[106] K. Svanberg, "The method of moving asymptotes—a new method for structural optimization," *Int. J. Numer. Meth. Engng.*, vol. 24, no. 2, pp. 359–373, Feb. 1987.

[107] K. Szajek, "Optimization of a two-component implantology system using genetic algorithm," PhD Thesis, Poznan University of Technology, Poznan, 2012.

[108] M. Szczepanik, "Optymalizacja układów powierzchniowych z wykorzystaniem algorytmów ewolucyjnych," PhD Thesis, Silesian University of Technology, Gliwice, 2003.

[109] H. H. Tan and K. H. Lim, "Review of second-order optimization techniques in artificial neural networks backpropagation," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 495, p. 012003, Jun. 2019.

[110] E.-M. Tarek, A. Hopgood, L. Nolle, and A. Battersby, "Hybrid Genetic Algorithms: A Review," *Engineering Letters*, vol. 3, no. 2, 2006.

[111] M. Treiber and V. Kanagaraj, "Comparing Numerical Integration Schemes for Time-Continuous Car-Following Models," *Physica A: Statistical Mechanics and its Applications*, vol. 419, pp. 183–195, Feb. 2015.

[112] S.-L. Twu and R. L. Geisler, "Structural Topology Optimization of Multilink Suspension System Using ATOM," 2012.

[113] E. Tyfopoulos, F. D. Tollnes, M. Steinert, and A. Olsen, "State of the art of generative design and topology optimization and potential research needs," *DS 91: Proceedings of NordDesign*, 2018.

[114] S. L. Vatanabe, T. N. Lippi, C. R. D. Lima, G. H. Paulino, and E. C. N. Silva, "Topology optimization with manufacturing constraints: A unified projection-based approach," *Advances in Engineering Software*, vol. 100, pp. 97–112, Oct. 2016.

[115] G. G. Wang and S. Shan, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," *Journal of Mechanical Design*, vol. 129, no. 4, pp. 370–380, Apr. 2007.

[116] M. Y. Wang, X. Wang, and D. Guo, "A level set method for structural topology optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 1–2, pp. 227–246, Jan. 2003.

[117] S. Wang and M. Y. Wang, "Radial basis functions and level set method for structural topology optimization," *Int. J. Numer. Meth. Engng*, vol. 65, no. 12, pp. 2060–2090, Mar. 2006.

[118] D. Whitley, S. Rana, and R. B. Heckendorn, "The Island Model Genetic Algorithm: On Separability, Population Size and Convergence," *Journal of Computing and Information Technology*, vol. 7, 1998.

[119] G.-H. H. Won-Tae Jang, "An Adaptive Evolutionary Algorithm Combining Evolution Strategy and Genetic Algorithm (Application of Fuzzy Power System Stabilizer)," *Advances in Evolutionary Algorithms*, 2008.

[120] P. Wriggers, *Nonlinear finite element methods*. Berlin: Springer, 2008.

[121] G. Wszołek, "Multi-objective model-based design optimization of hydraulic shock absorbers," *Computer Assisted Methods in Engineering and Science*, vol. 23, no. 2/3, pp. 147–166, 2016.

[122] Y. M. Xie and G. P. Steven, *Evolutionary Structural Optimization*. London: Springer London, 1997.

[123] Y. M. Xie and G. P. Steven, "A simple evolutionary procedure for structural optimization," *Computers & Structures*, vol. 49, no. 5, pp. 885–896, Dec. 1993.

[124] J.-J. Zhang, L.-W. Xu, and R.-Z. Gao, "Multi-island Genetic Algorithm Opetimization of Suspension System," *TELKOMNIKA*, vol. 10, no. 7, pp. 1685–1691, Nov. 2012.

[125] H. Zhao, R. Zhang, and B. Zeyun, "A Review of Automotive Lightweight Technology," *Advances in Engineering Research*, vol. 149, 2018.

[126] O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, 5th ed. Oxford ; Boston: Butterworth-Heinemann, 2000.

[127] O. C. Zienkiewicz and R. L. Taylor, *The finite element method*, 5th ed. Oxford ; Boston: Butterworth-Heinemann, 2000.

[128] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*, Seventh edition. Amsterdam: Elsevier, Butterworth-Heinemann, 2013.

[129] "Abaqus: Mechanical and Civil Engineering Simulation." [Online]. Available: https://www.3ds.com/products-services/simulia/products/abaqus/. [Accessed: 13-Jun-2023].

[130] "Isight & SIMULIA Execution Engine by Dassault Systèmes®." [Online]. Available: https://www.3ds.com/products-services/simulia/products/isight-simulia-execution-engine/. [Accessed: 13-Jun-2023].

[131] "Minitab: Data Analysis, Statistical & Process Improvement Tools." [Online]. Available: https://www.minitab.com/en-us/. [Accessed: 13-Jun-2023].

[132] "Welcome to Python.org," *Python.org*, 22-Jun-2023. [Online]. Available: https://www.python.org/. [Accessed: 25-Jun-2023].

[133] "PyTorch Machine Learning Framework." [Online]. Available: https://www.pytorch.org. [Accessed: 13-Jun-2023].

[134] "Scikit-learn Machine Learining Framework." [Online]. Available: https://scikit-learn.org/stable/. [Accessed: 13-Jun-2023].

Abstract

This dissertation deals with the optimization of the shape of automotive shock absorber components in regard to their mass, taking into consideration the constraints associated with the stability of the entire suspension system. The first part of the dissertation presents the fundamental theoretical background needed to understand and reproduce the topology optimization process, including the work principles of automotive shock absorbers, their role in the chassis, the method of simulating their behavior under critical loads using the finite element method, as well as optimization and metamodeling techniques.

The main part of thesis focuses on the adaptation and extension of the existing methods of topology and shape modification to the considered shock absorber problem. As a result, two methods based on the implicit description of the geometry are proposed: *the spatial level set method* and *the projection method*. Design variables from both methods were combined with the proposed hybrid optimization algorithm, consisting of consecutive execution of the genetic algorithm (in one of two variants) and the evolutionary strategy (1+λ).

Subsequently, this hybrid optimization algorithm was subjected to the process of tuning its parameters, i.e. the genetic operators and the parameters of the penalty function, in order to increase its efficiency within the assumed number of iterations (simulations). For this purpose, two groups of metamodels were created: artificial neural networks and Krigings. Best-performing models were then used to replace the costly FEM analysis in the DOE plan utilizing the response surface method, which enabled the verification of the nonlinearity of the input parameters of the hybrid algorithm (genetic operators, parameters of the penalty function) on its ability to minimize the mass of the suspension system while maintaining its stability.

The algorithm adapted in this way was verified in the final part of the dissertation on two real industrial examples: optimization of a passive and semi-active automotive shock absorber. Both methods of geometry modification and two versions of the hybrid algorithm were compared, as well as the metamodeling capabilities of the proposed surrogates to replace the FEM simulation during the optimization process.

The thesis is summarized with a discussion of the results and formulation of the further scope of work regarding the optimization of automotive shock absorbers in terms of their mass, with consideration of their unstable behavior.

Streszczenie

Niniejsza rozprawa dotyczy zagadnienia optymalizacji kształtu komponentów amortyzatora samochodowego pod kątem masy, jednak z uwzględnieniem ograniczeń związanych ze statecznością całego układu zawieszenia. W pierwszej części dysertacji przedstawiono niezbędne podstawy teoretyczne do zrozumienia i odtworzenia procesu optymalizacji, włączając w to zasadę działania oraz rolę amortyzatorów samochodowych, symulowanie ich pracy pod działaniem krytycznych obciążeń z wykorzystaniem metody elementów skończonych, oraz techniki optymalizacji i metamodelowania.

W części głównej skupiono się na adaptacji i rozszerzeniu istniejących metod modyfikacji topologii i kształtu części do rozpatrywanego zagadnienia, proponując dwie metody oparte o niejawny opis geometrii: przestrzenną metodę poziomic oraz metodę rzutowania. Parametry zmienne z obu metod połączono z zaproponowanym hybrydowym algorytmem optymalizacji, składającym się z kolejno wykonywanych: algorytmu genetycznego (w jednej z dwóch odmian) oraz strategii ewolucyjnej $(1 + \lambda)$.

Następnie ów hybrydowy algorytm optymalizacji poddano procesowi dostosowania operatorów genetycznych oraz parametrów funkcji kary w celu zwiększenia jego efektywności przy założonych więzach co do całkowitej ilości wykonanych iteracji (symulacji). W tym celu najpierw stworzono dwie grupy metamodeli spośród których wybrano najlepiej rokujące: sztuczną sieć neuronową oraz Kriging. Posłużyły one do zastąpienia kosztownej analizy MES przy ich wielokrotnym użyciu, w celu wykonania założonego eksperymentu opartego o metodę powierzchni odpowiedzi, co umożliwiło weryfikację nieliniowości oddziaływania parametrów wejściowych algorytmu hybrydowego (operatory genetyczne, parametry funkcji kary) na jego zdolność do minimalizacji masy układu zawieszenia przy zachowaniu jego stateczności.

Tak dostosowany algorytm zweryfikowano w końcowej części pracy na dwóch rzeczywistych przykładach przemysłowych: optymalizacji pasywnego oraz półaktywnego amortyzatora samochodowego. Porównano obie metody modyfikacji geometrii oraz dwie wersje algorytmu hybrydowego, jak również możliwości metamodelowania do zastąpienia rzeczywistego modelu opartego o symulacje MES.

Całość pracy wieńczy podsumowanie, dyskusja wyników oraz sformułowanie dalszego zakresu pracy nad zagadnieniem optymalizacji amortyzatorów samochodowych pod kątem ich masy z uwzględnieniem stateczności całego układu zawieszenia.